u évora
ESCOLA DE CIÊNCIAS E TECNOLOGIA

JIUE 2010

Actas das Jornadas de Informática da Universidade de Évora 2010

José Saias, Luís Rato e Teresa Gonçalves

Évora, Portugal
17 de Novembro de 2010

Escola de Ciências e Tecnologia
Universidade de Évora
2010

ISBN: 978-989-97060-0-2

http://host.di.uevora.pt/jiue2010

Sobre o evento JIUE 2010

As primeiras Jornadas de Informática da Universidade de Évora (JIUE2010) tiveram no lugar no dia 17 de Novembro de 2010, no Anfiteatro 1 do CLV. Este evento tem como objectivo divulgar os trabalhos desenvolvidos na área de Informática, no âmbito de teses de mestrado e doutoramento, projectos de investigação e colaboração com empresas realizados em ligação com Unidades Orgânicas da Universidade de Évora.

À semelhança da submissão de artigos numa conferência, os trabalhos apresentados passaram por um processo de revisão. Dos 18 trabalhos submetidos, foram aceites 16. Cada artigo foi revisto por 2 ou 3 professores do Departamento de Informática (DI). Dos 16 trabalhos aceites, um reportou trabalho desenvolvido no Departamento de Engenharia Rural e os restantes trabalhos realizados em colaboração com o Departamento de Informática.

Dos trabalhos desenvolvidos em colaboração com o DI, a grande maioria inseriu-se em trabalhos desenvolvidos no âmbito de teses de mestrado (11). Houve também trabalhos desenvolvidos no âmbito de tese de doutoramento (1), de disciplinas de doutoramento (1) e de mestrado (1) e de projectos desenvolvidos em colaboração com empresas (2; um deles inserido numa tese de mestrado).

Foram aceites trabalhos em diferentes áreas da Informática: extracção de informação (3), processamento de imagem (4), classificação de documentos (1), reconhecimento de entidades nomeadas (3), redes (1), interfaces (2), modelos computacionais (1) e aplicações práticas (1).

As Jornadas começaram pela manhã com uma palestra convidada proferida pelo Professor Victor Lobo (Escola Naval e ISEGI/UNL), tendo-se seguido as apresentações de 15 trabalhos.

A organização das JIUE2010 agradece o patrocínio da Delta-Cafés.

JIUE 2010

Criação de uma Ontologia e Respectiva Povoação a Partir do Processamento de Relatórios Médicos

Luís Borrego
Universidade de Évora
Évora, Portugal
luis_borrego@hotmail.com

Paulo Quaresma
Universidade de Évora
Évora, Portugal
pq@di.uevora.pt

Resumo

A evolução tecnológica tem provocado uma evolução na medicina, através de sistemas computacionais voltados para o armazenamento, captura e disponibilização de informações médicas. Entre outros documentos médicos, os relatórios médicos são na maioria das vezes guardados num texto livre não estruturado e escritos com vocabulário proprietário, podendo ocasionar falhas de interpretação. Através das linguagens da Web Semântica, é possível utilizar ontologias como modo de estruturar e padronizar a informação dos relatórios médicos, adicionando-lhe anotações semânticas. Desta forma, a informação contida nos relatórios médicos pode ser publicada na *Web*, permitindo às máquinas o processamento automático da informação. O presente trabalho incide na criação de uma ontologia e respectiva povoação, através de técnicas de Processamento de Linguagem Natural e de Aprendizagem Automática que permitem extrair a informação dos relatórios médicos. À posteriori foi desenvolvida uma aplicação, que permite ao utilizador converter os relatórios do formato digital para o formato OWL.

1 Introdução

Nos últimos anos, tem-se verificado a criação e aperfeiçoamento de sistemas que permitem um melhor tratamento da informação da área médica, principalmente quando nos deparamos com a descoberta de novas patologias, com o aparecimento de novos medicamentos, etc, que tem provocado uma grande expansão da informação médica.

A integração da informação da área médica é um dos grandes desafios da informática na saúde. Os exames e consequentemente os relatórios médicos são muitas vezes efectuados em diversas instituições, que utilizam vocabulários proprietários, e se esta informação não for devidamente preparada e apresentada de maneira apropriada, pode ocasionar falhas de interpretação.

Para solucionar este problema pode-se recorrer ao uso de ontologias, que permitem estruturar e modelar conceitos e relações de forma correcta, evitando ambiguidades e estruturando o conhecimento existente de um dado domínio [10]. Através das linguagens da Web Semântica[1] é possível representar ontologias, adicionando anotações semânticas.

Para facilitar o preenchimento da ontologia, podem ser empregues processos automáticos ou semi-automáticos. Uma das áreas mais utilizada para extracção de informação é a área da Mineração de Textos, que utiliza técnicas, métodos e algoritmos de outras áreas, tais como, processamento de linguagem natural e aprendizagem automática [5].

Assim, através da criação de ontologias e do seu preenchimento, é possível representar a informação contida nos relatórios médicos, permitindo que as diversas instituições utilizem sistemas que processem automaticamente essa informação, trazendo consequentemente uma melhor manutenção do cuidado de saúde do paciente. Essa informação pode ainda ser usada para fins de investigação ou para criar sistemas de auxílio aos médicos no apoio à decisão diagnóstica.

[1] A Web Semântica é uma expansão da *Web* actual, onde a informação possui um significado claro e definido, possibilitando uma melhor interacção entre computadores e pessoas [2].

O objectivo principal deste trabalho é criar uma ontologia que permita representar a informação expressa nos relatórios médicos e criar mecanismos automáticos que extraiam a informação e efectuem a referida ”povoação”da ontologia. Deste modo, iremos obter uma base de conhecimento para este domínio, tornando possível uma nova utilização dos dados textuais através do processamento automático por máquinas.

2 Trabalho Relacionado

O desenvolvimento das linguagens ontológicas, tem permitido que nas mais diversas áreas se desenvolvam sistemas que usam ontologias como forma de estruturar os seus dados. Nos últimos anos na área da saúde, têm-se desenvolvido alguns trabalhos neste âmbito.

Em 2003, Nardon [7] utiliza ontologias e base de dados dedutivas para partilha de conhecimento da área da saúde. São utilizadas tecnologias da Web Semântica em conjunto com uma linguagem de consulta baseada em base de dados dedutivas, de modo a permitir a partilha da base de conhecimento e a resolução de consultas mais complexas.

Mais tarde, em 2005, Cantais et al. [3] desenvolveram um protótipo que faz parte do projecto Personalized Information Plataform (PIPS) que tem como objectivo auxiliar um paciente diabético na escolha dos alimentos. O sistema utiliza ontologias como base de conhecimento e foram utilizados os termos da base de dados de alimentos do instituto Entidade de Investigação e Desenvolvimento da Universidade Politécnica de Valência (ITACA).

Farias et al. [4] constroem uma ontologia para a gestão do conhecimento das Doenças Sexualmente Transmissíveis, trazendo consigo a conceitualização necessária para a compreensão do processo de desenvolvimento de ontologias.

Recentemente, Ferreira et al. (2008) desenvolveram o projecto MedAlert [6]. O MedAlert possuí um *corpus* com cerca de 48.229 textos relativos a episódios de internamento ocorridos no Hospital Infante D. Pedro, em Aveiro. Utiliza como suporte ontologias e outras fontes de conhecimento que permitem extrair a informação dos textos médicos, de modo a inferir, de uma forma automática, irregularidades/dúvidas suscitadas pelas decisões tomadas pelos profissionais de saúde.

Também em 2008, Annibal et al. [1] apresentam uma ontologia como forma de estruturar a informação de Relatórios Radiológicos. O objectivo deste trabalho é investigar o domínio léxico e semântico dos textos presentes em relatórios radiológicos, e desenvolver uma ontologia que permita o armazenamento destes documentos de forma estruturada, sem interferir no modo como os utilizadores interagem com um sistema já existente, sistema esse que é responsável pela informatização de vários processos, entre eles o de permitir criar relatórios resultantes da análise de imagens digitais.

3 Metodologia

O trabalho proposto é desenvolvido em quatro fases, como mostra a figura 1.

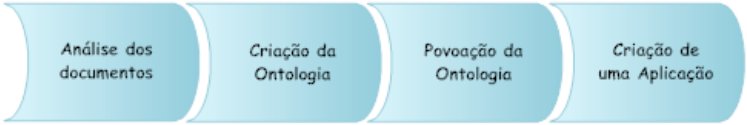


Figura 1: Fases de Desenvolvimento do Trabalho

3.1 Análise dos Documentos

Para desenvolver o trabalho é necessário obter um conjunto de documentos, neste caso de relatórios médicos, que formem o *corpus*. O conjunto de relatórios médicos utilizados no trabalho correspondem a relatórios neurovasculares, que foram fornecidos pelo Laboratório LUSCAN do Hospital do Espírito Santo de Évora. O apêndice A corresponde a um exemplo deste tipo de relatório, no formato digital (TXT).

Nesta fase do trabalho, é fundamental analisar e identificar todos os campos dos relatórios de forma a ser possível representar toda a informação. Deste modo, verifica-se a existência de cinco zonas bem distintas:

1. Identificação do Paciente (nº de processo, data de exame, nome, idade e serviço)
2. Informação Clínica
3. Exames realizados[2] (tipo de exame, aparelho e janelas ultra-sônicas caso existam)e respectiva descrição
4. Conclusão do relatório
5. Identificação do médico(s) e técnico(s)

3.2 Criação da Ontologia

A partir da análise dos documentos é possível criar uma ontologia que cubra o domínio dos relatórios neurovasculares. Para tal, é necessário escolher uma linguagem para representar ontologias e seguir uma metodologia de construção de ontologias.

Entre as várias linguagens existentes para representar ontologias, a linguagem OWL foi a escolhida, pois é mais eficiente na *Web* do que XML, RDF e RDFS, e é hoje em dia uma recomendação da W3C, principalmente devido à sua semântica formal capaz de processar conteúdo semântico na *Web*.

Na construção de uma ontologia, deve-se sempre aplicar uma metodologia durante o seu processo de desenvolvimento. Como tal, adoptou-se por seguir a metodologia proposta pelo grupo da Universidade de Stanford, que refere que a modelação de uma ontologia pode ser realizada seguindo alguns passos (figura 2) que representam a sequência de criação e modelação da ontologia [8].

Figura 2: Passos da Metodologia para a Criação da Ontologia

A ontologia foi construída de raiz devido à escassez de trabalhos desenvolvidos no âmbito deste trabalho, muito provavelmente, devido à Web Semântica ser uma área relativamente recente.

A partir da análise realizada anteriormente aos relatórios médicos foram enumerados todos os termos, e com a ajuda dos profissionais de saúde do laboratório LUSCAN foi possível criar todas as classes, as suas propriedades e restrições.

A ontologia foi desenvolvida utilizando a ferramenta Protégé[3]. O esquema global do domínio é composto por treze classes relacionadas entre si, como mostra a figura 3.

[2] cada relatório pode conter um ou vários exames

[3] `http://protege.stanford.edu/`

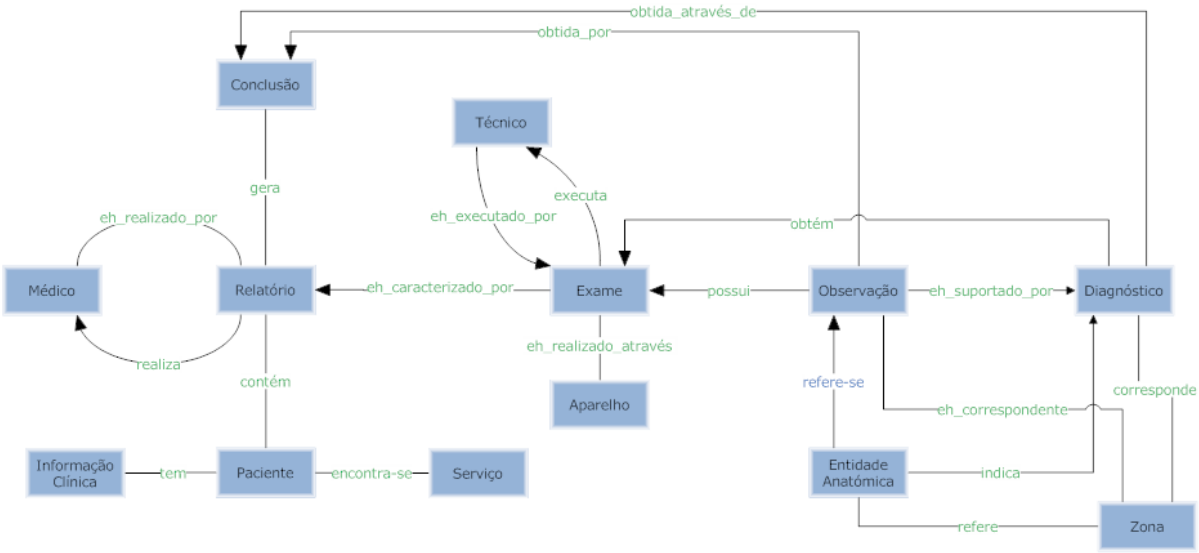


Figura 3: Modelação do Esquema Global do Domínio

- Relatório *eh_realizado_por* **um** Médico, *contém* **um** Paciente, *gera* **uma** Conclusão, *eh_caracterizado_por* **alguns** Exames
- Médico *realiza* **alguns** relatórios
- Paciente *tem* **uma** Informação Clínica, *encontra-se* **num** Serviço
- Conclusão *obtida_por* **algumas** Observações, *obtida_através_de* **alguns** Diagnósticos
- Exame *eh_executado_por* **alguns** Técnicos, *eh_realizado_através* de **um** Aparelho, *possui* **algumas** Observações, *obtém* **alguns** Diagnósticos
- Técnico *executa* **alguns** Exames
- Observação *eh_correspondente* a **uma** Zona, *refere-se* a **algumas** Entidades Anatómicas
- Diagnósticos *eh_suportado_por* **algumas** Observações, *indica* **algumas** Entidades Anatómicas, *corresponde* a **uma** Zona
- Zona *refere* **uma** Entidade Anatómica

3.3 Extracção da Informação e Povoamento da Ontologia

Para extrair a informação contida nos relatórios do *corpus* e povoar automaticamente a ontologia, foi desenvolvido um sistema que é constituído pelos seguintes módulos: 1) Preenchimento dos campos da identificação do paciente que se encontram vazios; 2) Preenchimento dos campos de uma tabela da base de dados com os dados contidos nos relatórios; 3) Criação de estruturas e aplicação de técnicas que permitem processar a informação e povoar a ontologia.

Os relatórios médicos fornecidos, possuíam os campos da identificação do paciente vazios de modo a proteger a identidade dos pacientes. Como esta informação é importante para inferir a ontologia, o primeiro módulo do sistema preenche automaticamente esses campos. Para o campo *Nº de Processo* é

atribuído um valor que é incrementado automaticamente, e em relação aos campos *Nome*, *Idade* e *Data de Exame* são atribuídos dados aleatórios.

Grande parte da extracção da informação ocorre no segundo módulo, com a criação de uma base de dados que permite separar toda a informação que cada relatório possui. Este processo vem facilitar o processamento dos dados de cada relatório e o respectivo povoamento da ontologia. A base de dados possui os campos referidos na secção 3.1, e o seu preenchimento é relativamente fácil, pois a informação que se necessita extrair dos relatórios possui *tags* de identificação.

Com a informação dos relatórios separada na base de dados, cabe ao sistema no terceiro e último módulo, processar essa informação de modo a criar as instâncias e a povoar a ontologia. Verifica-se, que exceptuando os campos com o conteúdo dos exames e da conclusão, toda a informação contida nos outros campos da base de dados permite praticamente criar as instâncias na ontologia. A criação de instâncias na ontologia ocorre utilizando o *framework* Jena[4], que permite manipular e inferir ontologias que possuam as especificações da Web Semântica.

Em relação ao conteúdo dos exames e da conclusão, a informação necessita de ser processada porque corresponde a um texto com um conjunto de diagnósticos, observações, entidades anatómicas, etc, relacionadas entre si. Nesta fase do trabalho, foram aplicadas algumas técnicas de Processamento de Linguagem Natural (PLN) e de Aprendizagem Automática. O processamento do texto é executado parágrafo a parágrafo, onde cada parágrafo:

- É transformado em minúsculas;
- São eliminados os acentos, cedilhados e caracteres especiais;
- É processado por uma lista de *stopwords* que possui palavras a eliminar.

Depois de passar por este processo, o parágrafo está pronto a ser analisado. Devido à especificidade do vocabulário usado no texto, onde por exemplo, na construção das frases a utilização de verbos e adjectivos é praticamente inexistente, recorreu-se às árvores de decisão para conseguir processar o texto e classificar cada termo.

A figura 4, corresponde a uma parte da árvore de decisão.

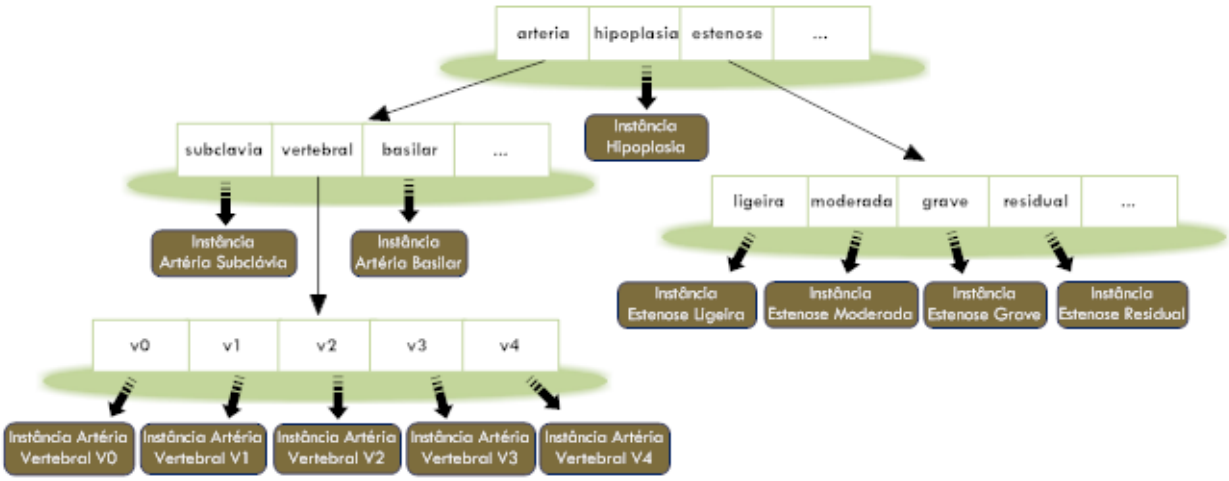


Figura 4: Árvore de Decisão

[4] `http://jena.sourceforge.net/`

Para construir a árvore de decisão, fez-se algumas adaptações em relação aos algorítmos tradicionais de modo a simplificar a classificação. Assim, cada nó de decisão é composto um por conjunto de termos e cada termo tem apenas uma ramificação possivel, que aponta para o próximo nó de decisão ou para a folha da árvore que corresponde a uma instância da classe do termo classificado.

O parágrafo é deste modo processado, e o resultado consiste num vector de instâncias. Por exemplo, o parágrafo com a frase **"Eixos Carotídeos, Artérias Subclávias e Artérias Vertebrais (V0 a V3), permeáveis, com paredes arteriais e fluxos normais."** depois de ser processado, iria obter o vector da figura 5.

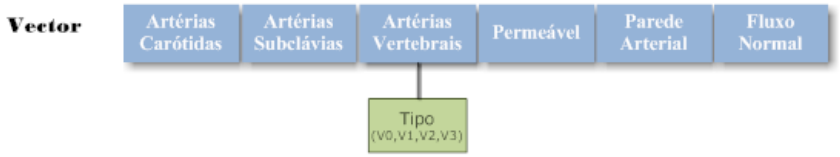


Figura 5: Vector com as Instâncias do Parágrafo

A este vector são aplicados alguns algoritmos que foram desenvolvidos e que permitem relacionar as instâncias, como por exemplo, entidades anatómicas com observações, observações com diagnósticos, etc. Por fim, e recursivamente, é possível criar as instâncias na ontologia para o parágrafo analisado. As instâncias criadas e que correspondem a termos médicos possuem a sua informação padronizada, através do UMLS[5] *(Unified Medical Language System)* O UMLS é um sistema, que se destina a ajudar os profissionais de saúde e investigadores biomédicos a utilizar informações de diversas fontes através do mapeamento de milhares de terminologias, ultrapassando o problema da semântica da área médica [9].

3.4 Aplicação e Resultados

Para facilitar todo o processo de inserção dos relatórios no *corpus* e a obtenção do respectivo ficheiro com o código OWL da ontologia, foi desenvolvida uma aplicação, que corresponde a uma interface desenvolvida em Java.

O utilizador tem assim apenas de introduzir os relatórios que quer converter e escolher qual o local onde o sistema irá guardar o ficheiro OWL. A figura 6, corresponde a um exemplo da utilização da aplicação.

Após a conversão dos relatórios do *corpus* foram realizados alguns testes. Para tal, recorreu-se à ferramenta Protégé onde foram analisadas todas as instâncias criadas para cada um dos treze relatórios que formam o *corpus*, e que segundo os profissionais de saúde do laboratório LUSCAN possuem praticamente todo o vocabulário que utilizam neste tipo de relatório.

As instâncias criadas na ontologia estão estruturadas de acordo com o texto presente nos relatórios. Verifica-se ainda que toda a informação que se pretendia extrair encontra-se instanciada na ontologia, não tendo sido detectados quaisquer erros. É portanto expectável, que para um conjunto maior de relatórios neurovasculares o sistema consiga extrair correctamente a informação e povoar a ontologia.

Os resultados apresentados são assim bastante positivos e vão ao encontro dos objectivos inicialmente propostos.

[5] `http://nlm.nih.gov/research/umls/`

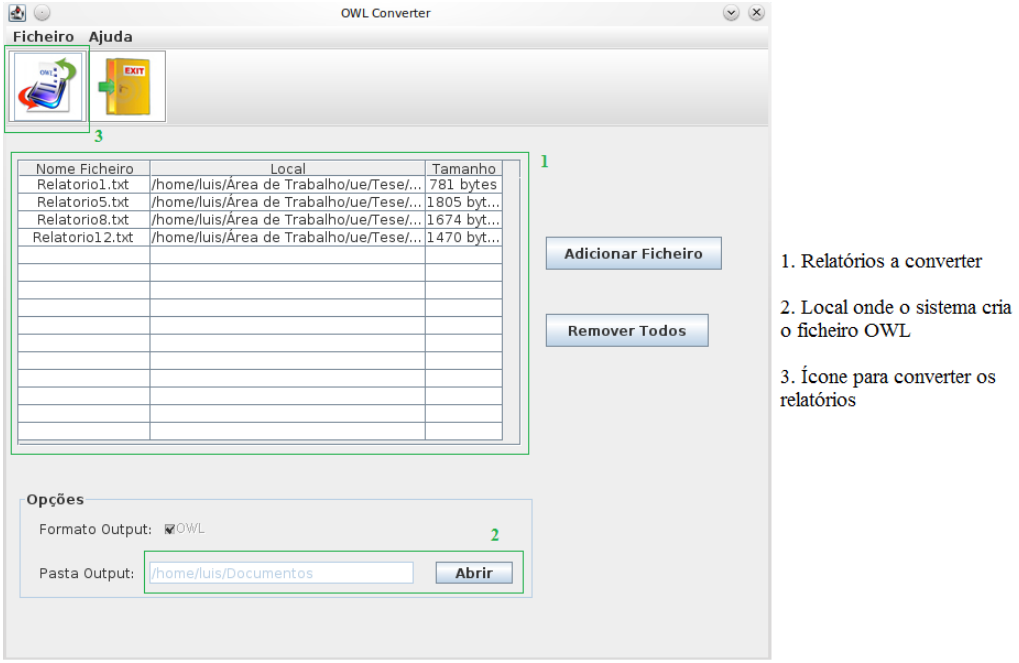


Figura 6: Exemplo da Utilização da Aplicação

4 Conclusão e Trabalho Futuro

Este trabalho abordou a problemática da criação de ontologias como forma de adquirir um conhecimento estruturado de relatórios médicos, neste caso de relatórios neurovasculares, que possuem textos em língua natural. Para tal, foi proposta uma metodologia como modo de atingir os objectivos.

A ontologia foi criada para um domínio específico que corresponde aos relatórios neurovasculares do *corpus*. Os resultados obtidos permitem afirmar que é possível utilizar ontologias em relatórios médicos, trazendo consequentemente várias vantagens principalmente através da criação de sistemas que processem automaticamente a informação dos relatórios.

A aplicação desenvolvida é de fácil utilização, permitindo que os utilizadores convertam os relatórios neurovasculares do formato digital para o formato OWL, sem terem de possuir grandes conhecimentos de informática.

Como trabalho futuro, pretende-se que o sistema de extracção de informação consiga extrair a informação de qualquer relatório neurovascular, independentemente da forma como a informação esteja disposta nos relatórios. Além disso, para permitir que o domínio da ontologia cubra sempre toda a informação, principalmente dos termos médicos, possibilitar que o sistema consulte fontes externas tais como a Wikipédia e o UMLS, expandindo automaticamente a ontologia.

Após a identificação do termo médico, para se expandir a ontologia terá de se acrescentar a classe correspondente a esse termo à respectiva hierarquia de classes da ontologia e adicionar esse termo à árvore de decisão para que esta o possa classificar.

Referências

[1] Luana P. Annibal, Natalia A. Rosa, Paulo M. A. Marques, José A. Baranauskas, and Joaquim C. Felipe. Uma ontologia para estruturação da informação contida em laudos radiológicos. In *XI Congresso Brasileiro de*

Informática em Saúde, Campos do Jordão, São Paulo, 2008.

[2] T. Berners-lee, J. Hender, and O. Lassila. The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*, 284 (5):34–43, May 2001.

[3] Jaime Cantais, David Dominguez, Valeria Gigante, Loredana Laera, and Valentina Tamma. An example of food ontology for diabetes control. In *Proceedings of the International Semantic Web Conference 2005 workshop on Ontology Patterns for the Semantic Web*, 2005.

[4] Renan Figueredo Farias, Merisandra Côrtes de Mattos, and Priscyla Waleska Targino de Azevedo Simões. *Ontologia para a Gestão do Conhecimento em Saúde por Meio da Metodologia Methontology*. Brasil, 2006.

[5] Ronen Feldman and James Sanger. *The Text Mining Handbook – Advanced Approaches in Analyzing Unstructured Data*. Cambridge, Cambridge University Press, 2007.

[6] Liliana Ferreira, César Telmo Oliveira, António Teixeira, and João Paulo Silva Cunha. *Extracção de Informação de Relatórios Médicos*. Universidade de Aveiro, 2008.

[7] Fabiane Bizinella Nardon. *Compartilhamento de Conhecimento em Saúde Utilizando Ontologias e Banco de Dados Dedutivos*. Escola Politécnica da Universidade de São Paulo, São Paulo, 2003.

[8] N. Noy and D. McGuinness. *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

[9] P. Quaresma, C. Bratsas, P. Bamidis, G. Pangalos, and N. Maglaveras. Knowledge management for medical computational problem solving: An ontological approach. In *2nd Balkan Conference in Informatics*, Ohrid, Macedonia, 2005.

[10] P. Quaresma, Ch. Bratsas, and N. Maglaveras. A framework to describe problems and algorithms in medical informatics via ontologies. In *4th European Symposium on Biomedical Engineering*, Patras, Greece. University of Patras, 2004.

A Relatório Neurovascular

Laboratório de Ultrassons Cardio e Neurovascular

RELATÓRIO DE NEUROVASCULAR

IDENTIFICAÇÃO DO PACIENTE:

Nº de Processo: 111

Data de exame: 2009-05-17

Nome: Luís Clérigo

Idade: 66

Serviço: UAVC

INFORMAÇÃO CLÍNICA: AVC

TRIPLEX SCAN CERVICAL realizado em Ecografo IE33/HD11 XE da Philips e transdutor L3-12Mhz, revela:

Eixos Carotídeos, Artérias Subclávias e Artérias Vertebrais (V0 a V3) permeáveis, com paredes arteriais e fluxos normais.

TRIPLEX SCAN TRANSCRANEANO (Via Transtemporal e Suboccipital) realizado em Ecografo IE33/HD11 XE da Philips e transdutor S1-5MHz, revela:

Permeabilidade de todos os segmentos arteriais estudados (ACM, ACI, ACA, ACP1, ACP2, AV4 e AB), com fluxos normais.

CONCLUSÃO:

Exames normais.

Técnica: Sónia Mateus/Vanda Pós-de-Mina

Médica: Irene Mendes

Extracção de informação aplicada ao domínio do Turismo

Nuno Miranda, Ricardo Raminhos, Pedro Seabra
VIATECLA SA
nmiranda@viatecla.com, rraminhos@viatecla.com, pseabra@viatecla.com

Teresa Gonçalves, José Saias, Paulo Quaresma
Universidade de Évora
tcg@uevora.pt, jsaias@uevora.pt, pq@uevora.pt

Resumo

As descrições de produtos de Turismo são fortemente apoiadas em expressões de língua natural. A selecção de oferta, de acordo com as necessidades do tipo de utilizador, depende muito da forma como estes produtos são comunicados. Uma vez que não existe interacção humana na apresentação *online* de produtos turísticos, a forma como estes são apresentados, mesmo quando se utiliza apenas a informação textual, é um factor chave de sucesso para sítios *web* de turismo. Devido à grande quantidade de oferta turística e à alta dinâmica deste sector, a gestão manual de dados não é escalável nem confiável. Este trabalho apresenta um protótipo desenvolvido para a extracção automática de informação relevante de turismo em textos de língua natural. O conhecimento adquirido é representado num formato normalizado e são produzidas novas descrições textuais de acordo com o perfil do utilizador para os canais de comercialização disponíveis. Este protótipo opera sobre a extracção de informação em descrições de hotéis e utiliza dados operacionais reais existentes na plataforma de turismo KEYforTravel.

1 Introdução

A presença *online* de sítios *web* relacionados com o turismo tem acompanhado o crescimento exponencial da Internet. Como exemplo, as receitas do turismo *online* dos EUA entre 2001 e 2004 aumentaram em média 29% ao ano [5]. Uma parte importante dessas receitas, correspondem a sítios comerciais (por exemplo, 'Expedia') onde os utilizadores podem navegar e pesquisar ofertas de turismo, inspeccionar os detalhes e efectivar reservas.

Hoje em dia, mesmo com a presença crescente de conteúdos multimédia na *web*, as descrições textuais em língua natural continuam, de longe, o formato mais utilizado em *e-marketing* e promoção aplicados aos produtos do turismo (hotel, aviação, *rent-a-car* e pacotes de férias). As descrições de hotéis consistem principalmente em enumerações simples de serviços e equipamentos e são fornecidas por conectores de serviços externos (por exemplo, 'GTA' ou 'HotelBeds' para produtos de hotel). Normalmente essas descrições textuais são estruturadas de forma diferente (complementar ou sobreposta) consoante o conector e são são apresentadas directamente ao utilizador final sem preparação prévia.

Este trabalho apresenta um protótipo desenvolvido para a extracção automática de informação relevante de turismo descrita em língua natural. Esta extracção permite a criação de descrições adequadas ao perfil de utilizador de modo a proporcionar uma segmentação correcta do *e-marketing* e processo de promoção. Este protótipo foca o subconjunto de descrições de hotel e utiliza dados operacionais reais obtidos a partir plataforma de turismo KEYforTravel [21].

O artigo está organizado da seguinte forma: a Secção 2 apresenta o domínio da aplicação e a Secção3 introduz as tarefas de Extracção de Informação e de Classificação. A Secção 4 descreve a arquitectura do sistema e as experiências e avaliação são apresentadas na Secção 5. Finalmente, a Secção 6 apresenta algumas conclusões e aponta os possíveis trabalhos futuros.

2 Domínio da aplicação

As descrições de hotéis são commumente disponiblizadas em textos de língua natural e incluem geralmente informação sobre os serviços do hotel, os equipamentos disponíveis nos quartos e a localização. A localização é normalmente descrita relativamente a um ou mais pontos de interesse conhecidos (por exemplo, rua, monumento ou estação de metro). Cada descrição resume (numa quantidade de texto mínima) os vários recursos e benefícios disponibilizados pelo hotel. Estes, juntamente com as condições de preço e disponibilidade, são responsáveis pela captação de clientes.

Apesar de toda a informação relevante estar contida na descrição, esta é utilizada principalmente para fins de apresentação não sendo potencializada para fins de pesquisa e aperfeiçoamento da oferta (por exemplo, pesquisar todos os hotéis com jacuzzi e piscina, situados perto de uma estação de metro em Londres).

Os hotéis que pretendem ter projecção de mercado forte estão normalmente presentes num ou mais agregadores de serviços de hotel. Os sítios comerciais de turismo podem utilizar vários desses agregadores para apresentar a oferta de hotel a nível mundial, o que resulta em possíveis múltiplas descrições para o mesmo hotel. Dependendo do foco de negócios do conector, algumas características do hotel podem ser consideradas mais ou menos relevantes, resultando em descrições distintas. Quando detectada esta duplicação, uma destas descrições é geralmente eleita passando a ser a única considerada para a apresentação do hotel (as outras descrições são descartadas, incluindo as informações complementares que não estão presentes na descrição eleita). Além disso, para sítios comerciais de turismo que disponibilizam oferta hoteleira a nível mundial (na ordem de vários milhares), não é possível gerir individualmente a descrição de cada hotel. Assim, muitos optam por apresentar directamente ao utilizador a descrição disponibilizada pelo conector o que resulta em três problemas principais:

- não há diferenciação entre os operadores de turismo *online* que partilham os mesmos conectores;
- as descrições não são orientadas para o utilizador / segmento do mercado;
- as descrições não são controladas nem normalizadas, apresentando desta forma descrições heterogéneas.

Ao construir um sistema que extrai todas as características relevantes do hotel e as normaliza é possível resolver estes problemas.

3 Extracção de Informação e Classificação de Textos

Extracção de Informação (EI) é um tipo de recuperação de informação [2] cujo objectivo consiste em extrair automaticamente informação estruturada de um determinado domínio (isto é, categorizada e semanticamente bem definida) a partir de documentos não estruturados.

A extracção de informação tem sido uma área de pesquisa activa, culminando numa série de conferências MUC[1] [6, 11] e, mais recentemente, em avaliações ACE [2] [12]. A detecção de informação relevante em textos de língua natural envolve, tipicamente, a desambiguação de frases (com base nas próprias palavras da frase) e do contexto em torno da informação candidata. As abordagens exploradas incluem regras definidas manualmente [7], aprendizagem de regras (por exemplo, [1, 19]) e outras abordagens de aprendizagem automática (por exemplo, [3]).

[1] *MUC: Message Understanding Conference*.
[2] *ACE: Automatic Content Extraction*.

3.1 Classificação de Textos

A Classificação Texto (CT) é uma tarefa bem estudada com muitas técnicas eficazes. Actualmente, os algoritmos mais populares e bem sucedidos para a classificação de texto são baseados em técnicas de aprendizagem automática. Têm sido aplicados diversos algoritmos, tais como árvores de decisão [20], análise discriminante linear e regressão logística [18], classificadores naïve de Bayes [14] e máquinas de vectores de suporte[3] [9].

Na classificação de textos, os documentos são pré-processados para obter uma representação mais estruturada que serve de entrada ao algoritmo de aprendizagem que na presença de um novo documento o classifica numa das classes semânticas previamente observadas. A abordagem mais comum é utilizar uma representação saco-de-palavras[4] [17] onde cada documento é representado pelas palavras que contém, sendo a ordem e pontuação ignoradas. Normalmente, as palavras são ponderadas por alguma medida da sua frequência no documento e, possivelmente, no corpus. Na maioria dos casos, um subconjunto de palavras (*stopwords*) não é considerado por não ter poder de discriminação sobre as diferentes classes, já que uma vez que o seu papel na frase está relacionado com a organização estrutural daquela. Alguns trabalhos utilizam um processo de radicalização ou lematização para reduzir os termos semanticamente relacionados.

Árvore de Decisão. Um classificador de texto representado por uma árvore de decisão é uma árvore na qual os nós internos representam palavras, os ramos representam a ponderação associada às palavras e as folhas representam as classes. Um documento é classificado verificando recursivamente (a partir da raíz) a palavra no nó e atravessando o ramo com a ponderação da palavra até chegar à folha; o documento pertence à classe rotulada pela folha.

A indução de uma árvore de decisão obtém-se aplicando uma estratégia de divisão e conquista. No seu modo mais básico, o algoritmo verifica se todos os exemplos têm a mesma etiqueta; caso não seja verdade, é selecciona uma palavra w e o conjunto de documentos é dividido em subconjuntos com a mesma ponderação para w, colocando cada subconjunto em subárvores distintas. Este processo é repetido recursivamente para cada uma das subárvores até que todas as folhas contenham apenas exemplos da mesma classe, sendo essa classe a escolhida como etiqueta do documento. A etapa chave do algoritmo é a escolha da palavra w que faz a partição dos documentos; esta escolha é geralmente feita de acordo com um critério de informação mútua ou entropia.

Classificador naïve Bayes. O classificador naïve Bayes é um classificador probabilístico que define a classe de um documento como aquela que maximiza a probabilidade do documento pertencer à classe, dados os seus atributos (palavras). Esta probabilidade é calculada aplicando o teorema de Bayes e assumindo que cada atributo é independente dos outros. Mesmo supondo ingenuamente a independência dos atributos, este algoritmo tem mostrado bons resultados na classificação de textos.

Máquinas de Vectores de Suporte. As máquinas de vectores de suporte são um algoritmo motivado pelos resultados teóricos da teoria de aprendizagem estatística e juntam uma técnica de núcleo com o modelo estrutural de minimização de risco[5]. As técnicas de núcleo têm duas componentes: um módulo que realiza um mapeamento do espaço de dados original para um espaço de características adequado e um algoritmo de aprendizagem desenhado para descobrir padrões lineares nesse (novo) espaço de características.

[3] Do inglês *Support Vector Machines*.

[4] Do inglês, *bag-of-words*.

[5] Do inglês, *structural risk minimization framework*.

A função de núcleo que implicitamente executa o mapeamento depende do tipo de dados e de conhecimento específico do domínio dos dados. Este algoritmo de aprendizagem é robusto e de âmbito geral. É também eficiente, pois a quantidade de recursos computacionais necessários é polinomial com o tamanho e o número de exemplos, mesmo quando a dimensão do espaço de características cresce exponencialmente.

Joachims [8] refere que a utilização de classificadores de texto baseados em máquinas de vectores de suporte constituem a primeira abordagem computacionalmente eficiente e robusta apresentando também, na prática, bons desempenhos.

4 Arquitectura do sistema

O sistema tem como objectivo receber uma descrição do hotel e produzir uma versão padronizada da mesma. Foi projectado utilizando uma estratégia de divisão e conquista, onde diversas pequenas ferramentas, que se concentram em problemas específicos e mais simples, são interligadas. Existem quatro ferramentas principais: o *Classificador de Frases*, o *Extractor de Informação*, o *Instanciador da Ontologia* e o *Tradutor Ontológico* que, quando integrados num *web service* permitem ao sistema estar disponível *online*. Esta arquitectura pode ser observada na Figura 1.

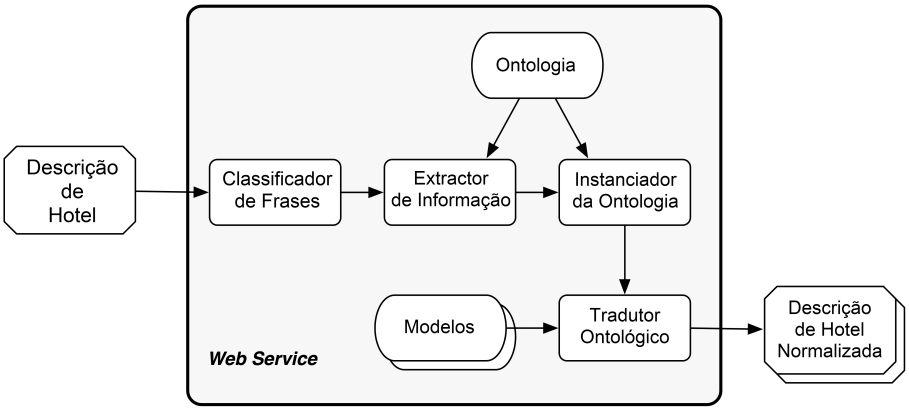


Figura 1: Diagrama da arquitectura do sistema

As próximas subsecções descrevem cada uma das ferramentas.

4.1 *Classificador de Frases*

O Classificador de Frases é responsável pela análise e classificação de blocos de texto em língua natural. Recebe a descrição do hotel e divide-a em frases que são examinadas individualmente e classificadas automaticamente num conjunto de classes predefinidas: `Equipamento`, `Serviço` e `Localização`. Cada frase pode pertencer a mais que uma classe, desde que contenha elementos dessas classes, ou a nenhuma, se não apresentar evidência. Esta classificação tem como objectivo filtrar as frases por tipo para assegurar um tratamento específico para cada uma pela ferramenta *Extractor de Informação*.

Abordagem seguida. O *Classificador de Frases* foi construído utilizando uma abordagem de aprendizagem automática. Como as frases podem pertencer a mais de uma classe trata-se de um problema de classificação de texto multi-etiqueta. Depois de efectuar um conjunto de experiências (apresentadas na Secção 5.2) os classificadores foram construídos utilizando o algoritmo naïve Bayes com as frases

representadas através de um saco-de-palavras com pesos binários para as palavras (indicam a presença ou não da palavra na frase).

Os classificadores foram construídos utilizando a ferramenta WEKA [23], uma aplicação que implementa um grande conjunto de algoritmos de aprendizagem automática, desenvolvida pela Universidade de Waikato (Nova Zelândia).

4.2 *Extractor de Informação*

Com as frases classificadas e agrupadas por tipo, o sistema tenta extrair informação útil. Este é o objectivo da ferramenta *Extractor de Informação*, que compreende duas etapas: encontrar a informação e lidar com erros de ortografia.

Como as descrições dos hotéis são fornecidas em língua natural sem qualquer padrão ou consistência, a mesma informação pode ser descrita não por um único termo, mas por um conjunto de sinónimos, ou mesmo abreviaturas; pode também existir informação incorrectamente escrita ou com falhas na acentuação. Esta última hipótese é muito comum pois as descrições originais são frequentemente traduzidas para diferentes idiomas desaparecendo os diacríticos regionais.

Abordagem seguida. Para encontrar a informação útil foi utilizada uma abordagem de correspondência de padrões conseguida através da definição de um conjunto de expressões regulares capazes de identificar sinónimos, abreviaturas e palavras 'quase' bem escritas (por exemplo `televisão`, `T.V.` e `TV` ou `mini-bar` e `minibar`). Os termos encontrados são depois substituídos por termos padrão que descrevem essa informação.

Para lidar com erros de ortografia, é medida a semelhança entre termos ainda não extraídos do texto e aqueles considerados relevantes utilizando a distância de Levenshtein [10].

4.3 *Instanciador de Ontologia*

Foi desenvolvida uma ontologia para o domínio dos hotéis com os objectivos de fornecer a estrutura básica de organização dos conceitos envolvidos e de guardar a informação extraída normalizada.

Embora na arquitectura actual as instâncias da ontologia constituam a entrada para o *Tradutor Ontológico*, este conhecimento normalizado (facilmente computável) pode ser aplicado para ampliar e melhorar substancialmente as capacidades de pesquisa no turismo. Os serviços, equipamentos e localização podem ser utilizados na consulta ou como parâmetros no processo de refinamento dos resultados de pesquisa. Além disso, uma vez que que o conhecimento está formalizado numa estrutura hierárquica, pode ser utilizado para mapear graficamente itens relacionados e estruturar a navegação.

Abordagem seguida. A ontologia para os hotéis foi concebida utilizando a linguagem OWL[6] [22], uma linguagem concebida para aplicações que necessitam processar o conteúdo de informação. Esta linguagem facilita a interpretabilidade de conteúdos *web* já que ao fornecer o vocabulário e a semântica formal do conhecimento representado permite, além de definir a estrutura do conteúdo, considerar possíveis relações semânticas entre os objectos e atributos.

Cada objecto da ontologia contém o conjunto de expressões regulares e funções Levenshtein utilizado pela ferramenta *Extractor de Informação*. Desta forma, aquele torna-se independente do problema específico em mãos. Utilizando a ontologia, o *Instanciador de Ontologia* gera uma instância OWL preenchida com as entidades extraídas e os seus atributos e relações semânticas. Esta instância torna-se então acessível através da utilização do *framework* de *web* semântica Jena [4].

[6] *OWL: Web Ontology Language*.

4.4 *Tradutor Ontológico*

Esta ferramenta é responsável por apresentar a informação extraída de forma atraente e de fácil leitura pelo utilizador final, permitindo diferentes tipos de textos de acordo com o público-alvo (por exemplo, clientes corporativos vs. lazer) ou a preferência do operador turístico.

Abordagem seguida. O *Tradutor Ontológico* utiliza um modelo XML que fornece o esqueleto para a informação final e preenche-o com a informação extraída.

5 Experiências e Avaliação

Esta secção introduz o conjunto de dados, descreve a configuração experimental e apresenta os resultados obtidos no desenvolvimento do *Classificador de Frases*. O sistema completo é depois apresentado através de um exemplo de descrição de hotel.

5.1 Descrição do conjunto

Para criar o *Classificador de Frases* foram extraídas aleatoriamente 330 frases de descrições de hotéis residentes na plataforma KEYforTravel [21]. Este conjunto de dados possui um total de 4271 palavras e uma média de 13 palavras por frase. As frases foram classificadas nas categorias semânticas de `Serviço`, `Equipamento` e `Localização`, obtendo conjuntos com 78, 56 e 96 frases, respectivamente.

5.2 Configuração experimental

Como já referido (ver Secção 4.1), e uma vez que algumas frases pertencem a mais do que uma das categorias possíveis (e algumas frases a nenhuma categoria), construiu-se um cenário de classificação multi-etiqueta.

Durante o desenvolvimento do *Classificador de Frases* fizeram-se várias experiências numa representação saco-de-palavras com diferentes ponderações de termos e diferentes algoritmos de classificação. Os melhores desempenhos foram conseguidos com:

- as representações binária (termo presente ou não), contagem de termos e medida *tfidf* [17] normalizada à unidade;
- os algoritmos árvores de decisão [16], naïve de Bayes [13] e máquinas de vectores de suporte (implementação SMO[7] [15]);

Os algoritmos foram executados com os seus parâmetros por omissão e os modelos foram avaliados com uma validação cruzada estratificada de 10 pastas [23]. O desempenho foi avaliado através das medidas precisão, cobertura e F_1 [17]. Foram realizados testes de significância com um nível de confiança de 95%.

5.3 Resultados

Nesta aplicação deu-se mais importância à cobertura em detrimento da precisão, já que os falsos positivos poderão ser tratados posteriormente pelo *Extractor de Informação* de cada classe. A Tabela 1 mostra a cobertura, precisão e medida F_1 para cada classe, ponderação e algoritmo.

[7] *SMO: Sequential Minimal Optimization*

		Serviço			Equipamento			Localização		
ponderação	classificador	cob	prec	f_1	cob	prec	f_1	cob	prec	f_1
binária	nBayes	.987	.795	.876	.969	.657	.776	.919	.703	.792
	SVM	*.816*	**.912**	.854	*.843*	**.951**	**.883**	*.724*	**.893**	.789
	árvore	*.601*	.863	*.696*	*.724*	**.961**	.802	*.589*	**.955**	.713
contagem	nBayes	*.844*	**.952**	.890	*.693*	**.990**	.800	*.510*	**.988**	*.658*
	SVM	*.830*	**.905**	.858	*.858*	**.951**	**.893**	*.711*	**.892**	.779
	árvore	*.608*	.858	*.699*	*.689*	**.954**	.775	*.623*	**.956**	.736
tfidf	nBayes	*.868*	**.969**	.911	*.749*	**.974**	.835	*.451*	**.960**	*.592*
	SVM	*.846*	**.904**	.866	*.865*	**.942**	**.893**	*.700*	**.891**	.771
	árvore	*.742*	.799	*.760*	*.716*	**.932**	.789	*.691*	**.919**	.773

Tabela 1: Cobertura, precisão, e F_1 das experiências realizadas. Os valores a negrito são significativamente superiores que a experiência `nBayes` com ponderação `binária`; os valores a itálico são significativamente inferiores.

A partir dos resultados é possível observar que a cobertura máxima é obtida pelo classificador naïve Bayes com ponderação binária, daí ter sido eleita a melhor experiência. Os valores a negrito são significativamente superiores a esta combinação e os valores a itálico são significativamente inferiores.

Através da tabela é também possível obervar que os valores de F_1 da experiência eleita (nBayes com ponderação binária) são apenas significativamente inferiores aos valores obtidos pelas máquinas de vectores de suporte para a classe `Equipamento`.

5.4 Avaliação do sistema

Definido o *Classificador de Frases* e tendo em conta a sua utilização futura, foram realizados vários testes com descrições de hotéis residentes na aplicação KeyforTravel. A Figura 2 mostra o exemplo de uma descrição do hotel original e Figura 3 apresenta o resultado da execução do sistema com três definições diferentes do *Tradutor de Ontologia*: uma corporativa, uma de lazer e uma de lazer na língua Inglesa.

A partir deste exemplo é possível observar que o sistema foi capaz de identificar a localização do hotel e todos os equipamentos e serviços disponíveis e gerar descrições com foco em diferentes conjuntos de atributos de acordo com o tipo de público-alvo (corporativo ou lazer). Por exemplo, enquanto a descrição corporativa apresenta uma descrição que foca o acesso `ADSL`, `sala de reuniões` e `ar condicionado`, a descrição de lazer refere a `TV satélite`, `piscina Olímpica` e `sauna`.

Descrição de entrada

O Tivoli Carvoeiro situa-se a 60 Km a Oeste do Aeroporto de Faro, na aldeia da Praia do Carvoeiro. Possui 293 quartos, Ar condicionado individual, TV satélite, telefone directo ao exterior, mini-bar, secador de cabelo, cofre e ADSL. O hotel dispõe de um café, uma piscina olimpica, Health Club com Sauna. Tem um parque Infantil. Também tem uma sala de reuniões equipada com ADSL.

Figura 2: Exemplo de uma descrição de hotel

Posteriormente ao desenvolvimento do protótipo, foi desenvolvido um módulo responsável por interligar o sistema a bases de dados de hotéis e processar sequelcialmente todas as descrições. Este módulo foi testado sobre uma base de dados da Viatecla onde se encontravam presentes vinte mil descrições

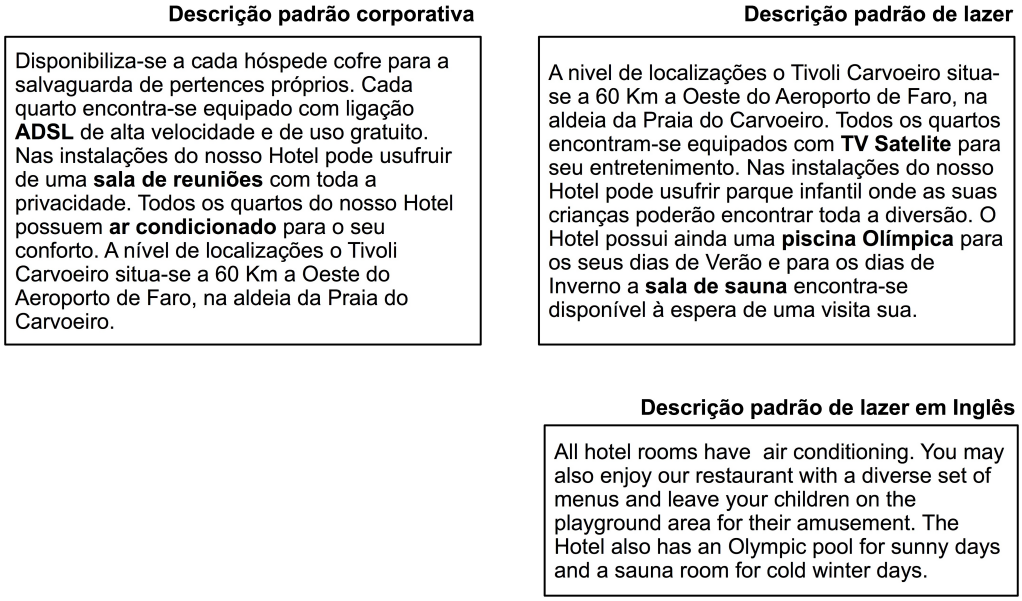


Figura 3: Três descrições distintas geradas pelo sistema

de hotéis. Após o processamento, foram extraídos aproximadamente 155 mil atributos resultando numa média de 8 atributos por descrição.

6 Conclusões e Trabalho Futuro

Este trabalho apresenta uma metodologia para extracção de informação útil a partir de descrições textuais de produtos turísticos e subsequente apresentação de descrições padrão. Este método foi aplicado ao domínio de descrições de hotel.

Os resultados mostram que o objectivo principal foi alcançado uma vez que é possível extrair informação útil, sistematizá-la e criar objectos computáveis a partir de descrições textuais simples. Relativamente à extracção da informação, esta abordagem mostrou-se capaz de detectar os tipos de descrição de hotel relevantes, ou seja, serviços de hotelaria, equipamentos e localização e, para cada um, extrair os recursos úteis que os descrevem.

A possibilidade de ter essa informação estruturada permite a criação de novos serviços de valor acrescentado tais como o refinamento de pesquisas de oferta. Além disso, permite a criação automática de descrições adequadas para diferentes segmentos de mercado antecipando novos passos rumo a uma diferenciação de clientes mais eficaz.

Relativamente a trabalho futuro, espera-se aumentar o desempenho global do sistema uma vez que há espaço para melhorias nos diversos módulos constituintes.

Pretende-se também abordar as áreas de suporte multi-língua (actualmente, apenas são suportadas descrições em Português) e normalização de outros produtos turísticos, tais como aluguer de viaturas e pacotes de férias.

Referências

[1] James S. Aitken. Learning information extraction rules: An inductive logic programming approach. In *ECAI*, pages 355–359, 2002.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[3] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110, 1999.

[4] HP Development. Jena – A Semantic Web Framework. *http://jena.sourceforge.net*, March 2010.

[5] J. Grau. Travel Agencies Online. eMarketer, 2005.

[6] R. Grishman. Information extraction: Techniques and challenges. In *SCIE'97, International Summer School on Information Extraction*, pages 10–27. Springer-Verlag, 1997.

[7] J. R. Hobbs, J. Bear, D. Israel, and M. Tyson. Fastus: A finite-state processor for information extraction from real-world text. In *IJCAI'93, Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1172–1178, August 1993.

[8] T. Joachims. *Learning to Classify Text Using Support Vector Machines*. Kluwer Academic Publishers, 2002.

[9] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *ICML'99, Proceedings of the Sixteenth International Conference on Machine Learning*, pages 200–209, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[10] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710, 1966. Originally publish in Russian.

[11] Linguistic Data Consortium. *MUC7, 7th Message Understanding Conference*. NIST, 1998.

[12] Alvin Martin Mark. Nist 2003 language recognition evaluation, 2003.

[13] Andrew McCallum and Kamal Nigam. A comparison of event models for naive bayes text classification. In *AAAI'98, Workshop on Learning for Text categorization*, pages 41–48. AAAI Press, 1998.

[14] Dunja Mladenic and Marko Grobelnik. Feature selection for unbalanced class distribution and naive bayes. In *ICML'99, Proceedings of the 16th International Conference on Machine Learning*, pages 258–267. Morgan Kaufmann Publishers, 1999.

[15] J. Platt. Fast Training of Support Vector Machines using Sequential Minimal Optimization. In B. Schölkopf, C. Burges, , and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 185–208. MIT Press, 1999.

[16] R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[17] G. Salton, A. Wang, and C. Yang. A vector space model for information retrieval. *Journal of the American Society for Information Retrieval*, 18:613–620, 1975.

[18] H. Schütze, D. Hull, and J. Pedersen. A comparison of classifiers and document representations for the routing problem. In *SIGIR-95, 18th ACM International Conference on Research and Developement in Information Retrieval*, pages 229–237, Seattle, US, 1995.

[19] S. Soderland, C. Cardie, and R. Mooney. Learning information extraction rules for semi-structured and free text. In *Machine Learning*, pages 233–272, 1999.

[20] Richard M. Tong and Lee A. Appelbaum. Machine learning for knowledge-based document routing (a report on the trec-2 experiment). In *TREC*, pages 253–264, 1993.

[21] ViaTecla. KEYforTravel platform. *http://www.keyfortravel.com*, March 2010.

[22] W3C. OWL Web Ontology Language Guide. *http://www.w3.org/TR/owl-guide*, March 2010.

[23] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, US, 2nd edition, 2005.

Extracção de Informação de Documentos em Língua Portuguesa: aplicação a domínios de anúncios

José L. Pedras
Universidade de Évora
Évora, Portugal

Paulo Quaresma
Universidade de Évora
Évora, Portugal

Resumo

Com o aumento exponencial de informação disponível na Internet e devido ao facto da maioria desta estar num formato não estruturado, surgiu o conceito de Extracção de Informação cujo principal objectivo consiste na transformação da informação desorganizada e não estruturada num formato adequado aos sistemas informáticos. Este trabalho incide sobre a Extracção de Informação de Documentos, mais precisamente na língua Portuguesa, sobre os quais é desenvolvido um sistema de extracção baseado em regras e padrões, e são realizados testes comparativos entre o sistema e os principais métodos de aprendizagem automática (*Hidden Markov Model*, *Hidden Semi-Markov Model*, *Maximum Entropy Markov Model*, *Conditional Random Fields* e *Support Vector Machines*). O domínio utilizado é na área dos anúncios de venda de automóveis, cujos resultados obtidos são em média superiores a 90% para o sistema desenvolvido. Numa segunda fase são efectuados vários testes com conjuntos de documentos de diferentes dimensões no domínio dos anúncios de venda de casas, utilizando métodos de aprendizagem automática.

1 Introdução

Com a explosão da popularidade e crescimento a larga escala da Internet, também a quantidade de Informação disponível aumentou exponencialmente. Vinte e cinco mil milhões de páginas Web são o valor estimado de endereços indexados pelos principais motores de pesquisa no ano de 2009 [1]. No entanto, apesar desta quantidade gigantesca de fontes de Informação disponíveis na Web, as mesmas sofrem de heterogeneidade e de falta de estrutura. Deste modo, o único método de acesso a tamanha quantidade de dados resume-se à navegação e pesquisa. A Extracção de Informação é um tipo de pesquisa documental que tem como principal objectivo a extracção de informação não estruturada a partir de fontes documentais e a posterior reestruturação desta em entidades, relações entre entidades e atributos descritores de entidades. Este método permite uma melhor pesquisa do que a procura por palavras-chave num conjunto não organizado. A estruturação da informação é um processo complexo que pode ser muito difícil de concretizar e é um desafio para muitas equipas de investigadores de todo o mundo que há duas décadas se dedicam à Extracção de Informação de Documentos.

Inicialmente surgiram os métodos baseados em padrões e regras, que tal como o nome indica, utilizam uma base de conhecimento ou um conjunto de regras rígidas e padrões na Extracção de Informação de Documentos. Com o aumento da capacidade de processamento dos sistemas informáticos, foram desenvolvidos vários algoritmos de aprendizagem automática que na última década foram a principal área de estudo na Extracção de Informação.

2 Trabalho Relacionado

Os métodos de Extracção de Informação assentes em regras evoluíram ao longo do tempo. Os primeiros métodos de extracção eram baseados em dicionários (padrões) aos quais se seguiram a utilização de um conjunto de regras rígidas codificadas manualmente. Como a inserção de regras era uma tarefa muito demorada e tediosa desenvolveram-se algoritmos de aprendizagem automática que se ocupavam de criar as regras a partir de exemplos. Um sistema de extracção baseado em regras é essencialmente

constituído por duas partes que compõem o método. A primeira parte é o conjunto de regras que vão ser aplicadas. A segunda parte do método é o sistema de controlo das regras existentes e definidas na primeira parte. Este sistema de controlo pode ser constituído de várias formas, tais como: o uso desordenado das regras e sobre as quais existe um sistema que resolve os conflitos quando existem incertezas; o uso de prioridades (ordenação) nas regras. O modelo de extracção baseado em regras tem como principal vantagem a possibilidade de consolidar ou aumentar o conjunto de regras de modo a obter melhores resultados. Ao longo dos anos desenvolveram-se vários sistemas baseados neste método, dos quais se destacam: ferramenta *FASTUS* que efectua a EI a partir de um sistema de regras predefinidas [2]; sistema de criação de regras a partir de textos anotados denominada *CRYSTAL* [3]; sistema de criação e aprendizagem de regras a partir de documentos semi-estruturados e não estruturados denominado *WHISK* [4]; desenvolvimento de um sistema de criação de regras a partir de exemplos denominado $(LP)^2$ [5]; ferramenta de EI que utiliza um conjunto de regras declarativas cuja sintaxe é semelhante ao *SQL* [6].

Os métodos baseados em aprendizagem automática surgiram como forma de ultrapassar as debilidades do uso dos sistemas de regras no que diz respeito à dependência do domínio e à dificuldade no aumento do conjunto de regras. Ao longo dos anos, devido ao aumento do poder computacional, foram desenvolvidos vários métodos de aprendizagem automática. O *Hidden Markov Model (HMM)* cujo modelo clássico foi publicado no final da década de 60 [7] e que ao contrário dos *Markov Models* em que são conhecidos os estados e os únicos parâmetros são as probabilidades de transição de estados, no HMM os estados não são directamente visíveis e apenas são conhecidos os resultados de saída e não a sequência de estados que levou a essa saída. Na Extracção de Informação foi estudada a sua aplicação e foram criados vários sistemas, dos quais se evidenciam: o sistema *Nymble* [8] que utiliza o modelo HMM na EI de nomes e outras entidades; o sistema de EI de restaurantes (nome, telefone e horário de funcionamento) [9] a partir de ficheiros HTML; o programa de EI de anúncios de emprego [10]. O *Hidden Semi-Markov Model (HSMM)* é visto como uma extensão ao modelo definido anteriormente, o *Hidden Markov Model*. A principal diferença entre os dois modelos reside no facto do HSMM definir-se pelo uso de uma cadeia semi-escondida de Markov assim como pelo facto de cada estado ter uma duração variável. Outra diferença importante é o facto de no HMM ser assumida uma observação por estado enquanto no HSMM cada estado pode emitir uma ou várias observações sequenciais. A sua aplicação na Extracção de Informação reside essencialmente no reconhecimento de texto impresso e manuscrito [12]. O *Maximum Entropy Markov Model (MEMM)*, também denominado de *Conditional Markov Model* é visto como uma extensão ao modelo *Hidden Markov Model*. A principal diferença entre os dois modelos reside no facto de o MEMM utilizar o conceito de maximização da entropia e das observações dependerem do estado actual e do estado anterior. Foi utilizado com sucesso nos estudos/sistemas: EI de artigos da *Usenet*, nomeadamente do cabeçalho, corpo, e perguntas-respostas [16]; reconhecimento de nomes de pessoas em textos antigos de fontes em mau estado [18] utilizando o algoritmo MEMM, entre outros. Os *Conditional Random Fields* foram desenvolvidos como uma melhoria às debilidades existentes nos algoritmos *Hidden Markov Model* e *Maximum Entropy Markov Model* no que diz respeito ao problema de *label bias* (as transições a partir de um estado competem apenas contra as desse estado ao invés de todas as transições existentes no modelo) [19]. Devido ao facto do CRF apresentar vantagens em relação aos métodos baseados em cadeias de *Markov*, foi utilizado com sucesso por vários investigadores na área da EI, dos quais se destacam os seguintes trabalhos: EI de contactos e redes sociais a partir de documentos de correio electrónico e Web [20]; sistema de EI de entidades a partir de artigos científicos [21]. O *Support Vector Machines* [23], definido como um classificador é utilizado na EI na medida em que a sua aplicação permite analisar dados e reconhecer padrões. Como o SVM é um classificador, dado um conjunto de treino anotado com categorias, ele constrói um modelo e aplica-o a novos dados (não treinados) identificando a que categoria pertencem. Devido ao facto dos resultados obtidos serem bastante favoráveis é utilizado por alguns sistemas desenvolvidos, tais como: EI de entidades utilizando o algoritmo SVM com várias variações de modo a obter a melhor relação extracção/tempo [24];

desenvolvimento de uma ferramenta para EI de entidades de um conjunto de artigos científicos [26].

3 Arquitectura Proposta

3.1 Programa ExtrAuto

Para a realização da Extracção de Informação de anúncios de automóveis desenvolveu-se uma aplicação baseada num sistema de regras de nome ExtrAuto. Esta aplicação é implementada na linguagem Java e permite realizar a extracção de informação das entidades usuais presentes no domínio dos anúncios de venda de automóveis.

Antes da descrição do sistema de EI é necessário evidenciar o funcionamento da base de conhecimento que serve de suporte ao programa. A BD é constituída por vários ficheiros com informação relacionada com o domínio em estudo (anúncios de venda de automóveis). Um dos ficheiros constituintes é a lista de vocábulos admitidos como nomes próprios na nação Portuguesa, disponibilizada pelo Instituto dos Registos e do Notariado e presente no sítio de Internet da instituição em `http://www.irn.mj.pt`. O ficheiro principal da BD é o que identifica os grupos e entidades. Para cada um destes elementos contém as palavras que o descrevem no anúncio. Este ficheiro está definido de forma a facilitar a introdução de novas palavras no sistema, possibilitando o refinamento da EI. O formato definido neste ficheiro é definido pela expressão `[grupo] | [entidade] > [palavra 1] # ... # [palavra x] #` em que a cada entidade corresponde uma linha. A primeira fase da tarefa de EI do programa ExtrAuto é definida pelo uso de um analisador LALR para extracção de entidades. O uso de um analisador sintáctico tem como principal função a extracção das entidades preço, localização, data do anúncio e número do anúncio. Para além das entidades obrigatórias é também possível extrair mais algumas entidades com esta ferramenta dependendo da sua presença e localização nos documentos do CD [1]. Ou seja, o analisador LALR desenvolvido está apto a extrair a marca, o modelo, o ano do veículo, a cor, o número de quilómetros e a potência. A segunda fase da tarefa de EI consiste na extracção das entidades pertencentes ao grupo dos contactos. As entidades que pertencem a este grupo são: os números de telefone e telemóvel; os endereços de correio electrónico; os endereços da Internet presentes no anúncio. Na terceira e última fase são tratados os restantes grupos de entidades (equipamento e informações). Para utilizar a pesquisa de informação na BD de uma ou mais palavras foi criado um método de pesquisa em janela. Quanto é atingido o limite (no domínio utilizado não existem conjuntos identificadores com mais de cinco palavras) e não foi identificada nenhuma possível entidade o programa avança para a próxima palavra. Se a BD inclui uma ou mais palavras que podem pertencer a uma entidade, são realizados diversos testes (procura no anúncio do valor a que corresponde a palavra identificada e validação desse valor).

3.2 Programa MinorThird

O programa MinorThird [28] define-se como uma ferramenta que permite realizar a categorização e extracção de documentos. Está disponível em `http://minorthird.sourceforge.net/` e apresenta vários modelos de Extracção de Informação de Documentos. Relativamente aos modelos de algoritmos de EI descritos no capítulo anterior, a ferramenta MinorThird implementa um conjunto de algoritmos de modo a obter a melhor relação tempo/qualidade de extracção. Assim, são implementados os seguintes algoritmos: VPHMM [29] que utiliza o *Voted Perceptron* para estimar os parâmetros do HMM; VPCMM, semelhante ao anterior mas para o algoritmo CMM; VPSMM e VPSMM2 [30] que utilizam o *Voted*

[1] Conjunto de Documentos

Perceptron para estimar os parâmetros do SMM; SVMCMM, que utiliza o modelo SVM para o algoritmo CMM; MEMM [16]; CRF [19, 31]; SemiCRF [32].

O método de funcionamento do programa para uma tarefa completa de EI (treino e teste) divide-se em duas fases: fase de treino (o conjunto de documentos previamente anotado é treinado para a entidade e algoritmo escolhidos e classificado de modo a obter um conjunto de dados anotados pelo programa) e fase de teste (é realizada a comparação de resultados entre o conjunto anotado e o conjunto que se pretende testar).

4 Resultados Experimentais

Durante o processo de Extracção de Informação é necessário efectuar a escolha das entidades que se pretendem extrair. Neste caso específico escolheram-se várias entidades, de acordo com o domínio em causa e o método de Extracção de Informação utilizado. No domínio dos anúncios de automóveis as entidades escolhidas dividem-se em dois grandes grupos de acordo com a técnica utilizada: os anúncios processados com o ExtrAuto têm um conjunto composto por 62 entidades; nos anúncios analisados pelo MinorThird estão definidas 6 entidades. O domínio dos anúncios de venda de casas apresenta um conjunto de entidades definidas para o programa MinorThird e é composto por 18 entidades.

4.1 Extracção de Informação em Anúncios de Automóveis

4.1.1 Extracção com o programa ExtrAuto

Nesta fase é analisado o nível de extracção do programa ExtrAuto através de um conjunto de testes realizados. Para a medição da precisão, abrangência e medida F, foi criado manualmente um conjunto de anotações do CD num formato interpretado pelo programa. A partir destas anotações e das entidades extraídas pelo programa foi escrita uma folha de cálculo e calculados os valores correspondentes. Para uma melhor apresentação dos resultados criaram-se dois grupos de entidades com as seguintes designações: grupo das informações (contém as entidades relativas à informação do veiculo e do anunciante); grupo dos equipamentos (contém as entidades dos equipamentos do veiculo).

A figura 1 apresenta os resultados obtidos para o grupo das informações, enquanto o grupo dos equipamentos é ilustrado pela figura 2.

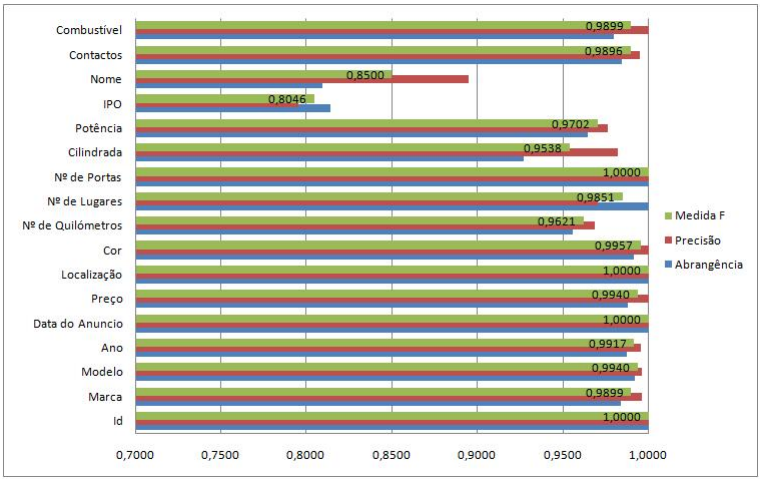


Figura 1: Gráfico dos resultados do grupo de informações dos anúncios automóveis

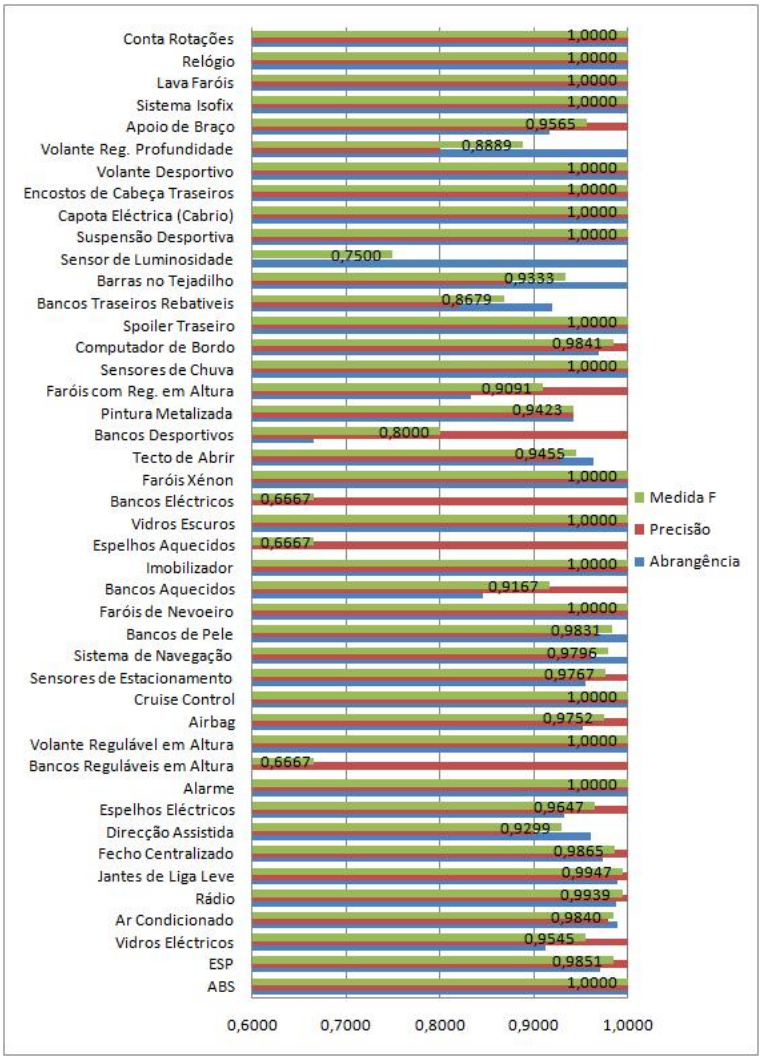


Figura 2: Gráfico dos resultados do grupo de equipamentos dos anúncios automóveis

4.1.2 Extracção com o programa MinorThird

Depois de realizada a extracção com o programa ExtrAuto é realizado um conjunto de testes com o programa MinorThird. Para a realização deste teste foram escolhidas 6 entidades do domínio dos automóveis, cujos critérios de escolha utilizados definem-se pelo grupo a que pertencem (informações e equipamentos) e pelo número de ocorrências no CD (baixa, média e alta).

Na realização deste teste foram escolhidos vários algoritmos de aprendizagem presentes no programa MinorThird. Assim, foram utilizados os seguintes algoritmos: VPHMM Learner; VPCMM Learner; CRF Annotator Learner; SVMCMM Learner; VPSMM Learner; VPSMM Learner 2; MEMM Learner. De notar que foram utilizadas todas as opções predefinidas nos algoritmos implementados no programa MinorThird. Na realização deste teste foi utilizada a validação cruzada com 5 pastas como método de estimação de resultados.

4.1.3 Comparação de resultados (ExtrAuto vs MinorThird)

Nesta secção são utilizadas as medidas F calculadas anteriormente para cada um dos programas da seguinte forma: No ExtraAuto é utilizada a medida calculada para cada uma das entidades; no MinorThird é utilizada a medida F calculada no teste de validação cruzada com 5 pastas. Os resultados obtidos nos testes anteriores estão presentes na tabela 1. De notar que no estudo com o MinotThird apenas foi utilizada a medida F do algoritmo com maior taxa de cada entidade.

Entidade	**MinorThird**		**ExtrAuto**
	Algoritmo	Medida F	Medida F
Marca	SVMCMMLearner	0,9452	0,9899
N.º de Lugares	SVMCMMLearner	0,7246	0,9851
Vidros Eléctricos	SVMCMMLearner	0,9167	0,9545
Bancos Rebatíveis	CRF Annotator Learner	0,9029	0,8679
Pintura Metalizada	SVMCMMLearner	0,8554	0,9423
Potência	SVMCMMLearner	0,9461	0,9702

Tabela 1: Resultados da extracção dos programas ExtrAuto e MinorThird

4.2 Extracção de Informação em Anúncios de Casas

Depois de concluído o conjunto de testes no domínio dos automóveis é apresentada uma nova fase de testes que incidem sobre um novo domínio, mais precisamente, o domínio dos anúncios de venda de casas. Na realização deste conjunto de testes foi utilizado o programa MinorThird de modo a aferir qual o melhor algoritmo de aprendizagem automática a usar e quais as relações entre o número de elementos a treinar e os valores da medida F obtidos. Neste estudo foram utilizadas as dezoito entidades definidas para este domínio e os seguintes algoritmos de extracção: VPHMM Learner; VPCMM Learner; CRF Annotator Learner; SVMCMM Learner; VPSMM Learner; VPSMM Learner 2; MEMM Learner. Para cada algoritmo testado são utilizadas as condições predefinidas do programa MinorThird. Quanto aos dois conjuntos de documentos em estudo, o primeiro conjunto é composto por 150 documentos e o segundo conjunto por 300 documentos dos quais metade são do primeiro conjunto. Foram usados os seguintes parâmetros para cada um dos conjuntos de documentos a estudo: no primeiro conjunto de documentos foi utilizada validação cruzada com 5 pastas como método de estimação de resultados; no segundo conjunto de documentos foi utilizada validação cruzada com 10 pastas. A escolha destes valores incide no facto de o número de documentos a teste no intervalo ser igual nos dois conjuntos, ou seja, em 150 documentos por 5 pastas, são atribuídos 30 documentos por intervalo, o mesmo valor para 300 documentos em 10 pastas. A tabela 2 ilustra os valores da medida F para o algoritmo com o resultado mais elevado de cada entidade estudada.

Entidade	**CD (150)**		**CD (300)**	
	Algoritmo	**Medida F**	**Algoritmo**	**Medida F**
Área Total	SVMCMM	0,6256	CRF	0,7385
Condomínio Fechado	SVMCMM	0,4286	CRF	0,8435
Área Cozinha	SVMCMM	0,6604	SVMCMM	0,8509
Equipamento Cozinha	SVMCMM	0,7963	CRF	0,7913
Disponível em	SVMCMM	1,0000	SVMCMM	1,0000
Estado	SVMCMM	0,6104	SVMCMM	0,6985
Garagem	SVMCMM	0,8649	SVMCMM	0,8963
Localização	CRF	0,7591	CRF	0,8249
Piso	SVMCMM	0,6069	SVMCMM	0,6543
Preço	SVMCMM	0,9528	SVMCMM	0,9664
Área Quarto	SVMCMM	0,6227	SVMCMM	0,8838
Equipamento Quarto	CRF	0,5559	CRF	0,6221
Área Sala	SVMCMM	0,7615	SVMCMM	0,7849
Equipamento Sala	SVMCMM	0,6702	SVMCMM	0,6310
Tamanho	SVMCMM	0,9630	CRF	0,9433
Tipo de Casa	CRF	0,9622	SVMCMM	0,9135
Área WC	SVMCMM	0,7737	CRF	0,7750
Equipamento WC	SVMCMM	0,5702	SVMCMM	0,7093

Tabela 2: Algoritmos com melhor resultado para as entidades analisadas no MinorThird

5 Conclusões

Para o programa ExtrAuto foram definidas um total de 62 entidades divididas em dois grupos principais (informações e equipamentos) de acordo com a temática a que pertencem. Para as 16 entidades do grupo das informações foram obtidos resultados na medida F superiores a 80% em todas as entidades. Em 14 das 16 obtiveram-se resultados superiores a 95% e em 8 dessas superiores a 99%. O grupo dos equipamentos, constituído por 44 entidades, apresenta resultados superiores a 95% em 32 entidades, entre 80% e 95% para 9 entidades e entre 65% e 80% para as 3 entidades restantes. Este conjunto de resultados evidencia a boa prestação da aplicação na Extracção de Informação de anúncios de venda de automóveis na medida em que a maioria apresenta resultados superiores a 95%. Para o conjunto de testes efectuados com a ferramenta MinorThird foram escolhidas 6 entidades das 62 definidas anteriormente. Na primeira fase o conjunto de testes visa apurar qual o algoritmo com maior valor na medida F para cada entidade. Na segunda fase é comparado esse resultado com o obtido pelo programa ExtrAuto para essa entidade. Os valores da medida F obtidos na primeira fase variam entre os 72% e 95% com 4 das 6 entidades a obterem valores acima dos 90%. Quanto aos algoritmos com melhor performance, em 5 das 6 entidades foi para o SVMCMM essa atribuição e na entidade restante para o CRF. Na comparação das medidas F entre as duas ferramentas o sistema ExtrAuto apresenta melhores resultados em 5 das 6 entidades, apresentando uma diferença média de 8% entre si e o MinorThird. De notar, que a ferramenta ExtrAuto apenas pode ser utilizada no domínio em causa e no caso da modificação deste são necessárias alterações profundas no sistema (dicionário e regras), o que poderá ser uma tarefa muito morosa. A ferramenta MinorThird pode utilizar um qualquer domínio desde que se procedam às correspondentes anotações nos conjuntos de documentos de treino. No segundo domínio em estudo, nos testes com os dois conjuntos de documentos verifica-se um aumento da medida F na grande maioria dos casos até

ser atingido um ponto de saturação, ou seja, quando os elementos utilizados como treino no primeiro conjunto de documentos são em quantidade e qualidade, os resultados obtidos com esse conjunto e com o segundo conjunto são muito semelhantes, existindo apenas diferenças residuais nos valores da medida F. Nos restantes casos existe uma variação absoluta positiva que é mais ou menos acentuada de acordo com a entidade em estudo e com o algoritmo utilizado. No que diz respeito aos algoritmos com melhores e piores resultados, o SVMCMM é o que obtém melhores resultados seguido do CRF. Os algoritmos HMM e MEMM são os que apresentam piores resultados, apresentando em muitos casos valores nulos para os dois conjuntos estudados. Relativamente a estes algoritmos verifica-se a influência da qualidade e quantidade do conjunto de treino, na medida em que para apresentarem bons resultados necessitam que esse conjunto tenha mais elementos e de melhor qualidade do que para os algoritmos CRF e SVMCMM. No entanto, quando o conjunto é completo os resultados obtidos são muito semelhantes aos obtidos pelos algoritmos com melhor performance.

6 Trabalho Futuro

Como trabalho futuro do sistema ExtrAuto pretende-se: a construção de um método de desambiguação para os casos em que existem dificuldades na atribuição de entidades (o uso de um sistema de prioridades é uma das várias soluções disponíveis para a resolução deste problema); o aumento do número de elementos presentes na base de conhecimento para as expressões que identificam os equipamentos/informações dos veículos. Quanto aos anúncios de venda de casas e ao conjunto de testes com a ferramenta MinorThird, e de modo a aumentar os valores da medida F obtidos: a utilização de grupos nas etiquetas XML nas entidades que no CD estão colocadas sequencialmente e separadas por virgulas; o *tuning* dos algoritmos existentes na ferramenta através da alteração dos vários parâmetros e opções existentes para cada algoritmo utilizado; a criação de um CD com informação em maior quantidade e qualidade para as entidades que apresentam piores resultados globais.

Referências

[1] WorldWideWebSize, Novembro 2009. `http://www.worldwidewebsize.com/`.

[2] Jerry R. Hobbs, John Bear, David Israel, and Mabry Tyson. Fastus: A finite-state processor for information extraction from real-world text. pages 1172–1178, 1993.

[3] Stephen Soderland, David Fisher, Jonathan Aseltine, and Wendy Lehnert. Crystal: Inducing a conceptual dictionary, 1995.

[4] Stephen Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34 (1-3):233–272, 1999.

[5] Fabio Ciravegna. Lp2 an adaptive algorithm for information extraction from web-related texts. In *In Proceedings of the IJCAI-2001 Workshop on Adaptive Text Extraction and Mining*, 2001.

[6] T. S. Jayram, Rajasekar Krishnamurthy, Sriram Raghavan, Shivakumar Vaithyanathan, and Huaiyu Zhu. Avatar information extraction system. *IEEE Data Engineering Bulletin*, 29:2006, 2006.

[7] L. E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probalistic functions of Markov processes and to a model for ecology. *Bull. Am. Math. Soc.*, 73:360–363, 1967.

[8] Daniel M. Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. Nymble: a high-performance learning name-finder. In *In Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 194–201, 1997.

[9] Nancy R. Zhang. Hidden markov models for information extraction, 2001.

[10] K. Au and K. Cheung. Information extraction for on-line job advertisements, 2004.

[11] Bo-Hyun Yun. Hmm-based korean named entity recognition for information extraction. In Zili Zhang and Jörg Siekmann, editors, *Knowledge Science, Engineering and Management*, volume 4798 of *Lecture Notes in Computer Science*, pages 526–531. Springer Berlin / Heidelberg, 2010.

[12] M. Y. Chen, A. Kundu, and S. N. Srihari. Variable duration hidden markov model and morphological segmentation for handwritten word recognition. *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*, pages 600–601, 1993.

[13] Reza Safabakhsh and Peyman Adibi. Nastaaligh handwritten word recognition using a continuous density variable-duration hmm. *The Arabian J. Science and Eng*, 30:95–118, 2005.

[14] Abdallah Benouareth, Abdellatif Ennaji, and Mokhtar Sellami. Hmms with explicit state duration applied to handwritten arabic word recognition. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*, pages 897–900, Washington, DC, USA, 2006. IEEE Computer Society.

[15] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review Online Archive (Prola)*, 106 (4):620–630, May 1957.

[16] Andrew Mccallum and Dayne Freitag. Maximum entropy markov models for information extraction and segmentation. pages 591–598. Morgan Kaufmann, 2000.

[17] Martin Jansche. Named entity extraction with conditional markov models and classifiers, 2002.

[18] Thomas Packer, Joshua Lutes, Aaron Stewart, David Embley, Eric Ringger, and Kevin Seppi. Extracting person names from diverse and noisy ocr text. Department of Computer Science Brigham Young University Provo, Utah, USA, 2010.

[19] John Lafferty. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. pages 282–289. Morgan Kaufmann, 2001.

[20] Aron Culotta, Ron Bekkerman, and Andrew Mccallum. Extracting social networks and contact information from email and the web. In *In CEAS-1*, 2004.

[21] Fuchun Peng and Andrew McCallum. Information extraction from research papers using conditional random fields. *Inf. Process. Manage.*, 42 (4):963–979, 2006.

[22] H. Dalianis and S. Velupillai. De-identifying swedish clinical text - refinement of a gold standard and experiments with conditional random fields, 2010.

[23] Corinna Cortes and Vladimir Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[24] Hideki Isozaki and Hideto Kazawa. Efficient support vector classifiers for named entity recognition. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[25] Jun'ichi Kazama, Takaki Makino, Yoshihiro Ohta, and Jun'ichi Tsujii. Tuning support vector machines for biomedical named entity recognition. In *In Proceedings of the ACL-02 Workshop on Natural Language Processing in the Biomedical Domain*, pages 1–8, 2002.

[26] Hui Han C. Lee Giles, Eren Manavoglu, Hongyuan Zha, Zhenyue Zhang, and Edward A. Fox. Automatic document metadata extraction using support vector machines. In *In JCDL 03: Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 37–48, 2003.

[27] Giorgio Lucarelli, Xenofon Vasilakos, and Ion Androutsopoulos. Named entity recognition in greek texts with an ensemble of svms and active learning, 2007.

[28] William Cohen. *MinorThird: Methods for Identifying Names and Ontological Relations in Text using Heuristics for Inducing Regularities from Data*. http://minorthird.sourceforge.net, 2004.

[29] Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. pages 1–8, 2002.

[30] William W. Cohen. Exploiting dictionaries in named entity extraction: Combining semi-markov extraction processes and data integration methods. In *Semi-Markov Extraction Processes and Data Integration Methods, Proceedings of KDD 2004*, pages 89–98, 2004.

[31] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. pages 213–220, 2003.

[32] Sunita Sarawagi and William W. Cohen. Semi-markov conditional random fields for information extraction. In *In Advances in Neural Information Processing Systems 17*, pages 1185–1192, 2004.

Algoritmo de Segmentação para a Detecção dos Olhos do Queijo Regional de Évora

Gaspar Brogueira
Universidade de Évora
gaspar_mrb@hotmail.com

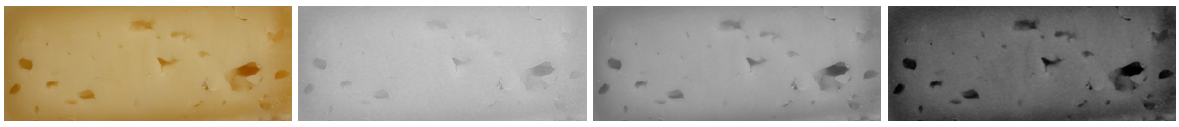
Luís Rato
Universidade de Évora
lmr@di.uevora.pt

Resumo

Em termos de cor, os olhos distinguem-se do fundo amarelado do queijo, pois apresentam uma cor mais escura, notando-se uma acentuada variação de cor nas fronteiras dessas regiões. Neste artigo, apresenta-se um algoritmo de segmentação que, por meio de técnicas de processamento e filtragem de sinais digitais, permite a detecção dos olhos do queijo mediante o processamento de imagens digitais.

1 Introdução

Os olhos ao apresentarem uma cor mais escura relativamente ao fundo do queijo, assumem uma cor acastanhada, o que se traduz numa diminuição do valor da componente azul do espaço de cores RGB. Portanto, a visualização da imagem de um queijo tendo apenas em consideração a componente azul do espaço RGB, permite uma melhor intuição sobre a localização dos olhos, como se pode observar na figura 1.



(a) Imagem original em RGB. (b) Vermelho. (c) Verde. (d) Azul.

Figura 1: Imagem visualizada em cada uma das três componentes de cor RGB.

A metodologia desenvolvida para a detecção dos olhos do queijo de Évora, pressupõe um processamento horizontal da imagem, ou seja, o processamento é apenas realizado sobre as linhas da imagem, a uma dimensão (1D). Esta opção deixa em aberto futuras melhorias no desempenho do algoritmo, nomeadamente no que diz respeito à paralelização do código. Por comparação com outros algoritmos, Limiar de Otsu, Limiar de Máxima Entropia e *K-Means Clustering*, demonstra-se que o algoritmo desenvolvido se adequa ao problema em causa, obtendo-se resultados significativamente melhores.

Na figura 2 é possível observar a variação de cor ao longo de uma linha da imagem da figura 1a, relativamente a cada uma das três componentes do espaço de cores RGB. De entre os três sinais presentes na figura 2, o sinal da componente azul apresenta maior variância, sendo esta a componente de cor que permite uma melhor detecção das regiões escuras da imagem.

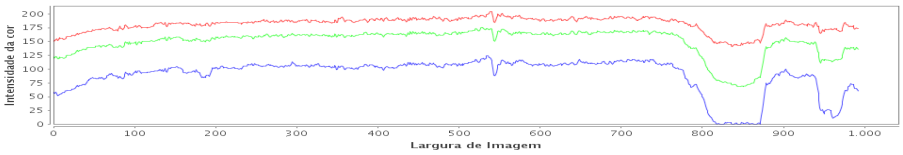


Figura 2: Variação de cada uma das componentes de cor para uma linha da imagem 1a.

Por forma a diminuir a presença de ruído e a detectar variações bruscas num sinal, foram desenhados e implementados filtros de média móvel, com características diferentes. O filtro de média móvel é provavelmente o filtro mais utilizado em processamento de sinal, visto ser bastante compreensível e fácil de implementar [4]. Este filtro, utilizado para a redução de ruído aleatório, realiza uma média móvel de K pontos do sinal de entrada, $x(n)$, para produzir cada ponto do sinal de saída, $y(n)$. A implementação, pela operação de convolução [1], resulta na equação 1.

$$y(n) = \frac{1}{K} \sum_{i=0}^{K-1} x(n+i) \tag{1}$$

O filtro de média móvel representado na equação 1, realiza a média dos pontos do sinal de entrada, compreendidos entre $x(n)$ e $x(n+K-1)$. No entanto, para a implementação de um filtro de média móvel, foi adoptada uma versão alternativa em que o conjunto de pontos do sinal de entrada é simétrico e centrado do ponto de saída, ou seja, a equação 1 toma a forma de:

$$y(n) = \frac{1}{K} \sum_{i=-\frac{K-1}{2}}^{\frac{K-1}{2}} x(n+i) \tag{2}$$

2 Algoritmo de Segmentação

O primeiro filtro de média móvel desenhado, tem como objectivo a redução de variações abruptas de cor, sobre regiões suficientemente grandes de modo a permitir uma aproximação à cor de fundo do queijo, e a despistar a presença de olhos no queijo. Este filtro tem uma janela de $K = W/6$ pontos, sendo W o número de pixels da largura da imagem. Foram realizados testes para diversos valores de K, tendo-se verificado que a relação entre o desempenho e o resultado do algoritmo em si, apresentava valores aceitáveis para $K = W/6$.

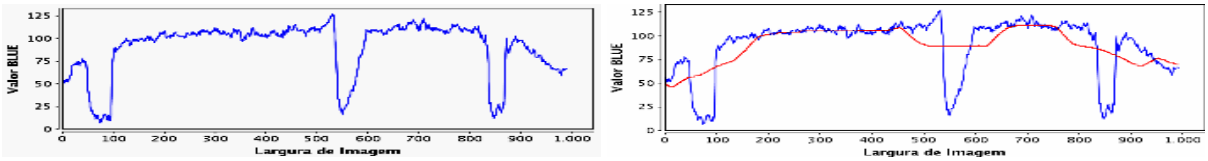
Todos os coeficientes do filtro são iguais, tendo soma unitária, pelo que cada coeficiente tem o valor de $1/K+1$. Assume-se que K é um valor par, caso contrário será considerado o próximo valor par, por excesso, relativamente a K. Este filtro será designado $H_{1/6}$, sendo definido pela equação 3.

$$y_a(n) = a_0 x(n+\frac{K}{2}) + a_1 x(n+\frac{K}{2}-1) + ... + a_{\frac{K}{2}} x(n) + ... + a_{-\frac{K}{2}+1} x(n-\frac{K}{2}+1) + a_{-\frac{K}{2}} x(n-\frac{K}{2}) \tag{3}$$

$$a(n) = \begin{cases} \frac{1}{K+1}, & se \quad -\frac{K}{2} \leq n \leq +\frac{K}{2} \\ 0, & c.c. \end{cases} \tag{4}$$

O filtro da equação 3, executa a convolução do sinal $x(n)$, valores da componente azul da imagem original, com os coeficientes dados pela equação 4. Este filtro produz um sinal mais suave comparativamente ao sinal original, reduzindo as variações bruscas de cor, permitindo uma aproximação do sinal à cor de fundo do queijo, tal como se verifica na figura 3.

Para a detecção dos olhos, será observada a diferença da cor de fundo, aproximada pelo sinal $y_a(n)$, para o sinal original $x(n)$. A presença de ruído no sinal $x(n)$, não permite que seja efectuada uma boa segmentação apenas com a subtracção de ambos os sinais e posterior comparação com um limiar. Filtrando o sinal $x(n)$ apenas com $H_{1/6}$, alguns pixels serão classificados erradamente como pertencentes a regiões escuras.



(a) Sinal original. (b) A linha vermelha, corresponde ao sinal gerado pelo filtro.

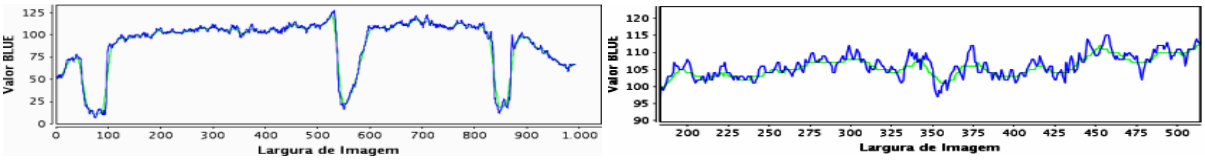
Figura 3: Resultado da aplicação do filtro $H_{1/6}$ a uma linha da imagem 1a.

A eliminação da influência do ruído no resultado da segmentação, obrigou ao desenho de um outro filtro. Novamente num filtro de média móvel, mas agora com uma janela de $Q = K/10 = W/60$ pontos. Este filtro designado por filtro $H_{1/60}$ é definido por:

$$y_b(n) = b_0 x(n+\frac{Q}{2}) + b_1 x(n+\frac{Q}{2}-1) + ... + b_{\frac{Q}{2}} x(n) + ... + b_{-\frac{Q}{2}+1} x(n-\frac{Q}{2}+1) + b_{-\frac{Q}{2}} x(n-\frac{Q}{2}) \quad (5)$$

$$b(n) = \begin{cases} \frac{1}{Q+1}, & se \quad -\frac{Q}{2} \leq n \leq +\frac{Q}{2} \\ 0, & c.c. \end{cases} \quad (6)$$

Tal como o filtro $H_{1/6}$, no filtro $H_{1/60}$ o número de pontos é ajustado em função do número de pixels do comprimento da imagem, sendo neste caso $1/60$ do comprimento da mesma. A figura 4 apresenta o resultado da aplicação do filtro $H_{1/60}$ e em 4b um *zoom* da figura 4a, de modo a se observar com mais pormenor o sinal $y_b(n)$, resultante da aplicação do filtro $H_{1/60}$ ao sinal $x(n)$.



(a) Aplicação do filtro $H_{1/60}$ (linha a verde). (b) Zoom de 4a.

Figura 4: Resultado da aplicação do filtro linear.

O sinal resultante da aplicação do filtro $H_{1/60}$, $y_b(n)$, é um sinal rápido que se ajusta bastante bem ao sinal $x(n)$, o que dificulta a sua visualização na figura 4a. A ligeira diferença existente entre o sinal original $x(n)$ e o sinal $y_b(n)$, embora quase imperceptível, permite a eliminação do ruído presente no sinal $x(n)$.

A conjugação dos dois filtros acima definidos, $H_{1/6}$ e $H_{1/60}$, permite não só a eliminação de ruído como a identificação de oscilações significativas da cor azul nas imagens de queijos, correspondendo estas oscilações às zonas onde se encontram os olhos do queijo. Este processo de filtragem é aplicado em todas as linhas da imagem, seguindo-se a subtracção do sinal resultante do filtro $H_{1/60}$ ao sinal resultante do filtro $H_{1/6}$.

$$y_{ab}(n) = y_a(n) - y_b(n) \quad (7)$$

Segue-se a comparação do sinal $y_{ab}(n)$ com um limiar (do inglês *threshold*) definido por 8, que varia em função da cor de fundo da linha a ser processada, de modo a adaptar o limiar a imagens muito escuras

e com pouco contraste entre o fundo do queijo e os seus olhos.

$$th_i = \begin{cases} 10 & se \quad max_i < 15 \\ (2 * max_i) - 20 & se \quad 15 \leq max_i < 20 \\ 20 & se \quad max_i \geq 20 \end{cases} \tag{8}$$

onde $max_i = \max_j(I_1(i,j) - I_0(i,j))$, I_0 a imagem original e I_1 uma imagem intermédia resultante dos filtros de média móvel $H_{1/6}$ e $H_{1/60}$.

Dessa comparação obtém-se uma nova imagem, $G(i,j)$, resultante da aplicação da segmentação definida na equação 9, segundo a qual se $y_{ab}(n)$ for superior ou igual ao limiar th_i, o pixel da nova imagem toma o valor *null* o que significa que pertence a zonas definidas como olhos do queijo, caso contrário o pixel receberá o valor da componente azul do pixel correspondente na imagem original. Na equação 9, $I(i,j)$ representa a imagem original.

$$Th(I(i,j), y_a(n), y_b(n)) = \begin{cases} null \quad se \quad y_a(n) - y_b(n) >= th_i \\ I(i,j) \qquad\qquad c.c. \end{cases} \tag{9}$$

Contudo, a tentativa de definição da cor de fundo do queijo pela aplicação do filtro $y_{ab}(n)$, é influenciada pela presença de zonas mais escuras, correspondentes aos olhos. Por isso, de modo a diminuir a influência dos olhos no nível de cor de fundo do queijo, o filtro $y_{ab}(n)$ irá ser aplicado à imagem $G(i,j)$, derivada da segmentação anterior. Este processo permite uma aproximação à verdadeira cor de fundo do queijo, de forma iterativa, melhorando o processo de segmentação.

O próximo filtro a aplicar terá de lidar com valores a *null*. Por isso o filtro $H_{1/6}$ é redefinido, para uma forma não linear, com os coeficientes da resposta impulsiva iguais a

$$a_{-\frac{K}{2}}, ..., a_{\frac{K}{2}} = \frac{1}{K-N} \tag{10}$$

onde N é o número de pixels a *null* da janela do filtro em cada momento. No caso de $K = N$ então $y_a = y_a(n-1)$, e no caso de $N = 0$ a definição do filtro mantém-se inalterada, tal como a equação 3.

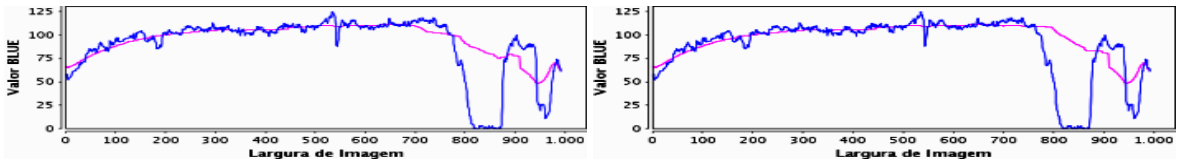
Após a aplicação de $H_{1/6}$, numa segunda iteração, é realizada nova segmentação mas agora comparando a subtracção do sinal resultante do filtro $H_{1/60}$ com o sinal filtrado pelo filtro $H_{1/6}$ (após a redefinição do filtro), com um novo limiar calculado novamente por 8, obtendo-se um novo limiar.

No entanto, esta segunda segmentação apresenta uma particularidade. Além dos sinais obtidos pelos filtros $H_{1/6}$ e $H_{1/60}$, é necessário ter conhecimento do resultado da segmentação anterior, de modo a que um pixel que tenha sido classificado como pertencente a um olho, não seja classificado como pertencente ao fundo, na segmentação seguinte. Embora tal facto seja assumido como natural, surgiram dois casos ao longo dos testes efectuados, que demonstram o contrário. Portanto, a equação 9 foi redefinida, assumindo a forma da equação 11.

$$Th(I(i,j), y_a(n), y_b(n), G(i,j)) = \begin{cases} null \quad se \quad y_a(n) - y_b(n) \geq th_i \vee G(i,j) = 0 \\ I(i,j) \qquad\qquad c.c. \end{cases} \tag{11}$$

A figura 5 apresenta o sinal resultante da aplicação da primeira e segunda iteração do filtro $H_{1/6}$, na forma não linear.

Uma vez diminuída a influência dos olhos, resultante da aplicação da filtragem não linear, consegue-se uma melhor aproximação à verdadeira cor de fundo do queijo, permitindo uma melhor definição dos contornos dos olhos.



(a) Sinal da primeira iteração do filtro não linear $H_{1/6}$. (b) Sinal da segunda iteração do filtro não linear, $H_{1/6}$.

Figura 5: Aplicação do filtro $H_{1/6}$ na forma não linear.

3 Considerações sobre o algoritmo

3.1 Número de iterações da filtragem não linear

Com o aumento do número de iterações do filtro não linear, assumem-se certos pressupostos, de entre os quais se destacam: melhor definição do contorno dos olhos; o acréscimo de área reconhecida como olhos diminui a cada iteração, tendendo para zero; o tempo de processamento aumenta.

Para conseguir um equilíbrio entre o tempo de processamento e o resultado final do algoritmo de segmentação, aplicou-se o algoritmo com diversas iterações às 34 fotografias de teste, tendo-se convencionado que a segmentação é aceitável, quando entre duas iterações consecutivas, o acréscimo de área reconhecida como olhos é inferior a 5%. No entanto, na eventualidade de ocorrer algum caso extremo, em que a condição anterior nunca se verifique, foi imposto um limite máximo de 5 iterações.

3.2 Valores nos limites da imagem

Os filtros descritos anteriormente, necessitam de valores exteriores à imagem nos limites laterais da mesma, sendo calculados segundo as equações 12 e 13, para o lado esquerdo e direito respectivamente.

$$x_{esquerda}(i,j) = \sum_{j}^{\frac{k}{2}-j} I(i,j) \times \frac{1}{\frac{k}{2}-j}, \quad j-\frac{k}{2} < i < 0 \tag{12}$$

$$x_{direita}(i,j) = \sum_{a=\frac{k}{2}-j}^{n} I(i,a) \times \frac{1}{\frac{k}{2}-a}, \quad n-(a-\frac{k}{2}) < i < n \tag{13}$$

onde n é a largura da imagem em pixels e k o número de impulsos da resposta impulsiva do filtro.

3.3 Resumo do processo de segmentação

O algoritmo de segmentação, poderá ser descrito pelas equações:

$$G^0(i,j) = Th\Big(I(i,j), \quad H_{1/6}(I(i,j)), \quad H_{1/60}(I(i,j)), \quad G^{-1}(i,j)\Big) \tag{14}$$

$$G^1(i,j) = Th\Big(I(i,j), \quad H_{1/6}(G^0(i,j)), \quad H_{1/60}(I(i,j)), \quad G^0(i,j)\Big) \tag{15}$$

$$G^2(i,j) = Th\Big(I(i,j), \quad H_{1/6}(G^1(i,j)), \quad H_{1/60}(I(i,j)), \quad G^1(i,j)\Big) \tag{16}$$

$$G^n(i,j) = Th\Big(I(i,j), \quad H_{1/6}(G^{n-1}(i,j)), \quad H_{1/60}(I(i,j)), G^{n-1}(i,j)\Big) \tag{17}$$

onde $G^{-1}(i,j) = I(i,j)$. De uma forma simplificada, o algoritmo pode ser igualmente descrito por:

$$G^n(i,j) = F(I(i,j), G^{n-1}(i,j)) \tag{18}$$

$$= Th\Big(I(i,j), H_{1/6}(G^{n-1}(i,j)), H_{1/60}(I(i,j)), G^{n-1}(i,j)\Big) \tag{19}$$

onde para $n = 0, G^{-1}(i,j) = I(i,j)$.

3.3.1 Representação em diagrama de blocos

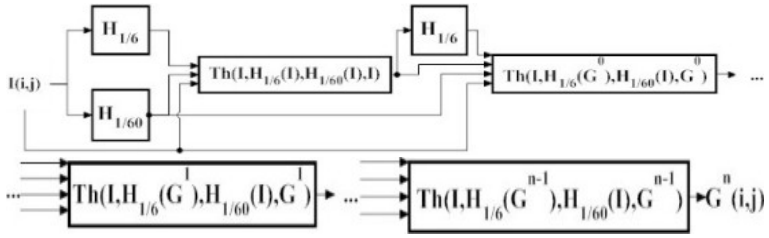


Figura 6: Diagrama de blocos que resume o algoritmo de segmentação.

3.4 Resultados

Nas imagens seguintes é apresentado o resultado do algoritmo de segmentação, para três imagens de queijos, onde a preto é apresentado as zonas reconhecidas como os olhos do queijo.

Figura 7: Resultados do algoritmo de segmentação.

O algoritmo desenvolvido, foi aplicado a um conjunto de 34 fotografias de queijos, sendo posteriormente analisados os resultados pela Eng. Maria da Graça Machado, responsável pelo Laboratório de Tecnologia e Qualidade dos Produtos Regionais, na Universidade de Évora.

Os restantes resultados obtidos para todas as fotografias utilizadas no teste ao algoritmo, podem ser consultados em `https://www.alunos.uevora.pt/(til)l20291/thesis/resultados.html`.

3.5 Análise de Desempenho

Foram efectuados testes com fotografias reais, sobre as quais foram registados os tempos de processamento. As 10 fotografias cujos tempos se encontram resumidos no gráfico de dispersão 8, têm a mesma dimensão de 1536×2048 pixels, verificando-se tal como seria espectável, que o tempo do processo de segmentação, varia em função da área da imagem reconhecida como queijo.

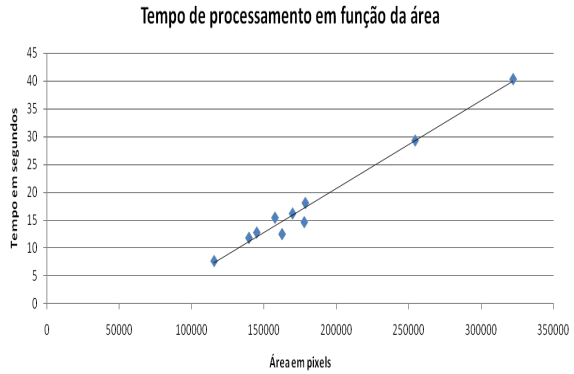
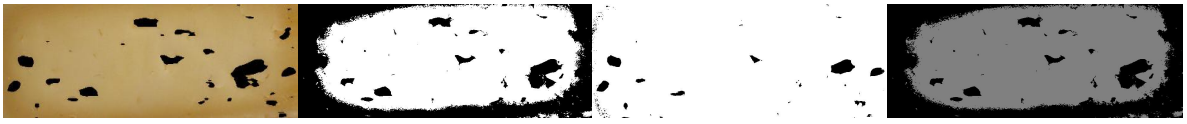


Figura 8: Tempo de processamento do algortimo de segmentação em função da área do queijo.

3.6 Comparação com outros algoritmos de segmentação

A avaliação do algoritmo desenvolvido, foi realizada por comparação com outros algoritmos de segmentação referidos na literatura e implementados na aplicação ImageJ. De entre os algoritmos testados destacam-se os algoritmos Limiar de Otsu [2], Limiar de Máxima Entropia e *K-Means Clustering* [3].



(a) Algoritmo descrito. (b) Otsu Threshold. (c) Max. Entropy Threshold. (d) K-Means Clustering.

Figura 9: Resultados do algoritmo de segmentação.

4 Conclusões

Analisando a comparação efectuada em 3.6, verifica-se que o algoritmo descrito apresenta uma melhor detecção dos olhos do queijo de Évora. No desenvolvimento do algoritmo foram ajustados diversos parâmetros, que tomando outros valores, podem possibilitar a aplicação do algoritmo a outro tipo de queijos assim como na análise de outros produtos naturais.

Além da detecção dos olhos, este algoritmo pode possibilitar a detecção de zonas escuras no queijo, ajustando os valores de cor, como por exemplo para os queijos azuis. A detecção de olhos maiores ou menores, pode ser igualmente ajustada em função do tamanho das janelas dos filtros.

Referências

[1] Isabel Lourtie. *Sinais e Sistemas*. Escolar Editora, 2nd edition, 2007.

[2] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9 (1):62–66, 1979.

[3] Guan Yong Shi Na, Liu Xumin. Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm. pages 63–67, 2010.

[4] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 2nd edition, 1999.

Lofoscopia Assistida por Computador: Técnicas Computorizadas para Comparação de Impressões Digitais

Nuno M. Chaves
Universidade de Évora
Évora, Portugal.
nmbchaves@gmail.com

Luís M. Rato*
Universidade de Évora
Évora, Portugal.
lmr@di.uevora.pt

Resumo

Este trabalho consiste num estudo sobre técnicas para comparar impressões digitais. O estudo foi feito através do desenvolvimento de uma aplicação, que implementa as técnicas e algoritmos necessários, e da parametrização destes algoritmos, de forma a obter os melhores resultados possíveis. Foi realizada uma pesquisa bibliográfica sobre o problema e as principais técnicas utilizadas, sendo proposta e testada uma implementação de um AFIS (Sistema Automatizado de Identificação de Impressões Digitais). Além dos métodos para fazer o processamento das imagens de impressões digitais, foi ainda desenvolvido um algoritmo, invariante à rotação e ao deslocamento, para comparação das mesmas. É ainda utilizado um módulo de classificação das impressões digitais, de modo a reduzir os tempos de pesquisa em bases de dados. O sistema proposto, foi desenvolvido, recorrendo a aplicações, cujo código fonte foi disponibilizado como *software* livre.

1 Introdução

A unicidade das impressões digitais permite a sua utilização para identificação de pessoas, uma vez que não se conhecem duas pessoas que apresentem a mesma impressão digital. No entanto, os processos manuais de identificação das mesmas são bastante limitados, o que torna a eficiência de todo o processo bastante reduzida. A maior limitação existente prende-se com o tamanho da impressão digital, uma vez que este é bastante reduzido, tornando difícil a identificação dos pontos característicos a olho nu. Por outro lado, as bases de dados de impressões digitais costumam ser de grandes dimensões, o que torna todo o processo manual bastante demorado.

Uma aplicação clássica de identificação por impressões digitais é a área criminal. No entanto, o mesmo estilo de sistema pode ser utilizado noutras áreas, sempre que seja necessário comprovar a identidade de uma pessoa. Áreas como sistemas de segurança e equipamentos de controle de acessos, autenticação em sistemas informáticos, ou mesmo marcadores de ”ponto”, são utilizações possíveis para sistemas de identificação.

Os métodos normais de identificação pessoal, usualmente utilizam outro tipo de informação, tal como uma senha ou mesmo uma chave ou um cartão. O problema é que este tipo de informação pode ser partilhado, esquecido, furtado ou perdido.

Por estes motivos, justifica-se o crescente desenvolvimento de sistemas automáticos, tendo como base os avanços tecnológicos nas áreas de processamento de imagens e reconhecimento de padrões. A utilização destes sistemas permite acelerar o reconhecimento de uma pessoa, com auxílio de sistemas computorizados, sem a necessidade de um especialista na área.

2 Lofoscopia e Dactiloscopia

A Lofoscopia[1] é a ciência que estuda os desenhos formados pelas cristas papilares das impressões digitais, palmas das mãos e plantas ou palmas dos pés. Esta ciência assenta nos pricípios de que as

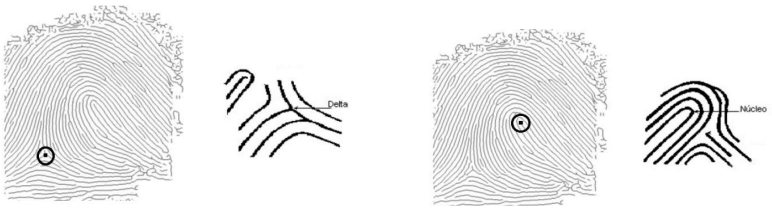
*Orientação, correcções e sugestões.

impressões digitais não sofrem alterações fisiológicas, patológicas ou voluntárias, desde a sua criação até ao apodrecimento dos tecidos, e que são todas diferentes de índividuo para índividuo.

A lofoscopia divide-se em três disciplinas diferentes, sendo a dactiloscopia aquela que é alvo de mais estudos. Por este motivo, é usual utilizar as duas palavras como sinónimos.

A dactiloscopia define-se como sendo o estudo detalhado das impressões digitais, com a finalidade de identificar indivíduos.

Uma impressão digital é uma reprodução das linhas das papilas digitais num qualquer suporte, sendo constituida por pontos singulares e pontos característicos.



(a) Delta (b) Núcleo

Figura 1: Pontos singulares das impressões digitais.

Os pontos singulares (núcleo e delta) são utilizados na classificação das impressões digitais, enquanto os pontos característicos (minúcias) são utilizados na comparação das mesmas.

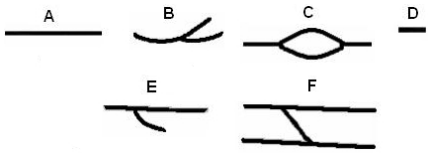


Figura 2: Pontos característicos das impressões digitais. [A - Crista final, B - Bifurcações, C - Ilha, D - Crista curta, E - Espora, F - Cruzamento]

3 Classificação de Impressões Digitais

Ao serem aplicadas técnicas de classificação sobre uma base de dados de impressões digitais, permite-se que os algoritmos de verificação sejam aplicados apenas a um sub-conjunto da mesma. Deste modo, é possível reduzir o tempo de pesquisa na base de dados.

O módulo de classificação utilizado, foi desenvolvido em [2], com base no trabalho de [3]. A descrição dos métodos utilizados (cálculo do mapa direccional, suavização e índice de Poincaré) pode ser consultado em [3].

4 Técnicas para o Processamento de Impressões Digitais

A construção de um sistema de identificação, através de impressões digitais, passa pela implementação de um conjunto de técnicas[4], que tentam tornar o sistema o menos falível possível. A estrutura do sistema pode ser dividida em pré-processamento, extracção de características e pós-processamento, sendo cada uma delas constituída por diferentes técnicas. O esclarecimento exaustivo dos métodos e algoritmos utilizados neste trabalho, pode ser consultado em [5].

4.1 Pré-processamento

Devido à má qualidade que as imagens de impressões digitais, frequentemente apresentam, é necessário aplicar um conjunto de técnicas, que tentem melhorar os aspectos importantes a extrair das mesmas. A fase de pré-processamento desenvolvida, inclui os seguintes métodos:

- Normalização de contraste 1
  As imagens de impressões digitais são, na maioria das vezes, de má qualidade, e têm deficiências nos níveis de contraste. Por este motivo, podem originar combinações de pixeis, que originem a extracção de falsas minúcias.

$$Pout = (Pin - imin)\left(\frac{pmax - pmin}{imax - imin}\right) + a \tag{1}$$

- Filtro de mediana
  O filtro de mediana tem como objectivo atenuar o ruído existente nos vales das impressões digitais, de modo a melhorar a diferenciação entre estes e as cristas das impressões digitais.

- Filtros de Gabor
  Os filtros de Gabor 2 são usualmente utilizados em imagens que apresentam ondas sinusóides, como é o caso das impressões digitais [6]. O filtro de Gabor consiste em substituir as linhas das impressões digitais por ondas sinusóides, criadas através de parâmetros, retirados da impressão digital. Mais informações sobre os filtros de Gabor podem ser consultadas em [7].

$$G(x,y:\theta,f) = e^{\left(-\frac{1}{2}\right)*\left(\frac{(x\cos(\theta)+y\sin(\theta))^2}{\sigma_x^2}+\frac{(-x\sin(\theta)+y\cos(\theta))^2}{\sigma_y^2}\right)} * \cos\left(2\pi f\left(x\cos\left(\theta\right)+y\sin\left(\theta\right)\right)\right) \tag{2}$$

- Binarização Adaptativa
  O processo de binarização, consiste em transformar uma imagem, com diferentes níveis de cinzento, numa outra, que apenas contêm pixeis de cor branca e preta. Uma vez que a imagem binarizada contêm menos informação, torna-se mais fácil e mais rápido o seu processamento [8].

(a) Impressão Digital Original. (b) Binarização após aplicação do filtro de Gabor. (c) Esqueletização.

Figura 3: Aplicação dos principais passos do pré-processamento numa impressão digital.

- Esqueletização
  A esqueletização tem como objectivo reduzir a quantidade de pontos de uma imagem, sem afectar a estrutura da imagem original. O esqueleto possui a largura de um pixel, e retêm as características da imagem inicial. A utilização do processo de esqueletização em impressões digitais, simplifica o processo de extracção de características [9].

4.2 Extracção de Características

Uma vez feita a optimização da imagem, através dos métodos anteriores, é agora possível, com maior exactidão, proceder à extracção dos aspectos característicos da mesma. Os métodos que constituem esta fase são os seguintes:

- Cálculo do ponto de referência
  O cálculo do ponto de referência é um passo crucial, uma vez que é através dele que se consegue uma invariância à rotação e ao deslocamento da impressão digital [10]. O ponto de referência de uma impressão digital está localizado nas linhas que apresentam a maior curvatura côncava. Mais detalhes sobre a implementação do cálculo do ponto de referência, podem ser consultados em [11].

Figura 4: Ponto de referência como origem de um sistema de eixos.

- Extracção de minúcias
  O método utilizado para a extracção de minúcias, foi o *Crossing Number* 3. Este método permite determinar se um pixel corresponde a uma minúcia, através do somatório de transições de preto para branco, que existem num conjunto de 9 pixeis (ou seja, uma janela de tamanho 3x3) [4].

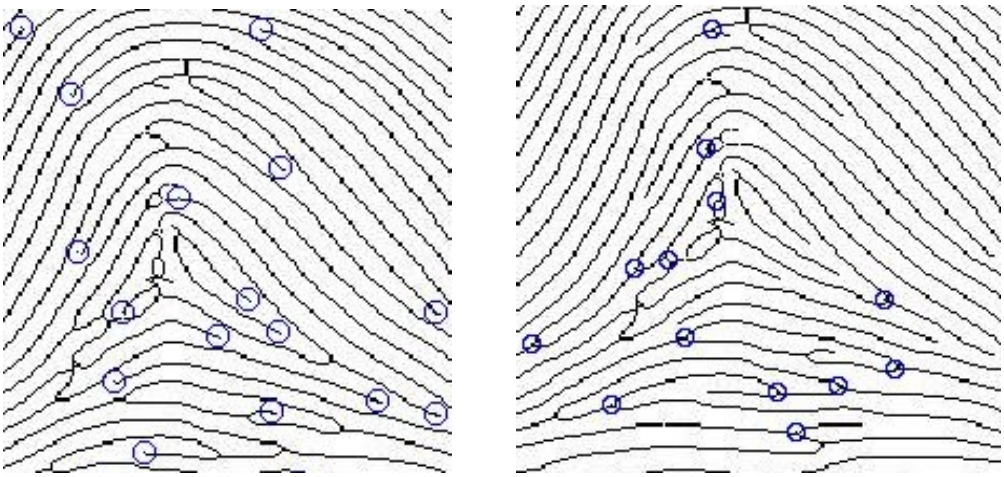
$$Cn(P) = \frac{1}{2} \sum_{i=1...8} |val(P_{mod(i,8)}) - val(P_{i-1})| \tag{3}$$

4.3 Pós-processamento

A última etapa da construção de um sistema AFIS, consiste no pós-processamento. Esta etapa é constituída pelos seguintes métodos:

- Validação de minúcias
  A etapa de validação de minúcias tem como objectivo, a localização e eliminação de pontos, que

foram, incorrectamente, marcados como minúcias. Este processo consiste em centrar cada minúcia em janelas de dimensões 11x11, verificando posteriormente, se as linhas que estão conectadas às minúcias terminam nas bordas das janelas. Caso isso não aconteça, significa que as linhas são demasiado pequenas, resultando de erros introduzidos pelos processos de pré-processamento. Para mais informações sobre este método, consultar [5].



(a) Terminações (b) Bifurcações

Figura 5: Pontos característicos de uma impressão digital, após detecção com o Crossing Number, e respectiva validação.

- Representação da impressão digital
  Antes da inserção da impressão digital na base de dados, é necessário criar uma representação 4 da mesma, que relacione a posição das minúcias com a localização do ponto de referência. A representação proposta consiste em calcular, para cada uma das minúcias da impressão digital, a sua distância euclidiana ao ponto de referência, e o ângulo que fazem com o mesmo. Uma vez calculadas as distâncias e os ângulos, cada minúcia é guardada numa lista, na forma de $[(D,\theta,T)_k]$, onde D representa a distância, θ o ângulo e T, o tipo de minúcia: terminação ou bifurcação.

$$[[D,\theta,T]_1,[D,\theta,T]_2,\ldots,[D,\theta,T]_k] \tag{4}$$

5 Comparação de Impressões Digitais

Para ser possível utilizar o algoritmo de comparação proposto, além da existência de uma base de dados com impressões digitais, a impressão digital a comparar tem de estar na representação proposta.

Sendo I a impressão digital a comparar, BD a base de dados das impressões digitais e C a classe da impressão digital a comparar, o algoritmo de comparação proposto apresenta o funcionamento descrito a seguir.

1. Recorrendo à classe C, seleccionam-se todas as impressões digitais em BD, que correspondem à mesma. Cria-se assim uma lista com todas as impressões digitais a ser comparadas com I.

2. Para cada imagem da lista anterior, é feita a comparação das distâncias das suas minúcias com a distância das minúcias de I. Esta comparação é feita minúcia a minúcia. Sempre que é feita uma

correspondência entre duas minúcias, estas são removidas. Procede-se deste modo para todas as minúcias. Se não existir um minímo de 12 minúcias em comum, a impressão digital é rejeitada, uma vez que não apresenta pontos suficientes em comum.

3. De forma a resolver o problema da rotação, são efectuados diversos incrementos de 1^o de cada vez (até um máximo de 359^o, que corresponde a uma rotação total da impressão digital. Esta rotação é feita em torno do ponto de referência), a todos os ângulos das minúcias de I. Em cada um dos incrementos, são comparadas as minúcias (da mesma forma do passo anterior), ângulos e tipo, guardando-se o valor máximo de minúcias comuns, entre as duas impressões digitais.

4. Terminada a rotação da impressão digital, é guardado, numa lista ordenada (M), o valor máximo de minúcias em comum entre as duas impressões digitais. A lista ordenada M, contêm as impressões digitais em que foram detectados mais pontos característicos em comum com a impressão digital a comparar (I).

6 Testes e Resultados Obtidos

De forma a verificar o desempenho do sistema proposto, foi feito um conjunto de testes, utilizando diferentes imagens de impressões digitais. Estas foram criadas com base nas imagens originais, de forma a testar a resposta da aplicação, em relação a impressões digitais com ruído, deslocadas, rotacionadas e parciais.

Algoritmo	**Tempo Médio (s)**
Classificação	2,28
Normalização do contraste	0,03
Filtro de Mediana	0,09
Filtro de Gabor	11,44
Cálculo do ponto de referência	66,14
Binarização	0,42
Esqueletização	1,67
Crossing Number	0,28
Validação das Minúcias	0,85
Identificação	5,07
Total	**88,27**

Tabela 1: Tempo de execução de cada algoritmo.

A tabela 1, apresenta os tempos médios de execução de cada algoritmo utilizado.

Para efeito de testes, considerou-se que uma impressão digital foi devidamente identificada com outra na base de dados, no caso de aparecer entre as 20 impressões digitais mais semelhantes. A tabela 2, mostra os resultados obtidos nos testes efectuados. A tabela 3, mostra os erros verificados na não identificação das impressões digitais.

	Originais	**Ruído**	**Deslocamento**	**Rotações**	**Parciais**
Não Identificadas	8	17	20	44	24
Identificadas	42	33	30	56	26

Tabela 2: Resultados dos testes aplicados às imagens de impressões digitais.

	Originais	**Ruído**	**Deslocamento**	**Rotações**	**Parciais**
Mal Classificadas	0	11	9	43	15
Ponto Referencia	0	6	5	1	1
Outro	8	0	6	0	8

Tabela 3: Erros verificados nas impressões digitais não identificadas.

Como se pode verificar, nos resultados dos testes anteriores (tabelas 2 e 3), o elevado número de impressões digitais não identificadas, deveu-se a erros de classificação. De modo a comprovar a influência da classificação errada, efectuou-se um teste recorrendo apenas a essas imagens. Neste teste, foram seleccionadas todas as imagens de impressões digitais usadas anteriormente, que resultaram numa não identificação.

	Mal Classificadas
Não Identificadas	12
Identificadas	66

Tabela 4: Resultados do teste aplicado às impressões mal classificadas.

Como se pode observar pelos resultados obtidos, apenas 12 das 78 impressão digitais mal classificadas, submetidas a teste, não foram identificadas. O único motivo da falha na identificação destas 12 impressões digitais, foi a incorrecta localização do ponto de referência. Com base nestes resultados, conclui-se que o processo de classificação utilizado, é o maior responsável pela não identificação das impressões digitais.

7 Conclusões e Trabalho Futuro

O sistema de identificação de impressões digitais proposto assenta sobre um conjunto de técnicas necessárias para o aperfeiçoamento da imagem, conjugadas com algoritmos de extracção de características. Os pontos característicos, juntamente com um ponto de referência, foram utilizados para criar uma representação da impressão digital, que, recorrendo ao algoritmo de comparação desenvolvido, se torna invariante à rotação e ao deslocamento.

Os resultados obtidos nos testes realizados, apesar de apresentarem um número elevado de impressões digitais que não foi possível identificar (apenas pela baixa eficácia demonstrada pelas técnicas de classificação), são bastante optimistas, uma vez que demonstram o correcto funcionamento dos métodos propostos, juntamente com o algoritmo de identificação desenvolvido. Os dois maiores problemas, relativos à comparação de impressões digitais (deslocamento e rotações das mesmas), podem ser resolvidos recorrendo aos métodos propostos.

Com este estudo, foi possível concluir que o processo de comparação de impressões digitais, é altamente complexo, e dependente da qualidade das imagens. Nas imagens de impressões digitais em que a qualidade da imagem é relativamente boa, a facilidade de localizar os pontos característicos é maior, originando uma extracção dos mesmos mais viável, e como consequência, uma identificação mais eficaz.

Como continuidade deste trabalho, tendo em vista o melhoramento do desempenho do sistema, sugerem-se as seguintes modificações:

- Modificação da implementação do algoritmo de cálculo do ponto de referência, de modo a tentar diminuir o tempo de processamento do mesmo.

- Implementação de todos os algoritmos na mesma linguagem de programação, com o objectivo de eliminar as chamadas ao sistema, diminuindo assim o tempo de processamento da aplicação.
- Alteração do mecanismo de classificação utilizado, com o objectivo de aumentar a taxa de sucesso da mesma.

Referências

[1] A. Simas and F. Calisto. *Metodologias de investigação criminal: Lofoscopia*. Centro de Recursos Didácticos e Audiovisuais, 1999.

[2] M. S. Lima. Gaudi-afis: Manual de referência, utilização e desenvolvimento., 2007.

[3] S. M. Costa. Classificação e verificação de impressões digitais. Master's thesis, Universidade de São Paulo, 2001.

[4] T. S. Castro. Identificação de impressões digitais baseada na extracção de minúcias. Master's thesis, Universidade Federal de Juiz de Fora, 2008.

[5] N. Chaves. Lofoscopia assistida por computador: Técnicas computorizadas para comparação de impressões digitais. Master's thesis, Universidade de Évora, 2010.

[6] F. M. Viola. Estudo sobre formas de melhoria na identificação de características relevantes em imagens de impressão digital. Master's thesis, Universidade Federal Fluminense, 2006.

[7] S. L. G. Oliveira, F. Viola, and A. Conci. Filtro adaptativo para melhoria de imagens de impressões digitais utilizando o filtro de gabor e campos direcionais. *4º Congresso Temático de Dinâmica, Controle e Aplicações*, 2005.

[8] L. P. P. Santos. Visão por computador, 1995. Texto de Apoio.

[9] D. R. Faria. Reconhecimento de impressões digitais com baixo custo computacional para um sistema de controle de acesso. Master's thesis, Universidade Federal do Paraná, 2005.

[10] C. M. S. Reis and C. A. R. Pacheco. Autenticação com impressão digital. Master's thesis, Instituto Superior de Engenharia de Lisboa, 2003.

[11] K. Nilsson and J. Bigun. Localization of corresponding points in fingerprints by complex filtering, 2003.

Processing and Classification of Biological Images Application to Histology

Bruno Nunes
Departamento de Informática
Universidade de Évora
Évora, Portugal

Luís Miguel Rato
Departamento de Informática
Universidade de Évora
Évora, Portugal

Fernando Capela e Silva
Departamento de Biologia
Universidade de Évora
Évora, Portugal

Ana Rafael
Instituto de Patologia Experimental
Universidade de Coimbra
Coimbra, Portugal

Antonio Silvério Cabrita
Instituto de Patologia Experimental
Universidade de Coimbra
Coimbra, Portugal

Abstract

This article deals with a histological problem by using image processing and feature extraction in images of renal tissues of rats and their classification through various methods such as: Bayesian inference, decision trees and support vector machines.

Keywords: *Image Processing, Segmentation, Feature Extraction, Classification, Histology, Kidney, Renal Glomeruli, Xenobiotics*

Artigo apresentado na conferência *TMSi 2010 - 6th International Conference on Technology and Medical Sciences*, encontrando-se disponível em: `http://hdl.handle.net/10174/2174`

Processamento de Imagens Digitais para o Controlo de Qualidade do Queijo Regional de Évora

Gaspar Brogueira
Universidade de Évora
gaspar_mrb@hotmail.com

Luís Rato
Universidade de Évora
lmr@di.uevora.pt

Maria Machado
ICAAM
mgjm@uevora.pt

Resumo

O queijo de Évora é um queijo curado, com Designação de Origem Protegida. O controlo de qualidade destes produtos exige usualmente a avaliação por painéis especializados. Esta avaliação tem em consideração três variáveis: o avaliador, o produto e a avaliação em si. A conjugação das características de cada variável, pode conduzir à não existência de consenso na avaliação de um queijo por parte dos elementos do painel.

Neste artigo são descritos diversos algoritmos, que resultaram num software, que permite a extracção de características objectivas do queijo, de modo a efectuar a avaliação dos mesmos, mediante o processamento de imagens digitais.

1 Introdução

O controlo de qualidade de produtos certificados exige usualmente a avaliação por painéis de provadores. Um dos objectivos deste trabalho é colocar à disposição de um painel de provadores um instrumento que extraia medidas rigorosas e objectivas de um produto certificado. Este trabalho concentra-se na avaliação do queijo de Évora. O queijo de Évora obedece a critérios bem definidos para a apreciação das suas características organolépticas, mediante diversos critérios [11]. Alguns desses critérios podem ser analisados visualmente. No que diz respeito à avaliação externa, o queijo deve apresentar cilindro baixo, com abaulamento lateral e um pouco na face superior, sem bordos definidos, crosta inteira, bem formada, lisa ou ligeiramente rugosa, amarelada, que escurece quando em contacto com o ar, consistência dura ou semidura. Na apreciação interna a pasta deve ser fechada e bem ligada, com aspecto untuoso e com alguns olhos pequenos, cor amarelada e uniforme.

Os critérios de apreciação externa e interna podem ser reconhecidos e analisados pelo processamento de fotografias digitais. Já os critérios de cheiro e sabor, são impossíveis de avaliar através de imagem.

Todos os produtores certificados têm de periodicamente submeter alguns queijos a uma Entidade Certificadora que, por sua vez, os envia para o painel de provadores do Laboratório de Tecnologia e Qualidade dos Produtos Naturais, no Pólo da Mitra da Universidade de Évora, que emite o seu parecer para a Entidade Certificadora.

Embora exista uma definição formal das características do queijo de Évora, usualmente não existe consenso relativamente à classificação e avaliação de um queijo por parte de todos os elementos do painel. A utilização de ferramentas com medidas rigorosas e objectivas, poderão potencialmente permitir uma avaliação mais objectiva e consistente por parte do painel de provadores.

Após cuidada pesquisa na mais diversa bibliografia especializada tanto no domínio de técnicas de processamento de imagem, como de análise de produtos naturais com as referidas técnicas, concluiu-se que se estava perante um problema ainda por explorar.

Posto isto, a abordagem seguida neste trabalho teve em consideração a complexidade crescente das soluções desenvolvidas, em função dos resultados e problemas encontrados.

2 Aquisição e Pré-processamento das imagens

2.1 Definição do Ambiente

Para que seja possível a avaliação de um queijo, mediante o processamento de uma imagem, é fundamental o seu reconhecimento na imagem. Como tal, o ambiente em que a fotografia é adquirida, deve proporcionar um reconhecimento simples do queijo, pelo que foi definido um conjunto de regras, que quando respeitadas, permitem um simples reconhecimento do queijo na imagem.

Outro aspecto que resulta da aquisição da fotografia num ambiente correcto, é a possibilidade de efectuar o cálculo aproximado da dimensão real do queijo. As condições a respeitar, são apresentadas de seguida:

1. Num fundo branco ou cinzento, definir um quadrado (moldura) de 15 cm de lado, com uma cor escura de forte contraste com o fundo;

2. O fundo deve ter cor suave e uniforme, não sendo necessariamente igual dentro e fora da moldura;

3. Cortar uma fatia de queijo com cerca de 2 a 4 mm de espessura;

4. A fatia cortada deve ser colocada com a maior dimensão paralela a um dos lados da moldura e sensivelmente no centro;

5. As dimensões da moldura devem ser superiores a 2/3 da fotografia.

2.2 Reconhecimento do Queijo

Para a avaliação de um queijo, considerou-se necessário a extracção de diversas características relativas ao queijo, nomeadamente sobre a cor e os olhos. Mas tal só é possível após o reconhecimento da região correspondente ao queijo na fotografia, pelo que, se o ambiente tiver sido definido correctamente, o queijo deverá estar algures no interior da moldura. Portanto, o primeiro processamento a aplicar sobre a imagem deverá permitir a detecção da moldura.

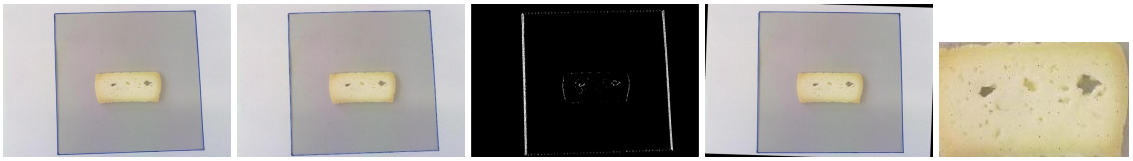
O processo de detecção de objectos em imagens não é simples de realizar de forma automática [13], pelo que deve iniciar-se o processamento da imagem utilizando algoritmos de detecção de contornos, como por exemplo, os operadores de operadores Sobel [9], ou operadores de Prewitt [17].

Os operadores de Prewitt efectuam o cálculo do gradiente da luminosidade de cada pixel da imagem, informando sobre como varia a luminosidade ao longo da imagem. As variações bruscas de cor, normalmente correspondem aos limites de objectos, pelo que os contornos dos objectos são identificados por uma nítida alteração de cor ao longo de determinada orientação. A abordagem desenvolvida efectua o processamento das imagens apenas a uma dimensão, direcção horizontal, pelo que se pretende apenas detectar as linhas verticais da moldura.

Com a aplicação dos operadores de Prewitt para a detecção das linhas verticais da moldura, é natural o aparecimento de algum ruído na forma de pontos de tonalidade branca dispersos indefinidamente. De forma a eliminar este tipo de ruído, foi utilizado um filtro de mediana [19].

Identificadas as linhas verticais da moldura, prossegue-se verificando se a moldura apresenta algum grau de inclinação. Se existir inclinação da moldura superior a um grau $(0.017453rad)$, procede-se à correcção dessa inclinação, de modo a não influenciar no cálculo da largura da moldura, e por conseguinte, não afectar o cálculo da dimensão real do queijo.

A figura 1 resume o procedimento descrito anteriormente.



(a) Imagem inicial (b) Filtro de Mediana (c) Operadores Prewitt (d) Rotação (e) Queijo

Figura 1: Sequência de algoritmos de pré-processamento, que permitem o reconhecimento da região correspondente ao queijo na imagem inicial.

3 Reconhecimento dos olhos do Queijo

A detecção dos olhos, consiste num problema de segmentação, conceito que nos remete para a separação de uma imagem em regiões com atributos em comum. Neste caso, o atributo é a cor, ou seja, as zonas correspondentes aos olhos do queijo apresentam uma cor mais escura, quando comparado com a cor de fundo do queijo.

Para este problema foram testados algoritmos de segmentação referenciados na literatura e implementados na aplicação ImageJ. De entre os algoritmos testados destacam-se os algoritmos *Otsu Threshold* [10], *Maximum Entropy Threshold* e *K-Means Clustering* [16]. Os resultados obtidos não satisfizeram o objectivo inicial, pelo que se optou pelo desenvolvimento de um algoritmo de segmentação para problema em causa.

Pela definição de sinal, apresentada em [7] [14], tem-se que a variação do brilho da cor ao longo das varáveis espaciais x e y, definem uma imagem digital. Para a detecção das regiões escuras da imagem, foi apenas considerado o processamento segundo a variável espacial x e a componente azul do espaço de cores RGB.

O algoritmo desenvolvido, onde se pretende determinar um sinal que se aproxime da cor de fundo do queijo, é descrito pelos seguintes passos:

1. Aplicação de um filtro de média móvel ao sinal da imagem original, I, que permita a redução de variações abruptas de cor, sobre regiões suficientemente grandes, com uma resposta impulsiva de $W/6$ impulsos com soma unitária, em que W é a largura da imagem, resultando no sinal $y_a(n)$;

2. Aplicação novamente de um filtro de média móvel ao sinal da imagem original, I, que permita a eliminação de bruscas oscilações de cor, tendo neste caso a resposta impulsiva com $W/60$ impulsos de soma unitária, permitindo a eliminação do ruído presente no sinal, resultando no sinal $y_b(n)$;

3. O sinal $y_b(n)$ é subtraído ao sinal $y_a(n)$, originando o sinal $y_{ab}(n) = y_a(n) - y_b(n)$. O sinal $y_{ab}(n)$ é posteriormente comparado com um limiar (*threshold*) que varia em função da cor de fundo da linha a ser processada, de modo a adaptar o limiar a imagens muito escuras e com pouco contraste entre o fundo do queijo e os seus olhos.

4. Da comparação efectuada no ponto anterior, obtém-se uma nova imagem, G, em que se $y_{ab}(n)$ for superior ou igual ao limiar, o pixel da nova imagem toma o valor *null* o que significa que pertence a zonas definidas como olhos do queijo, caso contrário o pixel receberá o valor da componente azul do pixel correspondente na imagem original. Neste ponto, ficam identificadas as regiões onde se encontram os olhos.

5. O processamento prossegue, com a aplicação de um novo filtro por redefinição do filtro aplicado no ponto 1, para uma forma não linear, visto que, o número de impulsos da resposta impulsiva,

é dado em função do número de pixels colocados a *null*, no ponto anterior. Para esta versão não linear, o número de impulsos será $W/6 - N$, onde N é o número de pixels a *null* resultantes do ponto 4.

6. O filtro do ponto 5, é aplicado iterativamente até que entre duas iterações consecutivas o acréscimo de área reconhecida como olhos do queijo, seja inferior a 5%. Neste ponto, os contornos dos olhos são melhor definidos, uma vez que a filtragem é efectuada sobre sinais cuja influência das regiões escuras dos olhos foi eliminada, possibilitando uma melhor aproximação à cor de fundo do queijo. A comparação com um limiar é de novo efectuada, a cada iteração do filtro.

7. O processamento termina, com a criação de uma imagem binária, em que o fundo do queijo é representado com a cor branca e a preto os olhos.

Os detalhes matemáticos deste algoritmo de segmentação, podem ser encontrados em [5].

3.1 Resultados

Nas imagens seguintes é apresentado o resultado do algoritmo de segmentação, para três imagens de queijos, onde a preto é apresentado as zonas reconhecidas como os olhos do queijo.

Figura 2: Resultados do algoritmo de segmentação.

4 Aprendizagem e Classificação

Relativamente às imagens binárias obtidas no ponto 7 da secção anterior, é possível extrair diversas características relativas aos olhos. Para tal, foi utilizado o algoritmo *Analyse Particles*, da aplicação ImageJ [1].

Este algoritmo realiza a análise de partículas em imagens binárias, neste caso os olhos, permitindo o cálculo do número de olhos, o seu tamanho médio, a fracção de área por eles ocupada, a área total em pixels e ainda o perímetro de cada olho. Esta última característica será útil para o cálculo da circularidade média, de forma ponderada, da generalidade dos olhos.

Além das características sobre os olhos, é efectuada a recolha de parâmetros relativos à cor do queijo, nomeadamente a média das componentes R, G e B do espaço de cores RGB, assim como a média das componentes H, S e V do espaço de cores HSV. É igualmente calculada a média da cor na escala de cinzentos. A imagem reconhecida como queijo, é dividia em quatro regiões: o fundo na sua totalidade, a crosta, cantos e centro. Para cada uma destas regiões são calculadas as medidas anteriormente enunciadas.

Para a definição do modelo de aprendizagem e classificação, foi utilizada a ferramenta de mineração de dados Weka [2], que disponibiliza a implementação de diversos algoritmos em JAVA, tais como, árvores de decisão, *support vector machines*, regras de decisão, entre outros.

Numa primeira abordagem, foi testada a hipótese de classificar um queijo recorrendo a apenas um algortimo, utilizando todos os atributos do queijo. No entanto, não se obteve resultados relevantes [4].

Partindo do princípio que foram recolhidas características relativas aos olhos e à cor do queijo, seguiu-se uma abordagem que permite a combinação de diversos algoritmos de classificação, de modo

a que o processo de aprendizagem assentasse em diversos parâmetros do queijo a serem considerados individualmente. Cada queijo é, então, avaliado relativamente a características dos olhos, à cor na sua generalidade, à homogeneidade da cor comparando o centro e a crosta e, finalmente, por uma combinação de parâmetros dos olhos e da cor. A avaliação final será por isso, apoiada nestas quatro avaliações intermédias.

A abordagem seguida neste trabalho e que se fundamenta na manipulação dos exemplos de treino com o objectivo de gerar múltiplos modelos, foi desenvolvida nos finais do século XX e tem por base a capacidade de combinar diversos algoritmos de aprendizagem, obtendo-se por vezes resultados surpreendentes [20], [8]. De entre os esquemas desenvolvidos, destacam-se os esquemas designados por *voting*, *bagging*, *boosting* e *stacking* [18].

O esquema de *stacking*, permite combinar diferentes tipos de algoritmos de aprendizagem [20], pelo que, este esquema perfila-se como um esquema adequado ao problema de classificação em análise, visto não impor restrições ao tipo de algoritmo a utilizar, dando total liberdade para a escolha do algoritmo que melhor se adequa ao parâmetro que se pretende analisar. A selecção do algoritmo para a avaliação de cada parâmetro, foi efectuada mediante o teste de todos os algoritmos disponíveis no Weka, escolhendo o algoritmo que apresentasse os melhores resultados para cada caso.

O modelo de *stacking* tenta aprender usando um algoritmo, o *metalearner*, que descobre a melhor forma de combinar o resultado dos algoritmos base [20]. Este processo é realizado em dois níveis, sendo que no *nível-0* são geradas as predições utilizadas como input para o meta-modelo do *nível-1*.

No *nível-0*, cada queijo é avaliado segundo quatro parâmetros, com quatro algoritmos de aprendizagem diferentes:

Classificação em função da cor - Algoritmo SMO [12] - Cada queijo é classificado numa das seguintes classes: ”branco”, ”escuro”, ”centro escuro”e ”centro claro”.

Classificação em função dos olhos - Algoritmo J48 [3] - Um queijo é classificado como tendo ”muitos”, ”poucos”ou ”médios”olhos.

Classificação em função da homogeneidade da cor - Algoritmo Best-First Decision Tree [15] - É atribuida a um queijo a classificação de ”sim”ou ”não”, se apresenta ou não cor homogenea, respectivamente.

Classificação em função da cor e dos olhos - Algoritmo Alternating Decision Tree [6] - Um queijo é classificado como ”muito bom”ou ”mau”.

O modelo do *nível-1* é definido em função dos resultados das classificações do *nível-0* e recebe como input as classes definidas em cada um dos parâmetros anteriores. Portanto, a classificação final de um queijo, resultado da classificação obtida no *nível-1*, corresponderá a uma das seguintes classes:

Muito Bom - Queijo conforme a definição, ou seja, com poucos ou nenhuns olhos e cor ligeiramente amarelada;

Bom - Queijo onde se verifica a presença de alguns olhos e cuja cor poderá não ser tão uniforme quanto os queijos da classe Muito Bom;

Suficiente - Queijo com olhos de dimensões médias e cor pouco uniforme;

Insuficiente - Queijo com olhos de dimensões consideráveis e cor muito pouco uniforme;

Mau - Queijos com cor muito escura ou muito clara denotando tempo de cura incorrecto. No caso de queijos muito brancos, os olhos têm dimensões elevadas, denotando muito pouco tempo de cura.

Tabela 1: Avaliação do modelo de classificação.

Classe	Sensibilidade	Especificidade	PICC
Muito Bom	75%	93.3%	91.18%
Bom	80%	93.1%	91.18%
Suficiente	83.3%	96.4%	94.12%
Insuficiente	83.3%	100%	97.06%
Mau	92.3%	100%	97.06%

4.1 Resultados

Quanto à avaliação do modelo de aprendizagem desenvolvido, este foi avaliado em termos de sensibilidade, especificidade e percentagem de instâncias correctamente classificadas (PICC), sendo os resultados apresentados na tabela seguinte.

Segundo [8] um classificador é tanto mais próximo do ideal quanto maiores forem as percentagens para as 3 métricas. No entanto, deve ter-se em conta que existe um compromisso entre sensibilidade e especificidade, pois um aumento da sensibilidade pode diminuir a especificidade.

Verifica-se que os resultados obtidos com o classificador descrito acima, apresentam para as três métricas percentagens elevadas, conseguindo-se inclusivamente a especificidade máxima para as classes ”Insuficiente”e ”Mau”.

5 Software

A integração dos algoritmos descritos neste artigo resultou numa aplicação de software, que foi desenvolvida com a preocupação de tornar a experiência para o utilizador o mais simples possível.

O código da aplicação foi implementado integralmente na linguagem de programação JAVA (versão 1.6.0_03), aproveitando todas as suas potencialidades de desenvolvimento e independência da plataforma onde a aplicação irá ser executada. O ambiente de desenvolvimento consistiu essecialmente no sistema operativo Ubuntu 7.04 e no editor de texto Kate 2.5.8. Numa fase final, com o evoluir da aplicação, houve a necessidade de utilizar um IDE mais completo, nomeadamente o NetBeans 6.9.1.

Na aplicação desenvolvida, foram integrados componentes de outros programas igualmente desenvolvidos em JAVA, tais como o ImageJ 1.42q, JFreeChart 1.0.13 e Weka 3.6.3.

Uma descrição com mais detalhe do funcionamento do software pode ser encontrada em [4].

6 Conclusões

A aplicação de software resultante da conjugação dos algoritmos descritos neste artigo, que permite a avaliação automática de alguns parâmetros dos queijos de Évora (Denominação de Origem Protegida). A aplicação desenvolvida é uma ferramenta auxiliar que poderá ser usada tanto por membros do painel de avaliação como no controlo de qualidade pelos próprios produtores, controlando assim o tempo de cura do respectivo queijo.

A aplicação analisa várias características das imagens de queijos. O número de olhos, assim como a fracção de área que ocupam, são características importantes para uma correcta avaliação da sua qualidade. Encontram-se na bibliografia diversos algoritmos e métodos de reconhecimento de objectos em imagens, no entanto, por nenhum dos algoritmos testados se adaptar ao problema de reconhecimento dos olhos do queijo, foi desenvolvido um método de segmentação adaptado ao problema.

O software desenvolvido foi testado com fotografias e dados reais, obtidos com a colaboração do painel de provadores do Laboratório de Tecnologia e Qualidade dos Produtos Naturais, da Universidade

de Évora, obtidas e pré-processadas segundo a metodologia descrita neste artigo.

Referências

[1] http://rsbweb.nih.gov/ij/. (Acedido em 13/11/2010).

[2] http://www.cs.waikato.ac.nz/ ml/weka/. (Acedido em 13/11/2010).

[3] V.P. Bresfelean. Analysis and predictions on students' behavior using decision trees in weka environment. pages 51–56, 2007.

[4] Gaspar Brogueira. Processamento de imagem digital para o controlo de qualidade do queijo regional de Évora. Master's thesis, Universidade de Évora, http://www.alunos.uevora.pt/(til)l20291/thesis, 2010. (Acedido em 13/11/2010).

[5] Gaspar Brogueira and Luís Rato. Algoritmo de segmentação para a detecção dos olhos do queijo regional de Évora. *JIUE - Jornadas de Informática da Universidade de Évora*, 2010.

[6] Yoav Freund and Llew Mason. The alternating decision tree learning algorithm.

[7] Isabel Lourtie. *Sinais e Sistemas*. Escolar Editora, 2nd edition, 2007.

[8] José Maia Neves Miguel Rocha, Paulo Cortez. *Análise Inteligente de Dados, Algoritmos e Implementação em JAVA*. FCA, 2008.

[9] R.L. Baker N. Kanopoulos, N. Vasanthavada. Design of an image edge detection filter using the sobel operator. *IEEE Journal of Solid-State Circuits*, 23:358 – 367, 1988.

[10] Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62 –66, 1979.

[11] Cristina Pinheiro. *Contributo para a caracterização do queijo de ovelha produzido na região de Évora: aspectos químicos, bioquímicos do leite obtido em diferentes sistemas de produção e físico-químicos, bioquímicos, tecnológicos e organolépticos do queijo* . PhD thesis, Universidade de Évora, 2001.

[12] John C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. *Technical Report MSR-TR-98-14*, 1998.

[13] Yong-Mei Cheng Quan-Xue Gao Qi-Chuan Tjan, Quan Pan. Fast algorithm and application of hough transform in iris segmentation. *Third International Conference on Mache Learning and Cybemetics*, 2004.

[14] B. A. Sehnoi. *Introduction to Digital Signal Processing and Filter Design*. Wiley, 2006.

[15] Haijian Shi. Best-first decision tree learning. Master's thesis, The University of Waikato, 2006.

[16] Guan Yong Shi Na, Liu Xumin. Research on k-means clustering algorithm: An improved k-means clustering algorithm. pages 63–67, 2010.

[17] Zhou Shisheng Wang Dong. Color image recognition method based on the prewitt operator. In *International Conference on Computer Science and Software Engineering*, volume 6, pages 170 – 173, 2008.

[18] GE Zhihui WEI Yanyan, LI Taoshen. Combining distributed classifies by stacking. *Third International Conference on Genetic and Evolutionary Computing*, 2009.

[19] Mark Burge Wilhelm Burger. *Digital Image Processing an Algorithmic Introduction using Java*. Springer, 2007.

[20] Ian H. Witten and Eibe Frank. *Data Mining, Practical Machine Learning Tools and Techniques*. Elsevier, 2005.

Classificação de Textos com Métodos de Núcleo para Grafos

Miguel Gaspar, Teresa Gonçalves, Paulo Quaresma
Universidade de Évora
miguel.ferreira.gaspar@gmail.com, tcg@uevora.pt, pq@uevora.pt

Resumo

A aproximação mais comum para a Classificação de Textos consiste em utilizar uma representação saco-de-palavras, onde cada documento é representado pelas palavras que contém, sendo a ordem e pontuação ignoradas. Este trabalho apresenta outra aproximação a este problema, onde é utilizada uma representação semântica dos documentos. Esta representação é obtida utilizando a Teoria de Representação do Discurso e cada documento é representado posteriormente através de um grafo, o que permite a utilização de Métodos de Núcleo para dados estruturados. A correcta utilização do núcleo para grafos exige a definição de uma função de semelhança entre grafos que foi avaliada utilizando documentos do conjunto de dados `Reuters-21578`.

1 Introdução

O problema de classificação automática de documentos de texto é de grande importância prática, dado o enorme volume de textos disponíveis na *World Wide Web*, catálogos de notícias, correio electrónico, bases de dados empresariais, fichas médicas de pacientes, bibliotecas digitais, etc.. Os algoritmos de aprendizagem podem ser treinados para classificar documentos, dado um conjunto suficiente de exemplos de treino previamente classificados. Estes algoritmos têm sido utilizados para classificar automaticamente novos artigos [15], páginas da internet [4], aprender automaticamente o interesse da leitura de utilizadores [19], e classificar automaticamente correio electrónico [20] [23].

A aproximação mais comum para a Classificação de Textos consiste em utilizar uma representação saco-de-palavras, onde cada documento é representado pelas palavras que contém, sendo a ordem e pontuação ignoradas.

Este trabalho apresenta uma outra aproximação a este problema, onde é utilizada uma representação semântica dos documentos. Esta representação é obtida utilizando a Teoria de Representação do Discurso e cada documento é representado posteriormente através de um grafo [11]. A definição de uma medida de semelhança entre esses grafos (que representam a informação semântica dos documentos) permite a utilização de Métodos de Núcleo para dados estruturados em problemas de Classificação de Textos.

O artigo está organizado da seguinte forma: a Secção 2 introduz os conceitos utilizados e a Secção 3 propõe uma representação dos documentos que inclui a sua representação semântica e é adequada à utilização de métodos de núcleo e uma medida de semelhança entre documentos representados desta forma; a Secção 4 descreve as experiências realizadas e faz uma avaliação dos resultados obtidos. A Secção 5 apresenta as conclusões e trabalho futuro.

2 Conceitos

Este trabalho incorpora conceitos sobre informação linguística para representar os documentos e conceitos sobre métodos de núcleo para dados estruturados para permitir a classificação desses documentos. Esta secção introduz esses conceitos.

2.1 Teoria da Representação do Discurso

A Teoria da Representação do Discurso (TRD)[1] desenvolvida por Kamp [16] é uma teoria de semântica dinâmica cujo principal interesse consiste em considerar o significado do texto dependente do contexto. Este teoria utiliza como linguagem de representação semântica as Estruturas de Representação do Discurso (ERD)[2] [17]. Estas são formadas por dois elementos básicos: um conjunto de Referentes do Discurso, geralmente referido como o seu "universo" e um conjunto de condições. Intuitivamente, os **referentes** identificam as entidades referidas no discurso, enquanto as **condições** representam as restrições (propriedades, relações) associadas a essas entidades.

A Figura 1 apresenta uma DRS para a frase *"He throws the ball."*. Nela é possível observar os referentes `X1`, `X2` e `X3` e as restrições `male(X1)`, `ball(X2)`, `throw(X3)`, `event(X3)`, `agent(X3,X1)` e `patient(X3,X2)`. Estas restrições indicam que `X1` é masculino, `X2` é uma bola e `X3` é um evento cujo agente é `X1` e o paciente é `X2`;

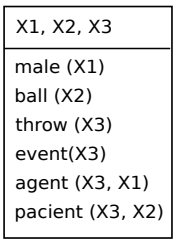


Figura 1: Exemplo da DRS para a frase *"He throws the ball."*

Uma teoria muito semelhante foi desenvolvida independentemente por [12], sob o nome de Lista de Mudança Semântica[3].

2.2 Máquinas de Vectores de Suporte

As Máquinas de Vectores de Suporte (MVS)[4] são um algoritmo de aprendizagem automática supervisionada proposto por Vapnik [25, 24]. Constitui uma implementação do método de minimização de risco estrutural[5] que utiliza uma técnica de núcleo.

Assumindo um conjunto de dados não separáveis linearmente, a MVS transforma do conjunto de dados iniciais num novo conjunto de padrões linearmente separáveis [22] obtendo um modelo onde a margem de separação entre classes seja máxima. A transformação do espaço de características é realizada implicitamente através de uma função de núcleo. Esta função depende do tipo de dados e de conhecimento específico do domínio dos dados.

2.2.1 Núcleo para Grafos

A aplicação de funções de núcleo a dados estruturados, representados por grafos foram introduzidos de forma independente [9] e [18]. Conceptualmente, estes núcleos baseiam-se numa medida sobre os percursos nos dois grafos que têm etiquetas em comum. No primeiro, contam-se os percursos com etiquetas iniciais e finais iguais e no outro são calculadas as probabilidades de percursos aleatórios com sequências de etiquetas iguais.

[1] Do inglês, *Discourse Representation Theory* (DRT)
[2] Do inglês, *Discourse Representation Structures* (DRS)
[3] Do inglês, *File Change Semantics* (FCS)
[4] Do inglês, *Support Vector Machines* (SVM)
[5] Do inglês, *structural risk minimization framework*

O **núcleo do grafo de produto** introduzido por Gartner [10] utiliza o grafo de produto directo[6], e baseia-se na ideia de contar o número de percursos nesse grafo de produto. O grafo de produto directo, obtido a partir de dois grafos, permite gerar de forma elegante todos os caminhos comuns entre esses grafos. Os grafos de produto são uma ferramenta da matemática discreta[14], sendo grafo de produto directo um dos mais importantes.

Formalmente, o núcleo do grafo de produto entre dois grafos G_1 e G_2 é definido por:

$$k(G_1, G_2) = \sum_{i,j=0}^{|V_x|} \left[\sum_{n=0}^{\infty} \lambda_n E_x^n \right]_{ij} \tag{1}$$

onde E_x é a matriz de adjacência do produto, $E_x = E(G_1 \times G_2)$; V_x é o conjunto de vértices do produto directo, $V_x = V(G_1 \times G_2)$; e $\lambda_i \in \mathbb{R}; \lambda_i \geq 0$ para todo o $i \in \mathbb{N}$.

A matriz de adjacência E_x^n é uma matriz quadrada onde o elemento (i, j) contabiliza os percursos de tamanho n entre o nó i e o nó j do grafo de produto directo.

3 Semelhança entre documentos

A utilização da informação semântica dos documentos constitui uma abordagem diferente para a classificação de textos. A representação dessa informação através de Estruturas de Representação do Discurso e a sua transformação em grafos possibilita a aplicação das máquinas de vectores de suporte com o núcleo do grafo de produto.

Esta secção introduz o processo de obtenção dos grafos representativos da informação semântica e a definição da medida de semelhança entre grafos que possibilita a utilização do algoritmo de núcleo de grafo de produto.

3.1 Representação semântica do documento

A Figura 2 apresenta a interligação entre as diferentes ferramentas que permitem transformar um texto na sua representação semântica através de uma ERD.

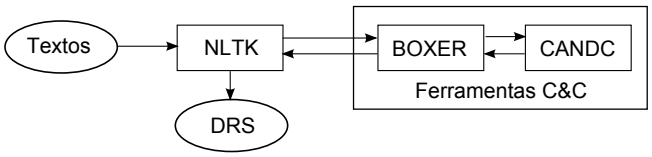


Figura 2: Ferramentas que transformam um texto numa ERD

Esta abordagem considera a utilização do NLTK, C&C e BOXER através de uma interface: o NLTK solicita ao BOXER que lhe seja devolvida a ERD de uma frase; este, por sua vez, solicita a C&C as derivações CCG que lhe permitem gerar a representação semântica que será devolvida ao NLTK, permitindo posteriormente a resolução de anáforas.

O NLTK (Natural Language Toolkit) [2, 8, 1] é uma ferramenta de processamento da linguagem natural que tem sido largamente utilizada no mundo académico, servindo de base de muitos projectos de pesquisa e foi originalmente criada em 2001 como parte de um curso de linguística computacional do Departamento de Computação e Ciência da Informação da Universidade da Pensilvania.

O NLTK implementa, entre outros, protótipos para etiquetagem morfo-sintáctica, classificação de textos, extracção de informação, análise sintáctica e semântica e resolução de anáforas. A inexistência

[6]Do inglês, *direct product graph*

de uma gramática robusta para a língua inglesa impossibilitou a sua utilização para a obtenção das ERD. Assim, foi necessário recorrer à ferramenta C&C, para obter tal representação.

A ferramenta C&C [7] é composta por duas aplicações:

- CANDC, que combina um analisador morfológico [21], marcadores morfo-sintácticos [6], marcadores de entropia máxima e de Gramática de Categorias Combinatória[7] e identificadores de entidades [5].
- BOXER, que permite obter a representação semântica dos textos. Esta aplicação necessita das derivações CCG obtidas pela da aplicação CANDC.

O documento como um grafo. A transformação da ERD num grafo, necessária para possibilitar a aplicação do núcleo do grafo do produto, foi realizada representando os referentes como nós do grafo e as restrições como arcos direccionados entre os referentes. A Figura 3 apresenta o grafo correspondente à ERD representada na Figura 1.

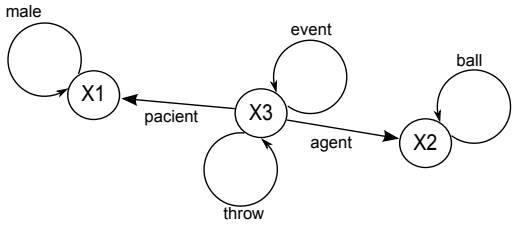


Figura 3: Grafo representativo da frase *"He throws the ball."*

Para representar os grafos foi utilizada a linguagem GML (*Graph Modelling Language*) [13], uma linguagem de sintaxe simples e que permite extensibilidade, flexibilidade e portabilidade.

3.2 Núcleo do grafo de produto

O grafo de produto directo obtido a partir de dois grafos G_1 e G_2 depende da definição de uma função de igualdade entre os nós e entre os arcos dos grafos factores. Esta função é responsável pela decisão de quais os nós e arcos que farão parte do grafo do produto directo medindo, indirectamente, a semelhança entre os dois documentos.

3.2.1 Mapeamento de referentes

Como a análise de cada texto é independente dos restantes, a igualdade dos nós não pode utilizar as etiquetas que lhes estão associadas (referentes das ERD) já que referentes diferentes em textos diferentes podem ter a mesma etiqueta, e a mesma entidade em textos diferentes pode ser identificada através de etiquetas diferentes.

Assim, e para mapear nós que possam referir a mesma entidade utilizaram-se os arcos que ligam esses nós, tendo sido construído o grafo de produto directo (que contém os nós comuns aos grafos factores) sobre o 'melhor' mapeamento entre os nós dos dois textos. Este mapeamento é feito através de uma pesquisa gulosa começando pelos nós que apresentam maior numero de arcos com etiquetas idênticas. Em caso de empate dá-se primazia aos nós com o maior percurso em arcos de etiquetas idênticas. Vejamos, por exemplo, a aplicação deste método na obtenção do grafo de produto entre os grafos **A** e **B**, representados na Figura 4: o arco '`a`' entre os nós `A` e `B` do grafo **A** e o arco '`a`' entre os nós

[7] Do inglês, *Combinatory Categorial Grammar (CCG)*

`Z` e `W` do grafo **B** têm etiquetas idênticas, logo, consideramos que `A` corresponde a `Z` e que `B` corresponde a `W`.

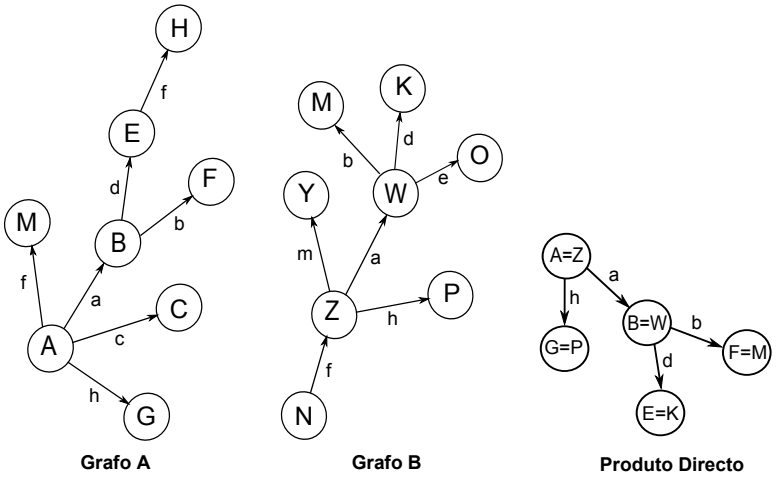


Figura 4: Os grafos **A** e **B** e o grafo do produto directo

3.2.2 Classificador de documentos

Obtido o grafo de produto directo entre dois grafos representativos de dois documentos é então possível calcular a função de núcleo (equação 1) que nos dará o grau de semelhança semântica entre esses documentos. Calculada a matriz de núcleo, que não é mais que o valor da função de núcleo para todos os pares de documentos resta utilizar as máquinas de vectores de suporte para optimizar a margem separadora entre as classes para obter o classificador de documentos. A Figura 5 ilustra o processo de obtenção desse classificador.

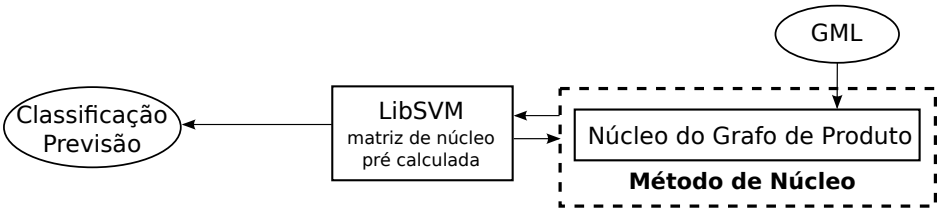


Figura 5: Processo de obtenção do classificador de textos

O LibSVM [3], a ferramenta utilizada para obter o classificador de textos, é uma implementação de máquinas de vectores de suporte para classificação, regressão e estimativa de distribuição, que permite a utilização de uma matriz de núcleo pré-calculada.

4 Experiências realizadas

Para avaliar a adequação quer das ferramentas que permitem obter a informação semântica dos documentos, quer da utilização dessa informação na construção de classificadores de texto foram realizadas um conjunto de experiências. Esta secção descreve e avalia essas experiências.

4.1 Conjunto de textos

Para avaliação da proposta utilizaram-se textos do corpus mais utilizado em investigação na área de classificação de textos – o conjunto `Reuters-21578`[8].

Para a análise do desempenho das ferramentas de obtenção da informação semântica escolheram-se aleatoriamente 150 textos de cada uma de cinco classes mais comuns do conjunto (`earn`, `acq`, `money-fx`, `grain` e `crude`. Para avaliar a proposta de utilização da informação semântica na criação de classificadores de texto, processaram-se 25 documentos de cada uma das classes `grain` e `crude` (também escolhidos aleatoriamente, mas com o cuidado dos documentos pertencerem apenas a uma das classes).

4.2 Representação semântica

Para gerar as ERD de cada documento utilizaram-se os parâmetros por omissão de todas as ferramentas envolvidas (NLTK, CANDC e BOXER).

O tempo médio de processamento por documento foi de 26,44 segundos e obteve-se uma taxa de sucesso média (relação entre o número de ERD obtidas e o número de textos analisados) de 83,58%. A Figura 6 mostra os valores dessas medidas por classe.

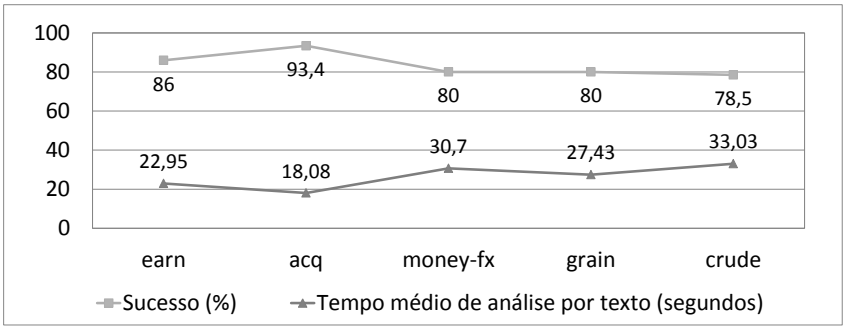


Figura 6: Resultados da representação semântica dos documentos

4.3 Classificador de textos

O núcleo do grafo do produto (equação 1) possui um parâmetro λ_n que pesa a matriz de adjacência E_x^n para percursos de tamanho n. Para este parâmetro considerou-se a série geométrica $\lambda_i = \gamma^{-i}$, com $\gamma = 2$, para percursos de tamanho menor ou igual a quatro (o que corresponde aos pesos de $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}$ para percursos de tamanho 1, 2, 3 e 4). Para percursos maiores considerou-se $\lambda_i = 0$.

As máquinas de vectores de suporte incluem um parâmetro de custo C que que controla o compromisso entre permitir a existência de erros de treino ou forçar margens de separação rígidas (sem erros).

Realizaram-se testes com validação cruzada de 5 pastas e diversos valores do parâmetro de custo. Para $C = 0.5$ obteve-se uma exactidão de 60%. Utilizaram-se também os mesmos documentos numa representação saco-de-palavras com validação cruzada de 5 pastas e obteve-se uma exactidão de 54%.

5 Conclusões e trabalho futuro

Este trabalho apresenta uma proposta para a utilização da informação semântica dos documentos no problema da classificação de textos. Embora o conjunto de experiências realizadas tenha sido limitado

[8]Disponível em `http://www.daviddlewis.com/resources/testcollections/reuters21578/`

pode concluir-se que este trabalho apresenta uma proposta válida já que:

- conseguiu-se representar a informação semântica numa estrutura passível de ser utilizada por funções de núcleo genéricas (núcleo do grafo do produto);
- o mapeamento de referentes permitiu a utilização desta função de núcleo por num algoritmo de máquinas de vectores de suporte;
- o classificador obtido tem valores de exactidão comparáveis com aqueles obtidos pela aproximação mais comum que utiliza uma representação saco-de-palavras.

No entanto, analisando os resultados obtidos individualmente pode concluir-se que a proposta é passível de melhoramentos pois subsistem muitos exemplos mal classificados. Estes melhoramentos passam por:

- rever a forma como é feito o mapeamento entre referentes. É necessário perceber qual a melhor forma de medir a semelhança entre os grafos gerados ERD dos textos analisados;
- rever os pesos atribuídos a percursos de diferentes comprimentos. É necessário avaliar a importância de percursos mais compridos dando-lhe pesos maiores (ao contrário da série geométrica utilizada em que os pesos penalizam os caminhos com maior comprimento)

Como trabalho futuro pretende-se avaliar outras formas de mapeamento de referentes. Uma das soluções poderá passar por considerar que as etiquetas dos nós passem a ser as etiquetas dos arcos que começam e terminam no nesse nó (ignorando posteriormente estes arcos).

Devem, por outro lado, ser realizados mais testes alargando quer o número de documentos, quer o número de classes. A utilização de outros conjuntos de documentos deverá também ser considerada.

Relativamente à obtenção da informação semântica dos documentos, os resultados mostram alguma morosidade das ferramentas CANDC e BOXER. Em futuras versões deste conjunto de ferramentas, o tempo de execução e os erros referenciados poderão ser corrigidos.

Referências

[1] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*, volume 1. O'Reilly Media Inc, 2009.

[2] Steven Bird, Ewan Klein, Edward Loper, and Jason Baldridge. Multidisciplinary instruction with the natural language toolkit. *Proceedings of the Third Workshop on Issues in Teaching Computational Linguistics, ACL*, 2008.

[3] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[4] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the world wide web. *Proceedings of the Fifteenth National Conference on Artificial Intellligence (AAAI-98*, pages pp.509–516, 1998.

[5] James R. Curran and Stephen Clar. Language independent ner using a maximum entropy tagger. *In Proceedings of CoNLL-03*, pages pp.164–167, 2003.

[6] James R. Curran and Stephen Clark. Investigating gis and smoothing for maximum entropy taggers. *In Proceedings of the 10th Meeting of the EACL*, pages pp.91–98, 2003.

[7] James R. Curran, Stephen Clark, and & Johan Bos. Linguistically motivated large-scale nlp with c&c and boxer. *Proceedings of the ACL 2007 Demonstrations Session (ACL-07 demo)*, pages pp.33–36, 2007.

[8] Dan Garrette and Ewan Klein. An extensible toolkit for computational semantics. *Proceedings of the Eighth International Conference on Computational Semantics*, 2009.

[9] T. Gartner. Exponential and geometric kernels for graphs. *In NIPS-02, 16th Neural Information Processing Systems*, Workshop on Unreal Data: Principles of Modeling Nonvectorial Data, 2002.

[10] T. Gartner, P. Flach, and S. Wrobel. On graph kernels: Hardness results and efficient alter-natives. *Proc. Annual Conf. Computational Learning Theory*, pages pp.129–143, 2003.

[11] T. Gonçalves. *Utilização de Informação Linguística na classificação de documentos em Língua Portuguesa*. Phd, Universidade de Évora, Évora, Portugal, 2007.

[12] I. Heim. *The Semantics of Definite and Indefinite Noun Phrases*. Phd, University of Massachusetts, New York, 1982.

[13] M. Himsolt. Gml: A portable graph file format. *Technical Report, Universität Passau*, 1997.

[14] W. Imrich and S. Klavzar. *Product Graphs: Structure and Recognition*. John Wiley, 2000.

[15] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. *In Machine Learning: ECML-98, Tenth European Conference on Machine Learning*, pages pp.137–142, 1998.

[16] H. Kamp. A theory of truth and semantic representation. *J.A.G. Groenendijk*, 1981.

[17] Han Kamp and Uwe Reyle. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, volume Studies in Linguistics and Philosophy, Vol. 42. Springer, 1993.

[18] H. Kashima and A. Inokuchi. Kernels for graph classification. *In ICDM-02, IEEE International Conference on Data Mining - Workshop on Active Mining*, 2002.

[19] L. Lang. Newsweeder: Learning to filter netnews. *In Machine Learning: Proceedings of the Twelfth International Conference (ICML '95)*, pages pp.331–339, 1995.

[20] D.D. Lewis and K.A. Knowles. Threading electronic mail: A preliminary study. *Information Processing and Management*, pages pp.209–217, 1997.

[21] Guido Minnen, John Carroll, and Darren Pearce. Applied morphological processing of english. *Natural Language Engineering*, pages pp.207–223, 2001.

[22] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Trans. Neural Netw*, pages pp.181–201, 2001.

[23] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A baysian approach to filtering junk e-mail. *AAAI-98 Workshop on Learning for Text Categorization. Tech. rep. WS-98*, 1998.

[24] Vladimir Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*.

[25] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. New York, 1995.

Reconhecimento de Entidades Nomeadas com SVM

Nuno Miranda, Ricardo Raminhos, Pedro Seabra
VIATECLA SA
nmiranda@viatecla.com, rraminhos@viatecla.com, pseabra@viatecla.com

João Sequeira, Teresa Gonçalves, Paulo Quaresma
Universidade de Évora
m5071@alunos.uevora.pt, tcg@uevora.pt, pq@uevora.pt

Resumo

Com a geração crescente de conteúdos textuais no mundo digital através de múltiplos criadores e distribuidores de informação (de qualidade variável) é cada vez mais difícil e complexo adquirir, manter e assimilar informação relevante que não esteja previamente tratada, catalogada e referenciada. A necessidade de extracção de conhecimento através de palavras-chave e a criação de ligações relacionais entre conteúdos com palavras-chave comuns constitui assim uma mais-valia importante em qualquer sistema de gestão de conteúdos. Devido à quantidade de informação envolvida é impossível que a extracção de palavras-chave seja feita de forma manual; este trabalho apresenta um protótipo desenvolvido para o reconhecimento de entidades nomeadas em documentos escritos na língua Portuguesa com recurso a técnicas de Aprendizagem Automática.

1 Introdução

Com a crescente informação presente na Web torna-se, muitas vezes, difícil navegar de forma objectiva sobre a informação relevante. A necessidade de extracção de conhecimento através de palavras-chave e a criação de ligações relacionais entre conteúdos com palavras-chave comuns constitui assim uma mais-valia importante em qualquer sistema de gestão de conteúdos.

Devido à quantidade de informação envolvida é impossível que a extracção de palavras-chave seja feita de forma manual (considere-se o simples exemplo das publicações diárias *online* dos vários jornais nacionais, que perfazem algumas centenas de notícias diárias). Surge assim a necessidade de criar uma ferramenta que consiga reconhecer entidades em conteúdos textuais.

Este reconhecedor de entidades poderá ser usado *per si* como uma aplicação autónoma ou embutido, como auxiliar, noutras aplicações que façam uso das entidades ou das relações entre elas. A sua utilização, poderá servir para:

- reconhecer as entidades e apresentá-las ao utilizador
- classificar o assunto geral dos textos de acordo com as entidades envolvidas
- permitir 'navegar' no conjunto de documentos, através das relações criadas a partir das entidades comuns entre diferentes conteúdos.

A ferramenta deve identificar entidades nomeadas e classificá-las numa das seguintes categorias: Pessoa, Organização ou Local. Para ser utilizável em diferentes contextos deve ser independente do domínio e da língua na qual os documentos estão escritos. A utilização de técnicas de Aprendizagem Automática vai de encontro a estes objectivos. Este trabalho apresenta os resultados obtidos num contexto de notícias de vários jornais nacionais: OJE [14], Público [18] e Record [6], onde a língua será obviamente o Português.

Apoiado no âmbito do QREN TV.COMmunity para a região do Alentejo, o protótipo foi desenvolvido pela Viatecla em colaboração com o Departamento de Informática nas instalações do Laboratório de Excelência .NET existente na Universidade de Évora. Este laboratório surgiu inicialmente de um protocolo tripartido entre a VIATECLA, a Universidade de Évora e a Microsoft e visa introduzir experiência

e conhecimento académico em contexto empresarial. Neste sentido o Laboratório surge como um espaço multi-disciplinar que conta com a presença de docentes, investigadores e alunos.

O artigo está organizado da seguinte forma: a Secção 2 introduz a área do reconhecimento de entidades nomeadas e a Secção 3 apresenta a arquitectura do sistema desenvolvido. A descrição das experiências realizadas e a sua avaliação são apresentadas na Secção 4 e as conclusões e trabalho futuro são discutidas na Secção 5.

2 Reconhecimento de Entidades Nomeadas

O Reconhecimento de Entidades Nomeadas[1] é uma tarefa de Extracção de Informação que pretende identificar e classificar elementos no texto que referem categorias predefinidas como nomes de pessoas, organizações, locais, expressões de tempo, quantidades, valores monetários, percentagens, etc.

Existem diversas aproximações ao problema [16], desde sistemas que utilizam regras definidas por humanos [1] até sistemas baseados em técnicas de aprendizagem automática [15] através da utilização algoritmos de classificação como árvores de decisão [2] e máquinas de vectores de suporte[2] [21]. A classificação automática utiliza um processo de inferência genérico (normalmente designado por aprendizagem) a partir do qual é construído, de forma automática, um classificador.

Durante a fase de aprendizagem são facultados vários exemplos de textos com entidades nomeadas classificadas manualmente a partir dos quais o algoritmo infere as características que definem essas entidades. Esta inferência é feita utilizando um conjunto de atributos que caracterizam cada uma das palavras existentes no texto. Como resultado é obtido um modelo de conhecimento que resume as regras de identificação de entidades nomeadas. Na fase posterior de classificação, é utilizado o modelo obtido e são submetidos novos textos onde são reconhecidas as entidades nele existentes.

3 Arquitectura da aplicação

A arquitectura do protótipo implementado pode ser observada na Figura 1.

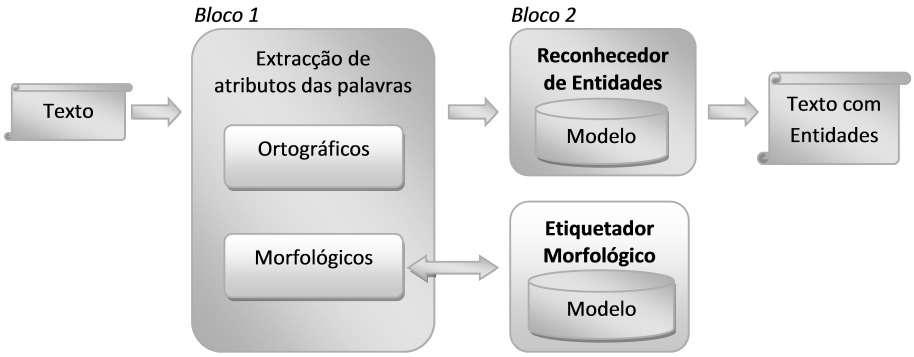


Figura 1: Arquitectura da aplicação

Existem dois blocos principais: o *Bloco 1* é responsável pela extracção dos atributos ortográficos e morfológicos de cada palavra; o *Bloco 2* é responsável por identificar e classificar entidades nomeadas no texto. Os atributos morfológicos são obtidos a partir de um **Etiquetador Morfológico** de palavras,

[1] Do inglês, *Named Entity Recognition*.
[2] Do inglês, *Support Vector Machines*.

baseado no SVMTool [9], que, juntamente com os atributos ortográficos, são utilizados no *Bloco 2* para construir o modelo do Reconhecedor de Entidades.

O SVMTool é um gerador de etiquetas sequencial baseado em máquinas de vectores de suporte [7]. Por se tratar de um algoritmo de classificação automática, para ser utilizado é necessário ter um modelo para a língua na qual os textos estão escritos. Como entre os modelos disponíveis (espanhol, catalão e inglês) não se inclui o da língua Portuguesa foi necessário desenvolver tal modelo.

Como o Etiquetador Morfológico, o **Reconhecedor de Entidades** utiliza máquinas de vectores de suporte. Este algoritmo foi escolhido entre árvores de decisão [17], variantes do algoritmo naïve de Bayes [4, 24] e máquinas de vectores de suporte com várias funções de núcleo [10]. Na escolha consideraram-se valores de precisão, cobertura e eficácia e as plataformas disponíveis de implementação dos algoritmos. Os testes de selecção foram efectuados com a ferramenta WEKA [23] mas a implementação do protótipo utilizou o LibSVM [5], uma ferramenta que implementa diversos tipos de máquinas de vectores de suporte.

Para criar o modelo de suporte ao Reconhecedor de Entidades, cada palavra do documento foi caracterizada utilizando atributos quer ortográficos, quer morfológicos e uma janela de contexto. As próximas subsecções descrevem esses atributos.

3.1 Atributos Ortográficos

Os atributos ortográficos, num total de 25 atributos binários, foram extraídos das características ortográficas das palavras. A Tabela 1 enumera os atributos considerados.

Atributos Ortográficos			
'!'	'?'	palavra só maiúsculas	carácter único
'('	')'	palavra só minúsculas	palavra alfanumérica
'"'	'''	palavra c/ inicial maiúscula	palavra numérica
';'	':'	maiúsculas e minúsculas	palavra c/ letras
'+'	'-'	inicial sozinha	palavra com hífen
','	'.'	palavra 'e'	pontuação fim de frase
'≪' ou '≫'			

Tabela 1: Atributos ortográficos das palavras

3.2 Atributos Morfológicos

Os atributos morfológicos, num total de 15 atributos binários, indicam a classe gramatical da palavra. A Tabela 2 enumera esses atributos.

Atributos Morfológicos			
nome comum	determinante	conjunção	contracção (determinante)
nome próprio	advérbio	prefixo	contracção (advérbio)
verbo	pronome	interjeição	contracção (pronome)
adjectivo	preposição	pontuação	

Tabela 2: Atributos morfológicos das palavras

3.3 Janela de Contexto

De modo a considerar o contexto da palavra no texto, ao conjunto dos atributos descritivos da palavra juntaram-se os atributos descritivos das palavras vizinhas (anteriores e posteriores), criando uma *janela de contexto*. Por exemplo, numa janela com dimensão 5, são considerados os atributos ortográficos e morfológicos de cinco palavras: a palavra em análise encontra-se na posição central com duas palavras posteriores e duas anteriores.

4 Experiências e Avaliação

Esta secção descreve os corpora utilizados, a configuração experimental e os resultados obtidos para o Etiquetador Morfológico e o Reconhecedor de Entidades.

4.1 Etiquetador Morfológico

Como referido na Secção 3, o Etiquetador Morfológico foi obtido utilizando a ferramenta SVMTool. Tratando-se de uma ferramenta que utiliza técnicas de Aprendizagem Automática foi necessário desenvolver um modelo para a língua Portuguesa. As próximas subsecções descrevem o corpus utilizado, a configuração experimental e os resultados obtidos.

4.1.1 Corpus

O corpus foi construído a partir do Bosque 8.0 [12], uma colecção de frases analisadas sintacticamente pela ferramenta Palavras [3] e revista manualmente por linguistas. O Bosque 8.0 foi desenvolvido pela Linguateca (um centro de recursos para o processamento computacional da língua portuguesa) e é composto por 9368 frases retiradas dos primeiros 1000 extractos dos corpora CETENFolha e CETEMPúblico (corpora com notícias do Jornal Público [18] e do Jornal Folha de São Paulo [8], respectivamente).

Sobre este corpus inicial retiraram-se as frases do CETENFolha e fizeram-se alterações para o Etiquetador funcionar directamente sobre o texto, ou seja, cada palavra do texto dar origem a uma etiqueta, característica que não acontece no corpus Bosque 8.0 já que nele as contracções são expandidas e as palavras integrantes de sintagmas preposicionais não estão classificadas. Estas alterações obrigaram a alguma etiquetagem manual.

Como se pretendia utilizar um léxico da língua Portuguesa, utilizaram-se as etiquetas do LabelLex [11], um léxico produzido pelo Laboratório de Engenharia da Linguagem do Instituto Superior Técnico.

Estas alterações deram origem a um corpus com 133 670 palavras e 18 etiquetas morfológicas. A Tabela 3 apresenta a percentagem de palavras para cada uma das principais classes gramaticais.

n_comum	n_próprio	verbo	adjectivo	determinante	pronome	advérbio	preposição
18.4%	8.5%	12.2%	5.0%	7.6%	5.3%	4.6%	9.4%

Tabela 3: Percentagem de palavras das principais classes gramaticais

4.1.2 Configuração experimental

O Etiquetador, criado com o SVMTool [9], foi obtido utilizando os valores por omissão para todos os parâmetros daquela ferramenta. O corpus foi dividido utilizando 2/3 das frases como treino (2983 frases) e as restantes como teste (1492 frases).

O desempenho foi avaliado através das medidas de precisão, cobertura e F_1 [19].

4.1.3 Resultados

Para as classes gramaticais principais (nome comum, nome próprio, verbo, adjectivo, determinante, pronome, advérbio e preposição) obtiveram-se valores médios de 96%, 94% e 95% para a precisão, cobertura e F_1, respectivamente. A Tabela 4 apresenta os valores máximo, mínimo, médio e desvio padrão das três medidas para essas classes.

	máximo	**mínimo**	**média**	**desvio padrão**
Precisão	.986	.886	.958	.040
Cobertura	.991	.891	.941	.068
F_1	.988	.888	.949	.053

Tabela 4: Variação dos valores de precisão, cobertura e F_1 do Etiquetador

A partir da tabela é possível observar que os valores obtidos são similares aos valores obtidos pela ferramenta para outras línguas e outros etiquetadores [9]. Entre as classes principais, e considerando os valores de precisão e cobertura, aquela que apresenta melhores valores é a dos nomes próprios; por outro lado, aquela com valores mais baixos é a dos adjectivos.

4.2 Reconhecedor de Entidades

Como já referido, o Reconhecedor de Entidades utiliza máquinas de vectores de suporte como algoritmo de aprendizagem, onde cada exemplo caracteriza uma palavra. Essa caracterização é feita através de um conjunto de atributos ortográficos e morfológicos. Assim, o classificador irá indicar, para cada palavra apresentada, se pertence ou não a uma entidade e, em caso afirmativo, qual o tipo (`Pessoa`, `Organização`, `Local`).

Como uma entidade pode ser composta por várias palavras utilizou-se a aproximação designada por IOB, normalmente utilizada para marcar expressões multi-palavra (ver por exemplo [22]). A etiqueta `B` (*begin*) marca a primeira da expressão e as restantes são marcadas com a etiqueta `I` (*inside*). As palavras que não pertencem à expressão são marcadas com a etiqueta `O` (*outside*). Como existem 3 tipos de entidade, isto corresponde a definir um problema de classificação multi-classe com um total de 7 classes (`O` e `B-xxx` e `I-xxx` para cada tipo de entidade).

Mais ainda, como a classificação é feita palavra a palavra é necessário reconstruir as entidades à saída do classificador. Para tal considerou-se entidades o mais extensas possível (sem sobreposição nem inclusão de outras entidades).

4.2.1 Corpus

O corpus utilizado para o desenvolvimento do Reconhecedor foi construído a partir de uma selecção de várias notícias dos jornais OJE, Público e Record marcadas manualmente segundo os tipos de entidades nelas encontradas (`Pessoa`, `Organização`, `Local` e `Outra`). Este corpus é constituído por 255 documentos num total de 3645 frases e cerca de 88000 palavras. Nele foram marcadas um total de 1267 pessoas, 2507 organizações e 1041 locais.

4.2.2 Configuração experimental

O Reconhecedor de Entidades, criado utilizando o SVMLib [5], foi obtido utilizando os valores por omissão para todos os parâmetros. Embora o corpus tenha sido marcado com 4 tipos de entidades, o Reconhecedor foi avaliado apenas para as 3 entidades de interesse (`Pessoa`,`Organização` e `Local`).

Mais uma vez, utilizou-se 2/3 das frases como treino e as restantes como teste. A Tabela 5 apresenta a distribuição de entidades pelos conjuntos de treino e teste.

	Treino	**Teste**
Pessoa	803	464
Organização	1581	926
Local	691	350
Total	3075	1740

Tabela 5: Distribuição de entidades nos conjuntos de treino e teste

Fizeram-se diversas experiências de modo a obter os melhores resultados (tanto de desempenho como de eficácia). Experimentaram-se diversas janelas de contexto (dimensões entre 1 e 15) e, além dos atributos ortográficos e morfológicos, experimentou-se utilizar outros 3 atributos binários com indicação de pertença da palavra a cada uma das seguintes listas de palavras:

- *léxico da língua Portuguesa*, com cerca de 940000 palavras (Label-Lex [11])
- *palavras funcionais*, com 24 palavras (palavras minúsculas que aparecem em entidades)
- *palavras raras*, com 4995 palavras (palavras que aparecem em menos de 4 documentos)

Criou-se também um classificador cuja função foi identificar as entidades presentes no texto sem as classificar segundo `Pessoa`, `Organização` ou `Local`. A este classificador deu-se o nome de **Identificador de Entidades**.

À semelhança do Etiquetador, o Reconhecedor e o Identificador foram avaliados utilizando as medidas de desempenho precisão, cobertura e F_1.

4.2.3 Resultados

Para o Identificador de Entidades os melhores resultados foram obtidos com uma janela de dimensão cinco sem utilização dos atributos de pertença às diversas listas de palavras. Os melhores resultados para o Reconhecedor corresponderam a uma janela de contexto de dimensão sete com a utilização dos três atributos relativos às listas de palavras.

As Tabelas 6 e 7 apresentam os resultados obtidos em cada um dos jornais para o Identificador e Reconhecedor, respectivamente. Como esperado, os resultados obtidos para o Identificador (precisão e cobertura acima de 81%) são superiores aos obtidos pelo Reconhecedor (precisão e cobertura entre 56% e 65%), uma vez que à partida este último é à partida um problema mais complexo.

	Precisão	**Cobertura**	**F_1**
OJE	.814	.852	.833
Publico	.832	.856	.844
Record	.818	.840	.829
Média	.821	.849	.835

Tabela 6: Precisão, cobertura e F_1 do Identificador de Entidades (por jornal)

Em ambas as tabelas é possível observar que os valores das medidas de desempenho não variam muito entre os jornais, excepto para o jornal OJE e o Reconhecedor de Entidades cujos valores das medidas de desempenho são consideravelmente mais baixos.

A Tabela 8 apresenta os valores de precisão, cobertura e F_1 obtidos pelo Reconhecedor para cada um dos tipos de entidades considerados. Nela, é possível observar que a classe mais difícil de reconhecer é `Organização` com um valor de F_1 muito perto de 50%, enquanto `Local` possui um valor de 74%.

	Precisão	**Cobertura**	**F_1**
OJE	.565	.595	.576
Publico	.650	.634	.639
Record	.654	.652	.640
Média	.623	.627	.618

Tabela 7: Precisão, cobertura e F_1 do Reconhecedor (por jornal)

	Precisão	**Cobertura**	**F_1**
Pessoa	.582	.701	.636
Organização	.497	.492	.494
Local	.774	.708	.740

Tabela 8: Precisão, cobertura e F_1 do Reconhecedor de Entidades (por tipo de entidade)

5 Conclusões e Trabalho Futuro

Este artigo apresenta um sistema para Reconhecimento de Entidades Mencionadas utilizando técnicas de Aprendizagem Automática. Embora tenha sido aplicado à língua Portuguesa, o sistema é independente da língua e do domínio, bastando para tal modificar o corpus de aprendizagem para a língua e/ou domínio escolhidos.

Os resultados obtidos estão dentro dos valores apresentados na comunidade científica para o mesmo tipo de problema [16, 22].

Como trabalho futuro pretende-se comparar este sistema com outros sistemas desenvolvidos para a língua Portuguesa utilizando os recursos disponibilizados pelo HAREM [13], uma avaliação conjunta na área do reconhecimento de entidades mencionadas em Português promovido pela Linguateca. Estes resultados encontram-se descritos em [20].

Para que a comparação com outros sistemas seja mais objectiva, pretende-se testar o sistema noutros domínios que não os das notícias diárias e noutras línguas nomeadamente a língua inglesa.

Com vista a melhorar o sistema, pretende-se criar melhores reconhecedores, quer através da utilização de corpora maiores, quer adicionado novos atributos que se considerarem críticos ao problema.

Referências

[1] J. Aberdeen, J. Burger, D. Day, L. Hirschman, P. Robinson, and M. Vilain. MITRE: description of the Alembic system used for MUC-6. In *MUC6 '95: Proceedings of the 6th Message Understanding Conference*, pages 141–155, Morristown, NJ, USA, 1995. Association for Computational Linguistics.

[2] S. Baluja, V.O. Mittal, and R. Sukthankar. Applying machine learning for high performance named-entity extraction. In *In Proceedings of the Conference of the Pacific Association for Computational Linguistics*, pages 365–378, 2000.

[3] Eckhard Bick. *The Parsing System "Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. PhD thesis, Aarhus University, Aarhus, Denmark, November 2000.

[4] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 161–168, New York, NY, USA, 2006. ACM.

[5] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[6] Edisport S.A. Cofina Media Grupo Cofina. Record. `http://www.record.xl.pt/`.

[7] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[8] Folhapress. Folha.com. `http://www.folha.uol.com.br/`.

[9] J. Giménez and L. Màrquez. SVMTool: A general POS tagger generator based on Support Vector Machines. In *Proceedings of the 4th LREC*, 2004.

[10] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, and K.R.K. Murthy. Improvements to Platt's SMO Algorithm for SVM Classifier Design. *Neural Comput.*, 13 (3):637–649, 2001.

[11] IST Laboratório de Engenharia da Linguagem. LABEL-LEX. `http://label.ist.utl.pt/pt/labellex_pt.php`.

[12] Linguateca. Bosque 8.0. `http://www.linguateca.pt/floresta/corpus.htmll#bosque`.

[13] Linguateca. Harem. `http://www.linguateca.pt/aval_conjunta/HAREM/`.

[14] Sociedade Editorial S.A. Megafin. Oje. `http://www.oje.pt/`.

[15] T.M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[16] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30 (1):3–26, 2007. Publisher: John Benjamins Publishing Company.

[17] R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, US, 1993.

[18] PÚBLICO Comunicação Social SA. Público. `http://www.publico.pt/`.

[19] G. Salton, A. Wang, and C. Yang. A vector space model for information retrieval. *Journal of the American Society for Information Retrieval*, 18:613–620, 1975.

[20] D. Santos and N. Cardoso, editors. *Reconhecimento de entidades mencionadas em português: documentação e actas do HAREM, a primeira avaliação conjunta na área*. Linguateca, 2007.

[21] K. Takeuchi and N. Collier. Use of support vector machines in extended named entity recognition. In *COLING-02: proceedings of the 6th conference on Natural language learning*, pages 1–7, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[22] E.F. Tjong and K. Sang. Introduction to de CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of CoNLL-2002*, pages 155–158, Taipei, Taiwan, 2002.

[23] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, US, 2nd edition, 2005.

[24] H. Zhang. The optimality of naive bayes. In *Proceedings of the 17th International FLAIRS conference*. AAAI Press, 2004.

Marcação de Nomes Próprios em Português recorrendo a diferentes fontes de conhecimento na Internet

João Laranjinho
Universidade de Évora
Évora, Portugal
joao.laranjinho@gmail.com

Irene Rodrigues
Universidade de Évora
Évora, Portugal
ipr@di.uevora.pt

Abstract

Neste artigo apresenta-se um sistema, independente do domínio, para marcação de nomes próprios para o Português. Este sistema é avaliado de forma a estudar o impacto de diferentes fontes de conhecimento nos resultados do sistema. O marcador usa informação morfo-sintáctica, sintáctica e semântica. A informação morfo-sintáctica vem de um dicionário local que completa a sua informação recorrendo a dicionários on-line como o da Priberam. A informação sintáctica das frases vem de uma gramática construída para analisar frases interrogativas. A informação semântica usada nas experiências de avaliação vem da Wikipédia. Na avaliação do sistema e do impacto das diferentes fontes de informação usaram-se três corpus distintos: um conjunto de documentos com 700 perguntas do CLEF; 300 frases de notícias do Público; e 200 frases do corpus CD do segundo HAREM.

1 Introdução

Os sistemas de marcação de nomes próprios são um caso particular (a identificação) dos sistemas de reconhecimento de entidades mencionadas que identificam e classificam as entidades de acordo com uma hierarquia pré-definida de categorias. Para o Inglês, alguns dos sistemas actuais conseguem um valor de 93% na medida F em fóruns de avaliação como o MUC-7[3]. Para o Português, no segundo HAREM [1], o melhor sistema, o sistema da Priberam[2] consegue atingir cerca de 71% na medida F na identificação de entidades mencionadas (ou marcação de nomes próprios).

O Sistema que apresentamos foi, inicialmente, desenvolvido para melhorar o desempenho de um sistema de Pergunta-Resposta[4], em particular para auxiliar na escolha das melhores análises sintácticas de uma pergunta em Português. A avaliação inicial do sistema foi feita usando um corpus anotado manualmente com as perguntas do CLEF [2] na tarefa de Pergunta-Resposta[5] de várias edições. O sistema usa três tipos de fontes de conhecimento para decidir a marcação de nomes próprios:

- Informação morfo-sintáctica — num dicionário local que completa a sua informação recorrendo a dicionários on-line como o Dicionário da Priberam [3] que está disponível via Web.
- Informação sintáctica — o resultado de um analisador sintáctico (as estruturas sintácticas das frases com os nomes próprios marcados). Uma frase pode ter zero, uma ou mais estruturas sintácticas associadas. A estrutura sintáctica pode ser parcial ou total. Esta informação é usada para decidir a melhor marcação de nomes próprios na frase.
- Informação semântica — informação de dicionários e enciclopédias que indicam se o nome próprio existe nalgum contexto. No sistema por enquanto só se usa a informação da Wikipédia [4], que como se pode ver na avaliação do sistema tem um grande impacto na correcção das marcações.

José Saias, Luís Rato and Teresa Gonçalves (eds.): JIUE 2010, 2010, volume 1, issue: 1, pp. 1-6

[1] http://www.linguateca.pt/HAREM/

[2] http://www.linguateca.pt/aval_conjunta/CLEF/

[3] http://priberam.pt/dlpo/dlpo.aspx

[4] http://pt.wikipedia.org/wiki/Página_principal

As técnicas usadas nos sistemas de REM (Reconhecimento de Entidades Mencionadas) dividem-se em: modelos baseados em regras e modelos estatísticos. Actualmente a técnica dominante é a baseada em modelos estatísticos e aprendizagem. Estes sistemas requerem grandes quantidades de dados anotados manualmente para a fase de treino e alguns sistemas têm um desempenho dependente do domínio.

Os modelos baseados em regras têm apresentado melhores resultados (ver MUC-7), no entanto, requerem muito trabalho feito por linguistas na sua implementação e muitas vezes o seu desempenho depende do domínio.

Algum dos recursos utilizados no REM [1] são:

- Corpora — conjuntos de textos anotados. Normalmente em conjunto com corpora, são utilizadas estratégias de aprendizagem, como por exemplo: modelos de Markov não observáveis (Hidden Markov Models - HMM), árvores de decisão, modelos de máxima entropia, SVMs [3]. Um dos problemas dos corpora está relacionado com a dificuldade da sua anotação;
- Almanaques — dicionários de Entidades Mencionadas (EM);
- Meta-palavras — representam as palavras próximas das EM. Estas são palavras que dão alguma informação sobre as entidades, normalmente são utilizadas na fase de desambiguação. Exemplos: escritor, rua, etc;
- Abreviaturas — palavras que fazem parte das entidades e dão informação sobre a classe da entidade. Normalmente são utilizadas na classificação. Exemplo: Dr., Sr., etc;
- Regras de similaridade — um conjunto de regras que definem semelhanças entre as entidades a serem classificadas e as entidades que existem em almanaques.

2 Arquitectura do sistema

O sistema recebe um ficheiro de texto para marcação de nomes próprios e retorna um ficheiro de texto com os nomes próprios marcados. Para a marcação do nomes próprios recorre-se a regras que codificam as preferências na escolha dos nomes próprios. As regras usam informação sobre: O número de átomos do nome próprio; A primeira letra dos átomos (maiúscula ou minúscula); A classe morfo-sintáctica de cada átomo; e Alguns caracteres como: aspas, números e sinais de pontuação.

O sistema contém 4 módulos. Na Figura 1 é apresentada a arquitectura do marcador de nomes próprios. Os módulos são: pré-processamento, análise lexical, pontuar interpretações e saída. No Pré-processamento um texto separado em frases, sendo uma frase é constituída por palavras, números, sinais de pontuação, etc. Na separação de um texto utiliza-se uma estrutura que guarda, em cada posição, o conjunto de interpretações de uma frase. O sistema numa frase com N candidatos a nomes próprios produz 2^N interpretações.

Na Análise Lexical faz-se a consulta das características gramaticais de cada palavra dum texto. O Priberam[5] é um dicionário on-line, no qual podem ser feitas consultas sobre palavras através da Internet. Em cada consulta obtém-se a informação morfo-sintáctico-semântica da palavra pesquisada.

Na Heurística para Pontuar Interpretações os candidatos a nomes próprios foram classificados em:

- NP WIKI — candidato a nome próprio que tem entrada na Wikipédia;
- NP SIMPLES — candidato a nome próprio que não tem entrada na Wikipédia e que pertence a uma das seguintes classes gramaticais: substantivo comum, adjectivo ou nome próprio;

[5] http://priberam.pt/dlpo/dlpo.aspx

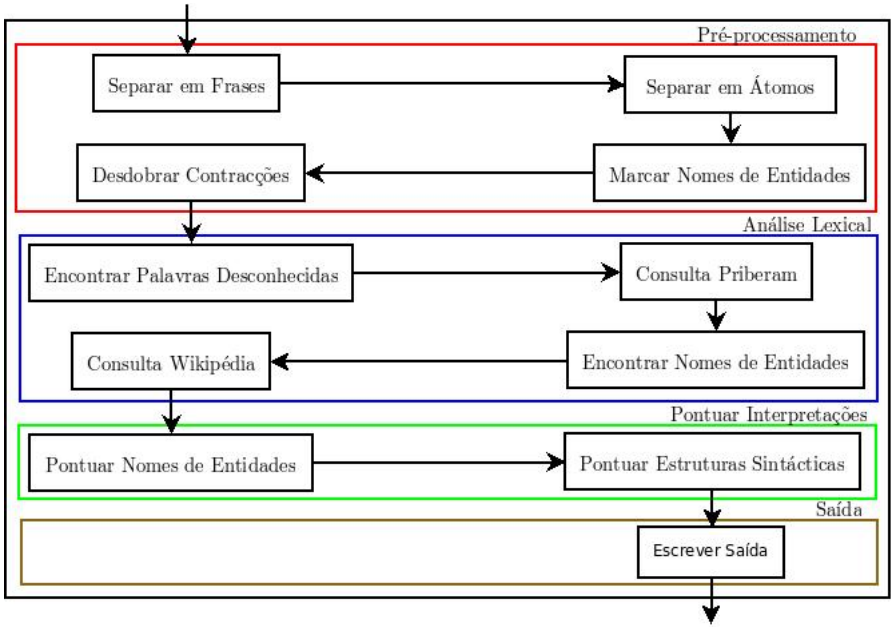


Figure 1: Arquitectura do marcador de nomes próprios

- NP COMPOSTO — candidato a nome próprio que não tem entrada na Wikipédia sendo constituído por vários átomos em que o primeiro pertence a uma das seguintes classes gramaticais: substantivo comum, adjectivo, verbo ou nome próprio;
- NP NUM — candidato a nome próprio que é um valor numérico;
- NP ASPAS — candidato a nome próprio delimitado por aspas (“”);
- NP DATA — candidato a nome próprio marcado como data;
- NP HORA — candidato a nome próprio marcado como hora;
- NAO NP — candidato a nome próprio que não reúne nenhuma das características anteriores.

As interpretações de frases que têm análise sintáctica, i.e. uma ou mais estruturas sintácticas, devem ser valorizadas face às que não têm. As interpretações com estruturas sintácticas foram classificadas em:

- TOTAL REP — interpretação totalmente representável por uma estrutura sintáctica;
- PARCIAL REP — interpretação parcialmente representável por uma estrutura sintáctica;

Para a análise sintáctica foi desenvolvido um analisador sintáctico recorrendo às DCG’s.

Na avaliação do marcador de nomes próprios utilizaram-se três métricas que são usadas na avaliação de Sistema de Recuperação de Informação: precisão, cobertura e medida F. Estas métricas foram adaptadas ao problema de marcação de nomes próprios.

3 Avaliação

Na avaliação do sistema de marcação de nomes próprios foram utilizados três corpora: 700 perguntas do CLEF, 300 notícias do jornal O Público [6] e 200 frases do HAREM. O corpus do CLEF foi usado para

[6] http://dossiers.publico.pt/

verificar até que ponto o uso da gramática desenvolvida para frases interrogativas tem impacto no desempenho do marcador de nomes próprios. O corpus do Público foi utilizado para testar o desempenho do sistema em frases para as quais a gramática tem um mau resultado, não consegue obter análise sintáctica em mais de 80% das frases do corpus. Os corpora do CLEF e do Publico foram anotados manualmente identificando os nomes próprios das frases, i.e sem classificação. O corpus do Segundo HAREM foi utilizado para obter uma avaliação independente.

Para cada corpora foram realizados oito testes, variando as pontuações atribuídas aos grupos de candidatos a nomes e ao facto de as frases terem análise sintáctica.

3.1 Heurística utilizada: pontuações atribuídas

Na Tabela 1 pode ver-se a heurística utilizada no Teste N com as pontuações (parâmetros) atribuídas aos candidatos a nomes próprios e as interpretações de frases representáveis por estruturas sintácticas. Na Tabela 2 pode ver-se as pontuações (valores numéricos) atribuídas nos 8 testes aos candidatos a nomes próprios e as interpretações de frases representáveis por estruturas sintácticas.

	Teste N
NP WIKI	NP1*N^{o}As
NP SIMPLES	NP2
NP COMPOSTO	NP3*N^{o}As
NP NUM	NP4*N^{o}As
NP ASPAS	NP5
NP DATA	NP6*N^{o}As
NP HORA	NP7*N^{o}As
NAO NP	NP8*N^{o}As
TOTAL REP	I1
PARCIAL REP	I2

Table 1: Heurística utilizada no Teste N

	Teste1	Teste2	Teste3	Teste4	Teste5	Teste6	Teste7	Teste8
I1	100	0	100	100	100	100	100	100
I2	60	0	60	60	60	60	60	60
NP1	10	10	0	10	10	10	10	10
NP2	4	4	4	5	4	4	4	4
NP3	9	9	9	5	9	9	9	9
NP4	-9	-9	-9	-9	0	-9	-9	-9
NP5	15	15	15	15	15	0	15	15
NP6	15	15	15	15	15	15	0	15
NP7	15	15	15	15	15	15	15	0
NP8	-10	-10	-10	-10	-10	-10	-10	-10
N^{o}As	N	N	N	0	N	N	N	N

Table 2: Pontuações atribuídas nos 8 Testes

Com estes testes pretendemos ver o impacto:

- Teste 2 — da análise sintáctica. O teste 2 não tem em conta a existência de estrutura sintáctica para a frase com os nomes próprios.
- Teste 3 — da Wikipédia. O teste 3 não tem em conta a informação sobre a existência de entrada para o nome próprio na Wikipédia.

- Teste 4 — do comprimento do nome próprio. No teste 4 não valorizamos o número de palavras do nome próprio.
- Teste 5 — dos números isolados. No teste 5 não valorizamos números isolados que sejam marcados como nomes próprios.
- Teste 6 — das expressões entre aspas. No teste 6 não valorizamos expressões entre aspas que sejam marcadas como nomes próprios.
- Teste 7 — das datas. No teste 7 não valorizamos expressões que sejam marcadas como datas.
- Teste 8 — das horas. No teste 8 não valorizamos expressões que sejam marcadas como horas.

	Prec	Cob	Med-F
Teste 1	0,9786	0,9796	0,9791
Teste 2	0,9715	0,9739	0,9727
Teste 3	0,9307	0,9320	0,9314
Teste 4	0,8956	0,9029	0,8993
Teste 5	0,9779	0,9796	0,9780
Teste 6	0,9786	0,9796	0,9791
Teste 7	0,9731	0,9754	0,9742
Teste 8	0,9786	0,9796	0,9791

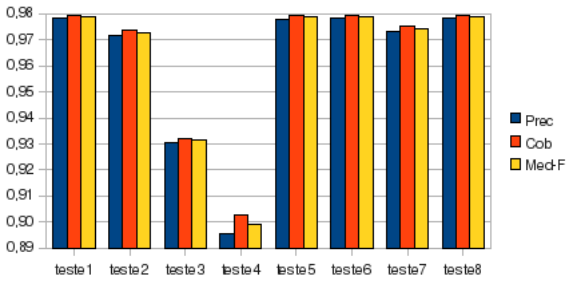


Figure 2: Testes com o corpus do CLEF

	Prec	Cob	Med-F
Teste 1	0,9615	0,9651	0,9633
Teste 2	0,9615	0,9651	0,9633
Teste 3	0,9278	0,9371	0,9324
Teste 4	0,7188	0,7851	0,7505
Teste 5	0,8961	0,9651	0,9293
Teste 6	0,9615	0,9651	0,9633
Teste 7	0,8670	0,9361	0,9002
Teste 8	0,9615	0,9651	0,9633

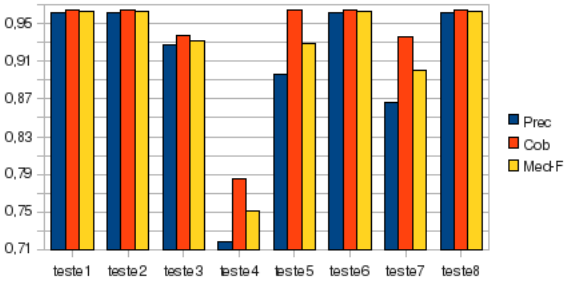


Figure 3: Testes com corpus do Público

4 Conclusões e Trabalho Futuro

Como se pode ver na secção 3, o teste 1 foi aquele que apresentou os melhores resultados nos 3 corpora. Neste teste valorizamos os nomes próprios que se encontram na Wikipédia, o comprimento dos nomes

	Prec	Cob	Med-F
Teste 1	0,7699	0,8069	0,7880
Teste 2	0,7649	0,8069	0,7854
Teste 3	0,7588	0,7965	0,7772
Teste 4	0,7160	0,7656	0,7400
Teste 5	0,7490	0,8069	0,7769
Teste 6	0,7699	0,8069	0,7880
Teste 7	0,7385	0,7983	0,7672
Teste 8	0,7699	0,8069	0,7880

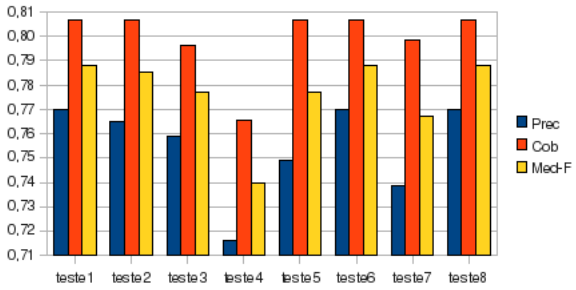


Figure 4: Testes com corpus do HAREM

próprios mesmo quando não temos entrada na Wikipédia, as expressões entre aspas, as datas e as horas. Além disso, também valorizamos as interpretações de frases que contêm estruturas sintácticas.

No teste 2 não temos em conta se as frases tem análise sintáctica. No corpus do CLEF, HAREM e Público a diferença na precisão é de 0.7%, 0.6% e nula respectivamente, entre o teste 1 e 2. Podendo concluir-se que o uso de uma gramática pode melhorar o desempenho do marcador, ainda que de forma ligeira e que o uso da gramática nunca baixa o desempenho do nosso sistema. O teste 3 demonstra que nos 3 corpora os resultados baixam: 5% na cobertura do CLEF, 3% na cobertura do Público e 1% na cobertura do HAREM. Permite-nos concluir que o uso de uma enciclopédia melhora os resultados. O teste 4 retira o peso ao comprimento dos nomes próprios. Os resultados deste teste nos diferentes corpora permitem concluir que escolher os nomes mais compridos é uma boa estratégia. O teste 7 não tem em conta os nomes próprios marcados como datas. Os resultados deste tese nos diferentes corpora permitem concluir que pontuar este tipo de nomes próprios é importante, visto ser a única forma de os detectar. Os outros testes mostram que esta informação não tem grande impacto no desempenho do sistema. Este sistema pode evoluir em diferentes direcções mas as mais imediatas incluem: utilizar uma gramática com maior cobertura, transportar o sistema para o Inglês e incluir a classificação das entidades mencionadas.

References

[1] Marcelo Adriano Amancio and Sandra Maria Aluísio. Explicitação de entidades mencionadas visando o aumento da inteligibilidade de textos em português. Technical report, Universidade de São Paulo, Agosto de 2008.

[2] Carlos Amaral, Helena Figueira, Afonso Mendes, Pedro Mendes, Cláudia Pinto, and Tiago Veiga. Adaptação do sistema de reconhecimento de entidades mencionadas da priberam ao harem. In Cristina Mota and Diana Santos, editors, *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*. Linguateca, 2008.

[3] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 1 (30):3–26, 2007.

[4] Paulo Quaresma, Irene Rodrigues, C. A. Prolo, and R. Viera. Um sistema de pergunta-resposta para uma base de documentos. *Letra de Hoje - Revista da Pontifíca Universidade Católica do Rio Grande do Sul*, 41 (2):43–63, Junho 2006.

[5] Diana Santos and Paulo Rocha. Chave: topics and questions on the portuguese participation in clef. In C. Peters and F. Borri, editors, *Cross Language Evaluation Forum: Working Notes for the CLEF 2004 Workshop*, pages 639–648, Bath, UK, September 2004 2004.

Geographic entity recognition and classification over journalistic text

Pedro R. P. Fialho
University of Évora
Évora, Portugal
prpfialho@gmail.com

Abstract

This work is aimed at defining a prototype system able to identify and classify geographic entities named in general news articles from a Portuguese newspaper. The approach here defined classifies entities according to a lexicon of Portuguese administrative divisions which serves as input to an inference engine based on rules and linguistic assumptions. The solution is evaluated for accuracy and coverage compared to the Human observation of the news articles.

1 Introduction

As the Web becomes a global container of information on (mostly) every subject, the knowledge representations tend to be careless about the organization and parsing of the fundamental elements that describe knowledge: Natural (Human) language. The lack of care on natural language processing (NLP) is due to the ease of deployment of social/news networks (specially in the current *cloud computing* scenario) grouping information/users based on content/popularity of blogs, message boards and alike instead of concerning their technological attributes. So, ambiguity, misspelled words (although better now thanks to the evolution on Web browsers) and inconsistent concept description/reference are common issues among Web sourced knowledge bases.

With such growth of available subjects becomes obvious the need of domain dependent query methods to filter information. Although keywords have shown to be useful on any query system, the constantly growing community of technology-aware users urges the need for constraints on search engines to fulfill today's information needs. For such information retrieval, research on optimal domain specific knowledge management techniques (named entity recognition, knowledge modeling, etc.) has to be done.

The system here described focuses on the retrieval of geographic information from newspaper articles, this being one of the constraints useful for searching knowledge bases consisting of blocks of free text (eventually isolated from global context) given that the use of geographical references in this type of writing allows contextualized search/reading. The input articles belong to a general newspaper without any type of information filtering so this system operates in an open domain (such as the Web) which increases its complexity. However, here is proposed a relatively simple solution.

1.1 State of art

Much interest has been formed around geographical information retrieval[10] systems due to their use in contextualizing the Web navigation[8] and consequently providing better/localized suggestions during Web browsing. The increasing availability of information on the Web already motivated the appearance of systems with high refresh rate, large knowledge bases[5] and search engines statistics[1] to support inference.

[1]The use of statistics increases the certainty that a term belongs to some domain by analyzing the properties of search engine output (number of *hits*, search time, top results relevance).

Despite being an extensive knowledge base, the Web use on building an inference engine implies high computational complexity (for content handling and organization) as well as uncertainty about the facts extracted due to the open domain (eventually ambiguous) property. This type of knowledge is here considered complementary but recognized as useful for defining trust levels of inferred facts.

1.1.1 Gazeteers

Some of the most popular systems for geographic information retrieval use geographical locations dictionaries [15] called *gazetteers* (also used for other domains). Although there are solutions that allow automatically generation/update of these dictionaries[20], the efficiency of these systems usually depends on the reliability of the knowledge source used for *gazetteer* generation, the update frequency and the process of useful term selection, bearing in mind that an extensive set of terms (eventually) has ambiguity issues. *Gazetteers* are also being adapted to current knowledge representation techniques such as ontologies[7] in order to improve their scalability and shareability.

1.1.2 Rules and statistics

In addition to availability of information, the performance of a recognition engine for a particular domain depends on whether it's based on rules or statistical analysis[17] of previously annotated *corpus*. The efficiency of the latter depends on the relevance to the target subject of the *corpus* used in training a learning algorithm, so a rule-based approach can achieve results in more documents but with less quality in those that resemble more (according to the inference algorithm[17]) with the ones in the training set. Modern named entity recognition systems use machine learning techniques[12] (such as statistical analysis) due to their high level of development (no handcrafted algorithms) and feasible extension to other domains and/or types of writing.

Given the variety of techniques for named entity recognition, the development of hybrid systems is desirable as a way of reaching a reference system in entity recognition. There are already several such systems which describe techniques for general subjects[1, 13, 11, 4] and also specific domains[5].

Named entity recognition systems (geographically oriented or not) can participate on specific contests such as MUC[16] (multiple languages) or HAREM[9, 18] (Portuguese only) where several evaluation metrics define algorithm performance apart from the application domain[19].

1.2 Journalistic style

As a way to achieve a simple and autonomous solution, here is used a lexicon of geographic categories ('county', 'district', etc.) in a rule-based approach, which reduces the effectiveness of the entity recognition engine compared to that of systems using *gazetteers* and imposes some dependence of writing assumptions. Namely, in a journalistic type of writing is common the use of geographic categories to contextualize news, refer to locations previously mentioned or disambiguate names, since the impact of the news may depend on the size/relevance of the place where it occurs. The system here proposed is based on this assumption and applies to the Portuguese language/grammar and eventually to other languages of similar grammatical structure and type of writing.

Although knowledge of Portuguese is considered optional for understanding this research/work, it will be necessary in order to fully understand/relate the examples provided in section 3.

2 Solution

Identification and classification of geographical entities is here accomplished without additional information about the language used in the *corpus*. However, a lexicon of target administrative divisions (categories) for recognition is used as a source of knowledge. This way it's possible to constrain/filter an ordered set of terms with keywords of the target subject. Despite being a somewhat naive and primitive form of natural language processing, results in a simple and efficient solution easily extensible to other subjects by changing the lexicon.

The inference is based on ad-hoc rules which leads to some inflexibility when certain rules/assumptions of writing do not occur in the articles. An alternative would be to use statistical inference[17], which requires configuration and training of a learning algorithm but allows dynamic resolution of unpredicted types of writing. Still, ad-hoc rules are more appropriate for quality evaluation of a specific technique such as the one here presented.

2.1 Resolution

The implemented system looks for words (from the lexicon) 'suspected' of occurring surrounded by entities of the target subject. Entities are words beginning by a capital letter (uppercase word) and they're assumed to occur before or after the suspicious word in the context of a sentence, i.e. after identification of a lexicon string as prefix of a word in the text (suspected administrative division) the corresponding sentence is searched for the first uppercase word that precedes or follows the suspicious word.

For inference are used all the words of a sentence in the form of a list. The elements of this list are sets of characters (words) that do not include blank spaces or punctuation except for commas since these are useful in detecting sequences as described below.

2.1.1 Compound names

If any uppercase word surrounds the detected category, a process of detection of compound nouns (location names composed of more than one word) is started. This process extracts uppercase words close (with tolerance of one lowercase word, delimiter of the multiple names that compose a geographical location) to the first found surrounding word, generating a single string containing the full name of the (supposed) location. This kind of inference is applied in a right-to-left and left-to-right manner since the first surrounding word found may precede or follow the detected category.

2.1.2 Sequences

In case the tolerated lowercase word belongs to the set [e, ou] (logical conjunction and disjunction respectively — [and, or]) or the last word of the composed string ends with ',' the detection of compound nouns is ended and a process of detection of geographical entities in sequence is started.

Since the empty space acts as delimiter in the process of word extraction from a sentence, to detect the comma it must be followed by a blank space and occur as the last character of an uppercase word, since the last word in a location's name begins with a capital letter by spelling imposition.

The main technical difference of including a sequence detection is that the basic inference algorithm in Figure 1 is applied to a list of (compound) location names. The process of registration/update that follows is identical between application to a single location (initial system sketch shown in Figure 1) or a list of them (current status of the system — not included in the basic diagram shown in Figure 1).

The purpose of a refined analysis for detection of fully valid expressions (no incorrect/irrelevant characters attached) is to increase coverage of the system and exploit the capabilities of the defined

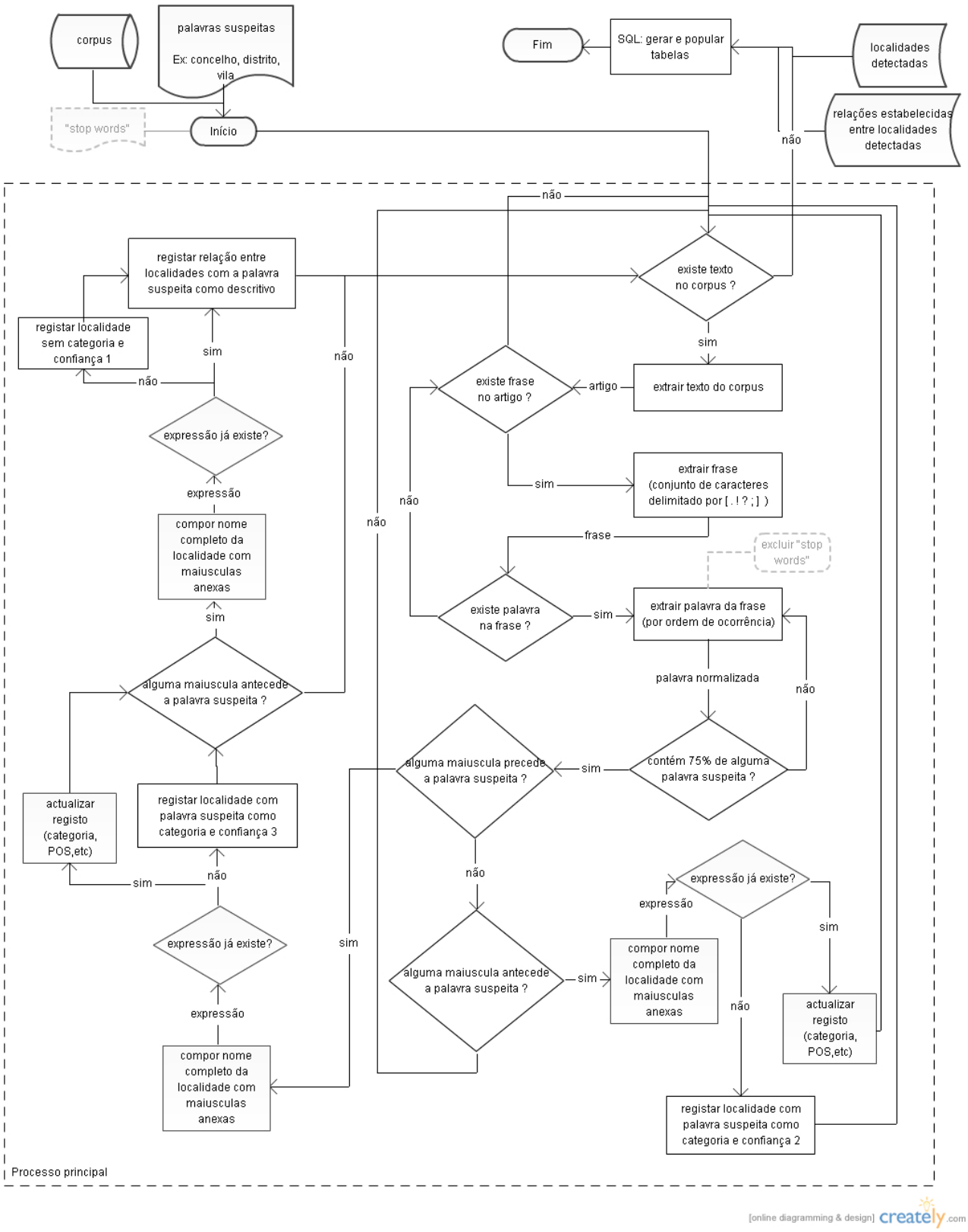


Figure 1: Workflow of the developed system without word sequence detection (labels in Portuguese).

inference rules. This refinement of the algorithm represents a high level of demand to the system since only strings that identify precisely the name of a geographical location are considered valid.

2.1.3 Grammatical form I

Due to the simplicity of rules and the journalistic nature of the text, the detected strings do not always match the name of a geographical location. On free text (such as the one in news articles) it's usual to state a geographical location in the (main) grammatical form <CATEGORY> [de,do,da,...][1] <NAME> and in following sentences use the form [o,a,neste,naquele,...][2] <CATEGORY> to refer to the previously stated location name.

This phenomenon represents the linguistic technique named **anaphora** (also used for writing this dissertation) whose resolution motivates some research[6]. For an anaphora to occur it's not necessary the presence of <CATEGORY> in the specified form (only [o,a,neste,naquele,...]). Here comes relevant the usage of *gazetteers* since the uppercase words on the anaphoric context would be excluded if not stated in these data structures. This system does not implement a solution for anaphoric contexts, as such in these sentences there is no guarantee that identified uppercase words correspond to geographical references.

During search for uppercase words that precede the detected category it is excluded the word that starts the sentence as this is always uppercase by traditional/cultural imposition. Despite the possibility of sentences being started with location names, these are a minority whose resolution through the stated approach generates too much entropy due to the excess of irrelevant records registered.

2.1.4 Grammatical form II

Registering uppercase words that precede the detected category (albeit without anaphora) also creates some errors in inference since these are not predicted by the main grammatical form to which is associated more confidence. However, (according to the grammar and observation of language) there is a way in which these expressions correspond to names of geographical location. Thus, to increase geographic coverage is assumed the grammatical form <NAME1> (...) <CATEGORY> [de,do,da,...] <NAME2> where <NAME1> is a location name contained in the set of locations (administrative division) that make up a <CATEGORY> named <NAME2>. This secondary form describes a relationship between entities (geographical context) which is useful for journalistic text since it removes ambiguity from locations with common names (different locations may have the same name) and better contextualize news about locations with unusual names.

This relationship is stored on an secondary/isolated data structure (separated from the registration of each location involved). Being separated, this registration allows us to analyze separately the situations where the secondary form applies in order to optimize this kind of detection.

The grammatical form I is a generic mean for detection of (uppercased) entities that are (usually) stated with some title/classification nearby, as such it is contained within the grammatical form II which represents an extension. The prepositions [de,do,da,...] referred on both forms do not belong to the algorithm and are only represented for enlightenment on typical language usage.

[1] The words 'de','do' and 'da' are preposition represented in English by 'of'.

[2] Mainly composed of grammar articles, the set with 'o','a','neste' and 'naquele' can be described in English roughly as 'the', 'on this' and 'on that'.

2.2 Storage

On the main data structure (only for location names), a geographical location is recorded along with the assigned category, the correspondent sentence (part-of-speech) and an initial level of trust according to the context in which it was detected. This level has a minimum value for a uppercase word that precedes the category (<NAME1> from the secondary form) when there is also an uppercase word that follows it. In this situation the preceding uppercase word is registered without category as the most reliable form (grammatical form I) has also been found, i.e. a category is only assigned to one of the (alleged) places that surround it but both are registered (separately).

The elements of this main data structure are updated as new information is inferred from the *corpus*. The textual information to update is added to the existing one in order to maintain an inference log. The trust level of a string (initialized to some static value1) is iterated on update. This iteration occurs only for a string that follows the category or, if none exist, for the string preceding the category.

Thus, this inference implies an order to the steps of the algorithm illustrated in Figure 1, with priority given to the identification of the main form. After identification of the main form, the secondary uppercase word (if any) is recorded without category or (if already registered) does not get updated.

At the end of system execution, these data structures are exported to a relational data model (represented on an SQL output). With this information applied to a DBMS, it's possible to perform queries over the knowledge base through a mature formal language (SQL) that allows a flexible viewing of the collected/inferred information.

3 Results

This system was applied separately to each of the suspicious words **distrito, concelho** and **freguesia** (district, county and civil parish respectively) in a sample of 581 newspaper articles from the *Público* newspaper resulting three SQL databases (one for each suspicious word) each with two tables/containers (classified locations and dependency relations between two locations). In case the input*corpus* gets 'substantially' increased (by thousands), the system is ready to receive a list of *stop words* (words with no semantic relevance) in order to enhance system performance through their exclusion.

3.1 Excess words

Although on defining both forms was only illustrated a single word (from the set [de,do,da,...]) separating <CATEGORY> from <NAME>, Example 1 in Figure 2 shows that more information (news related) can occur between a category and its instances. Thus, a rule defined for a specific objective/example turned out to be applicable to a wider range of situations, since there is no limit on the number of words that can occur between category and (supposed) location.

Also in Example 1 of Figure 2 is observable that 'António Guterres' (a Portuguese individual name) is a string registered as belonging to the category 'distrito' illustrating an error in inference caused by the unrestricted number of words between category and location name which is resolvable with *gazetteers*.

In addition to errors from inference, others occur from complex forms of writing as illustrated in Example 2 of Figure 2. This case illustrates a different use of commas, since these are not used as a separator in a sequence but rather to highlight facts that, in the grammatical forms set forth herein, would be referred in a proper sentence. In this example the resolution is possible by analyzing the grammatical number associated with categories.

Example 1 - Form <CATEGORY> (...) <NAME>:

'As menores subidas ocorreram em alguns dos **distrito**s de maior influência comunista, com destaque para **Beja** e **Setúbal**, no **distrito** de onde é natural **António Guterres**, **Castelo Branco**, e na **Madeira**.'

Example 2 - Form <CATEGORY> [de,do,da,...] <NAME> with sequence:

'A falta de abastecimento de água ao domicílio e os maus acessos rodoviários foram algumas das razões que levaram duas **freguesia**s do **concelho** da **Régua**, **Covelinhas** e **Sedielos**, a boicotar...'

Example 3 - Form <NAME1> (...) <CATEGORY> [de,do,da,...] <NAME2>:

'António Lopes, natural do lugar de **Lameiras**, **freguesia** de **Olalhas**, deixou a sua aldeia no princípio de Setembro, onde corriam vários abaixo-assinados ...'

Example 4 - Form <CATEGORY> (...) <NAME> incomplete:

'Os **distrito**s da **Guarda** (quatro boicotes), do Porto (três boicotes) e de Vila Real (dois boicotes) lideraram esta forma de protesto de populações insatisfeitas ...'

Example 5 - Inference error:

'Após mais dois anos de espera e de se ter visto obrigada a alugar instalações nas proximidades dos **Paços** do **Concelho**, a **Câmara de Alenquer** está disposta a tomar atitudes mais duras'

Figure 2: Examples of detected grammatical forms on an article of the input set with recorded words highlighted

3.2 Duplicates

There is a management of identified strings so that their characteristics (category, POS, etc) are updated whenever an identical string occurs. However, there is repetition of location names due to misspellings or incorrect composition of names originated by uppercase words placed next to the location's name, as illustrated in Example 5 of Figure 2 for 'Câmara de Alenquer' ('Alenquer' is the correct name and occurs within *corpus* in the predicted grammatical forms).

The existence of multiple records referring the same location (at most one is correct) implies a decrease in the algorithm's accuracy. However, given the simplicity of the inference rules, this system is satisfactory by the high coverage since accuracy can be achieved (essentially) by pattern matching with *gazetteers*.

For high coverage, term analysis with refined mechanisms (sequences, compound names) can detect correctly most geographical locations on text. However, these can be improved in cases such as the one illustrated in Example 4 of Figure 2 with isolated treatment of terms between contextual separators (parentheses, quotes, etc.).

3.3 Metrics

This system is evaluated by the accuracy and coverage of the results obtained having as reference a manual/Human term evaluation made on the sample. The performance of the system is evaluated separately for each possible administrative division for better result analysis.

Only the system's ability to detect locations correctly is evaluated. The feature that detects relationships between locations is not evaluated due to the scarcity of results generated by the basic implementation and experimental nature.

$$accuracy = \frac{\text{number of correctly classified districts}}{\text{total number of automatically classified districts}}$$

$$coverage = \frac{\text{number of correctly classified districts}}{\text{total number of manually classified districts}}$$

Figure 3: Definition of accuracy and coverage for the administrative division 'distrito'

Precision as defined in Figure 3 is the ratio of correct/valid terms among the total of terms automatically classified by the system. By definition, coverage as shown in Figure 3 shares the same numerator with the accuracy expression but the denominator here is the total number of terms manually classified in the sample.

Correctly classified according to the formula in Figure 3 means any string that is associated with the correspondent administrative division and maps precisely the name of the geographic location with no other character than those that describe the location name, though case is irrelevant.

To calculate the total number of terms automatically classified only are taken into account the ones with some administrative division associated, since there are cases where a location name is (correctly) recognized but no category is assigned. These cases are not taken into account because they're only relevant for geographical location recognition and the main target of this system is administrative division classification.

The process of manual/Human term detection and classification follows the rules defined by the algorithm (grammatical forms), i.e. only count locations manually identified by following the algorithm steps (though not/incorrectly classified) that have the correct category assigned. This aggregate does not include repeated or malformed location names, even though they contain correct information (Example 5 in Figure 2).

3.4 Discussion

From the observation of suspicious words (categories) usage, 'distrito' is the term less used in a context other than the reference to a locality, hence the total coverage shown in Figure 4. The suspicious words 'concelho' and 'freguesia' are used in the description/reference of other subjects (also location related — as in 'town' and 'town hall') like 'junta de freguesia' and 'paços do concelho'.

	accuracy	*coverage*
concelho	$\frac{43}{70} \approx 0.6$	$\frac{43}{46} \approx 0.9$
distrito	$\frac{18}{92} \approx 0.2$	$\frac{18}{18} = 1$
freguesia	$\frac{42}{89} \approx 0.5$	$\frac{42}{45} \approx 0.9$

Figure 4: Results for accuracy and coverage on the sample partitioned by category/suspicious word.

A quick analysis of accuracy and coverage values shown in Figure 4 suggests a system with high recognition capabilities but low filtering of entities not from the target subject. The high coverage is supported by the high level of trust assigned initially to the main grammatical form while the low accuracy motivates the use of some knowledge base as this approach is too naive in the conceptual definition of a geographical reference.

In fact, better solutions (without *gazetteers*) to distinguish a geographic location of any other proper name were not found. If less flexible constraints are applied to the inference rules (for example, limiting the number of terms that can appear between a category and the surrounding uppercase word) accuracy tends to increase inversely proportional to the coverage.

3.4.1 Tolerance

In a way, the evaluation of a system should not only be done by observation of the numbers that describe it (here accuracy and coverage) but also by analysis of the tolerance applied while counting valid terms.

Here, the exclusion of records from strings (albeit slightly) invalid in spelling or the exclusive accounting of records assigned to the category being tested (excluding records with no category) implies a very non-tolerant evaluation which accurately shows the expected value of the described techniques.

This system presents high coverage due to the heuristics established to describe the typical use of the input suspicious words. Not imposing a limit to the distance between uppercase and suspicious words also supports this high coverage but causes low accuracy in this non-informed approach because suspicious words are also used to refer geographical locations stated in other sentences (anaphora).

4 Conclusion

The developed system assumes that the use of suspicious words always happens (with or without anaphora) to refer a geographical entity, which is not a rule of language but rather an heuristic for recognizing relevant entities, since administrative division names can be used in a sentence which does not refer geographical location names.

The introduction of a measure of trust eases filtering valid results. However, this measure could be improved through the application of heuristics such as the definition of proportionality between the distance (in number terms) of a suspicious word to the surrounding uppercase word and the initial value of trust (such as $\frac{\text{initial trust}}{\text{number of terms}}$).

4.1 Non/other geographical entities

The suspicious words in the lexicon refer to geographic categories used for administrative divisions (county, district, etc.). On geographical subjects is also possible to detect entities relating to territorial dimension (city, village, etc.) as these are often used with the described grammatical forms. Since the prepositions [de,do,da,...] are optional, this approach is also valid for detecting geographical categories such as 'street' or 'avenue' where the (uppercase) location name immediately follows the category.

These geographical entities also apply compound names and sequences as well as their issues (excess words, duplicates, etc.). However, the name of a street or avenue usually starts with (ends with for the English grammar as in '49th Street') the category uppercased which can cause the preceding location name on grammatical form II to be invalid due to excess words on the compound name (always compound since the category is uppercased). As a fix for this issue an approach similar to *stop words* could be used to exclude administrative divisions from compound names.

The grammatical form II is intended for geographical hierarchy detection, as such it will be valid for streets or avenues that belong to cities or villages since both location names use uppercase words. However, detection of lower levels of the hierarchy like door numbers or postal codes that belong to a street or avenue is not possible since these entities use numbers instead of words. The introduction of a rule to also detect numbers that precede a category is possible but not applicable to a journalistic style since it's unusual (due to privacy issues) and often irrelevant to refer exact locations on news articles.

On other subjects the suspicious words correspond to an action in which entities are involved. For example, if the goal is to detect writers and their works, the lexicon could contain 'write', 'writing' or 'authorship', since these suggest (according to the grammatical forms assumed) the occurrence of writers names or titles of books surrounding them.

4.2 Punctuation

The approach here defined for recognizing location names written in sequence assumes that commas are always attached at the end of a word and followed by an blank space. To confirm this assumption, the substitution of '<word1> <word2>' for '<word1> <space> <word2>' and '<word1> <space> <space> <word2>' for '<word1> <space> <word2>' is easily attainable (using regular expressions) and applied to any set of texts by using modern text editors[14].

For cases in which additional information is described between elements of a sequence delimited by punctuation (usually parenthesis), is suggested an analysis of the punctuation involved for extraction of the 'sub-sentence' and recursively apply the same inference algorithm (eventually a specific one, depending on the grammatical forms observed) over these regions.

4.3 Syntactic parser

By integrating a syntactic parser in this system it's possible to identify more accurately if the uppercase word surrounding a suspicious word refers to it since a complete parser[3] is able to identify dependencies between the terms of a sentence, avoiding the use of *gazetteers* in anaphoric contexts.

This syntactic analysis associates grammatical determinants and articles to the related entities through the use of grammatical rules. Thus, a sentence is no longer just an ordered set of words and terms usually discarded for lack of atomic meaning are used to connect/highlight meaningful terms semantically rich.

Assuming this approach, each sentence is submitted to the parser which returns a set of grammatical dependencies (grouped words) pattern-matched with elements of the lexicon. Though, uppercase words sharing lexicon terms in the same dependence have more trust associated. Although valid and probably more effective, this approach suggests lower coverage (sentence without the original word order) and higher accuracy (dependency between words).

In case an Internet connection is required, an online/Web dictionary could be used to filter/exclude uppercase words typically used to refer people or institutions in a similar approach of using *gazetteers*.

Given the inference errors shown in Figure 2, the approach SQL/DBMS allows to easily integrate this system into a Web interface for manual correction/validation of information in a collaborative approach similar to that practiced in systems [2] suggesting community involvement in simple tasks for efficient development of complex systems.

References

[1] Alias-i. Lingpipe 4.0.1. `http://alias-i.com/lingpipe`, last viewed October 2010.

[2] Microsoft Language Development Center. Your speech. `http://pt.yourspeech.net/#Intro`, last viewed October 2010.

[3] ProLNat Universidade de Santiago de Compostela. Deppattern. `http://gramatica.usc.es/pln/tools/deppattern.html`, last viewed October 2010.

[4] Eduardo Ferreira, Jo ao Balsa, and António Branco. A.: Combining rule-based and statistical methods for named entity recognition in portuguese. in: Actas da 5 a workshop em tecnologias da informação e da linguagem humana, 2007.

[5] Ruituraj Gandhi and David C. Wilson. A multi-strategy approach to geo-entity recognition. In *Advances in Information and Intelligent Systems*, pages 189–200. 2009.

[6] Niyu Ge, John Hale, and Eugene Charniak. A statistical approach to anaphora resolution. In *In Proceedings of the Sixth Workshop on Very Large Corpora*, pages 161–170, 1998.

[7] Christopher B. Jones, Alia I. Abdelmoty, and Gaihua Fu. *Maintaining Ontologies for Geographical Information Retrieval on the Web*, pages 934–951. Springer Berlin / Heidelberg, 2003.

[8] Carsten Keßler, Martin Raubal, and Christoph Wosniok. Semantic rules for context-aware geographical information retrieval. In Payam Barnaghi, Klaus Moessner, Mirko Presser, and Stefan Meissner, editors, *4th European Conference on Smart Sensing and Context, EuroSSC 2009*, volume 5741 of *Lecture Notes in Computer Science*, pages 77–92, Berlin, 2009. Springer.

[9] Linguateca. Harem: Reconhecimento de entidades mencionadas em português. `http://www.linguateca.pt/HAREM/`, last viewed October 2010.

[10] A. Markowetz, T. Brinkhoff, and B. Seeger. Geographic information retrieval. In *3rd International Workshop on Web Dynamics*, 2004.

[11] Andrei Mikheev, Marc Moens, and Claire Grover. Named entity recognition without gazetteers, 1999.

[12] Nadeau, David, Sekine, and Satoshi. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30 (1):3–26, January 2007.

[13] Hien T. Nguyen and Tru H. Cao. Named entity disambiguation: A hybrid statistical and rule-based incremental approach. In *ASWC*, pages 420–433, 2008.

[14] Notepad++. How to use regular expressions in notepad++. `http://sourceforge.net/apps/mediawiki/notepad-plus/index.php?title=Regular_Expressions`, last viewed October 2010.

[15] University of Edinburgh / University of Essex. geoxwalk. `http://hds.essex.ac.uk/geo-X-walk/index.htm`, last viewed October 2010.

[16] National Institute of Standards and Technology. Muc-7: Message understanding conference proceedings. `http://www-nlpir.nist.gov/related_projects/muc/proceedings/muc_7_toc.html#named`, last viewed October 2010.

[17] David D. Palmer and David S. Day. A statistical profile of the named entity task. In *PROC. ACL CONFERENCE FOR APPLIED NATURAL LANGUAGE PROCESSING*, pages 190–193, 1997.

[18] Diana Santos and Nuno Cardoso, editors. *Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área*. Portugal, 2007.

[19] Nuno Seco. MUC vs HAREM: A contrastive perspective. In Santos and Cardoso [18], pages 35–41.

[20] Ziqi Zhang and Jose Iria. A novel approach to automatic gazetteer generation using wikipedia. In *People's Web '09: Proceedings of the 2009 Workshop on The People's Web Meets NLP*, pages 1–9, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

NeMODe - Detecting Network Intrusion with Constraints

Pedro Salgueiro
Universidade de Évora
and CENTRIA FCT/UNL, Portugal
pds@di.uevora.pt

Salvador Abreu
Universidade de Évora
and CENTRIA FCT/UNL, Portugal
spa@di.uevora.pt

Abstract

In this work we present NeMODe a declarative system for Computer Network Intrusion detection providing a declarative Domain Specific Language for describing computer network intrusion signatures that can spread across several network packets, which allows to state constraints over network packets, describing relations between several packets. NeMODe provides several back-end detection mechanisms relying on Constraint Programming (CP) methodologies to find those intrusions.

Keywords: Constraint Programming, Intrusion Detection Systems, Domain Specific Languages

1 Introduction

Network Intrusion Detection Systems are one of the most important tools in computer network management to maintain the security, integrity and quality of computer networks and keep the users data safe. To maintain the quality and integrity of the services provided by a computer network, some aspects must be verified in order to maintain the security of the users data. The description of those conditions, together with a verification that they are met can be seen as an Intrusion Detection task. These conditions, specified in terms of properties of parts of the (observed) network traffic, will amount to a specification of a desired or an unwanted state of the network, such as that brought about by a system intrusion or another form of malicious access.

Those conditions can naturally be described using a declarative programming approach, such as Constraint Programming [1] or Constraint Based Local Search Programming (CBLS) [2], enabling the description of these situations in a declarative and expressive way. To help the description of those network situations, we created NeMODe, a Domain Specific Language (DSL) [3], which enables an easy description of intrusion signatures that spread across several network packets, which will then translate the *program* into constraints that will be solved by more than one constraint solving techniques, including Constraint Based Local Search and Propagation-based systems such as Gecode [4]. It will also have the capabilities of running several solvers in parallel, in order to benefit from the earliest possible solution.

Throughout this paper, we mention technical terms pertaining to TCP/IP and UDP/IP network packets, such as *packet flags, ACK, SYN, RST, PSH, URG, acknowledgment, source port, destination port, source address, destination address, payload*, which are described in [5].

2 State of the art

2.1 Intrusion Detection Systems

Intrusion Detection Systems(IDS) play an important role in computer network security, which focus on traffic monitoring, trying to inspect traffic to look for anomalies or undesirable communications. There are two major methods to perform intrusion detection; 1) based on the

network intrusion signatures, and 2) based on the detection of anomalies on the network [6]. In this work, we adopted an approach based on signatures.

Snort is a widely used and efficient Intrusion Detection System [7], but only provides very basic mechanisms to write signature rules that spread across several network packets and in a very limited way.

Most of the recent work in intrusion detection systems has been focused on the performance [8].

2.2 Constraint Programming

Constraint Programming (CP) is a declarative programming paradigm which consists in the formulation of a solution to a problem as a *Constraint Satisfaction Problem* (CSP) [1], in which a number of variables are introduced, with well-specified domains and which describe the state of the system. A set of relations, called *constraints*, is then imposed on the variables which make up the problem. These constraints are understood to have to hold true for a particular set of bindings for the variables, resulting in a *solution* to the CSP.

Constraint Based Local Search

Constraint Based Local Search (CBLS) [2] is a fundamental approach to solve combinatorial problems such as Constraint Satisfaction Problems. CBLS is a method that can solve very large problems but its not a complete algorithm, and is unable to provide a complete or optimal solution. Usually, this approach starts with an initial, tentative solution to the problem, which is iteratively improved though minor modifications until a termination criterion is satisfied.

Adaptive Search (AS) [9] is a CBLS [2] algorithm and uses variable-based information to design general heuristics which help solve the problem. AS was recently been ported to Cell/BE, presented in [10].

Gecode

Gecode [11] is a constraint solver library based on propagation [1], implemented in C++. With Propagation-Based constraint solving, the problem is described by stating constraints over each variable that composes the problem, which states what values are allowed to be assigned to each variable, then, the constraint solver propagates all constraints and reduce the domain of each variables in order to satisfy all the constraints and instantiate the variables with valid results, thus reaching a solution to the initial problem.

3 Intrusion Detection with Constraints

Our approach to intrusion detection relies on describing the desired signatures through the use of constraints and then identify a set of packets that match the target network situation in the network traffic window, which is a log of the network traffic in a given time interval.

The network intrusion needs to be modeled as a Constraint Satisfaction Problem (CSP) in order to use the constraint programming mechanisms. A CSP which models a network situation is composed by a set of variables, V, which represents the network packets involved necessary to describe the network situation; the domain of the network packet variables, D, and a set of constraints, C which relates the variables in order to describe the network situation. We call

Listing 1 Representation of a network CSP

$$P = \{(P_{1,1}, \ldots, P_{1,z}), \ldots, (P_{n,1}, \ldots, P_{n,z})\}$$
$$D = \{(V_{1,1}, \ldots, V_{1,z}), \ldots, (V_{x,1}, \ldots, V_{x,z})\}$$
$$Data = \{Data_1, \ldots, Data_x\}$$
$$\forall P_i \in P \Rightarrow P_i \in D$$

such a CSP a network CSP. On a network CSP, each network packet variable is a tuple of integer variables, 19 variables for TCP/IP [1] packets and 12 variables for UDP packets [2], which represent the significant fields of a network packet necessary to model the intrusion signatures used in our experiments. For both TCP and UDP network packets, the individual variables of the tuples represent the time-stamp, the source/destination addresses, the source/destination ports and the packet number, used to match the packet with its data. The TCP packets have more significant fields than UDP packets, so these have some more variables, which represents the extra TCP flags and the packet sequence numbers. This number of fields may increase over time with the evolution of the work and the use of more complex intrusions.

The domain of the network packet variables, D, are the values actually seen on the network traffic window, which is a set of tuples of 19 integer values (for the TCP variables) and 12 integer values (for the UDP variables), each tuple representing a network packet actually observed on the traffic window and each integer value represents each field relevant to intrusion detection. The packets payload is stored separately in an array containing the payload of all packets seen on the traffic window. The correspondence between the packet and its payload is achieved by matching the packet number, i, which is the first variable in the tuple representing the packets and the i^{th} position of the array containing the payloads.

Listing 1 shows a representation of such CSP, where P represents the set of network packet variables, where P_n, z, is each of the individual integer variables of the network packet, in a total of z fields for each network of the n packets, with $z = 19$ for TCP packets and $z = 12$ for UDP packets. D is the network traffic window, where $D_i = (V_{i,1}, \ldots, V_{i,z}) \in D$ is one of the real network packets on the network traffic window, which is part of the domain of the packets P. $Data$ is the payloads of the network packets in present in the network window, where $Data_i$ is the payload of the packet $P_i = (V_{i,1}, \ldots, V_{i,z}) \in D$. The associated domains of the network packet variables is represented by $\forall P_i \in P \Rightarrow P_i \in D$, forcing all packets belonging to P obtain values from the set of packets in the network window D.

A solution to a network CSP, if it exists, is an assignment of network packet values, $D_i = (V_{i,1}, \ldots, V_{i,z}) \in D$, to each packet, $P_i = (P_{j,1}, \ldots, P_{j,z}) \in P$, that models the desired situation, thus identifying the network packets that identify the intrusion being detected.

4 NeMODe - A DSL to describe network signatures

We created a declarative, intuitive domain-specific programming language [3] for NeMODe, which talks about network entities, their properties and relations between them, allowing to describe network intrusion signatures, and, with base on those descriptions, generate Intrusion

[1] Here, we are only considering the "interesting" fields in TCP/IP packets, from an IDS point-of-view.
[2] Here, we are only considering the "interesting" fields in UDP packets, from an IDS point-of-view.

Detection mechanisms. A complete description of this DSL as well as other examples is presented in [12].

The key characteristic of NeMODe is to ease the way how network attack signatures are described using constraint programming, hiding from the user all the constraint programing aspects and complexity of modeling network signatures in a Constraint Satisfaction Problem(CSP), but still using the methodologies of CP to describe the problem at a much higher level, allowing to describe how the network entities should relate among each other and what properties they should verify.

LANG is a front-end to several back-ends, one to each intrusion detection mechanism, allowing to generate several detection mechanisms from a single description. Having a single specification to several constraint solvers allows the search of a solution using different methods of search, allowing to run each of those methods in parallel, allowing to choose the first result to be produced. In the current implementation, the NeMODe provides two back-end mechanisms; (1) based on the Gecode constraint solver and (2) based on the Adaptive Search algorithm.

4.1 Examples

In this work we present as an example of the DSL of NeMODe the description of a DNS Spoofing attempt. Other examples are described in [13, 14, 12].

DNS Spoofing is a Man in The Middle (MITM) attack. In this attack, the attacker tries to provide a false DNS query posted by the victim. In order to arrange this attack, the attacker tries to respond with a false DNS query faster than the legit DNS server, providing a false IP address to the name that the victim was looking for. This kind of attacks is possible to detect by looking for several replies to the same DNS query. Listing 2 shows how this attack can be programmed using NeMODe. Line 2 describes the packet that makes the DNS request. Lines 4-5, describes a first reply to the DNS request and lines 7-8 describes the second reply. Lines 10-12 states that packets `B` and `C` should be different and that the *DNS id* of both replies should be the equal to the one of the DNS request, which are the first two bytes of the packet payload.

Listing 2 A DNS Spoofing attack programmed in NeMODe

```
1  dns_spoofing {
2      udp_packet(A), dst_port(A) == 53
3
4      udp_packet(B), src_port(B) == 53,
5      dst(B) == src(A), dst_port(B) == src_port(A),
6
7      udp_packet(C), src_port(C) == 53,
8      dst(C) == src(A), dst_port(C) == src_port(A),
9
10     B != C,
11     data(B,0,2) == data(A,0,2),
12     data(C,0,2) == data(A,0,2)
13 } => {
14     alert('DNS Spoofing attempt')
15 };
```

5 Experimental Results

While developing this work, several experiments were done. We have tested several network situations, described in [13, 14, 12], as well as the DNS Spoofing presented in Sect. 4.1. All these

Table 1: Average time(in seconds) necessary to detect the intrusions using Gecode and Adaptive Search

Intrusion to detect	**Gecode (seconds)**	**A.S (seconds)**
DNS Spoofing	0.0069	0.3512
DHCP Spoofing	0.0082	0.3924
SYN flood	0.0566	0.0466

network intrusions were successfully described using NeMODe and valid Gecode and Adaptive Search code were produced for all network signatures. The code generated by NeMODe was then executed in order to validate the code and ensure that it could indeed find the desired network intrusions.

The code generated for Gecode was run on a dedicated computer, an HP Proliant DL380 G4 with two Intel(R) Xeon(TM) CPU 3.40GHz and with 4 GB of memory, running Debian GNU/Linux 4.0 with Linux kernel version 2.6.18-5. As for the Adaptive Search code, it run on an IBM BladeCenter H equipped with QS21 dual-Cell/BE blades, each with two 3.2 GHz processors, 2GB of RAM, running RHEL Server release 5.2.

In all the experiments we used log files representing network traffic which contains the desired signatures to be detected. These log files were created with the help of `tcpdump`, which is a packet sniffer, during an actual attack to a computer, which was induced to simulate the real attacks described in this work.

In the DNS spoofing attack, we used tcpdump to capture a log file, composed of 400 network packets, while a computer was under an actual attack. The attack was described using NeMODe, which successfully generated code for Adaptive Search as well as for Gecode and successfully detected the intrusions.

Table 1 presents the time(user time, in seconds) required the DNS Spoofing attempt as well as for other network situations presented in [13, 14, 12]. The execution times presented in Table 1 are the average times of 128 runs.

6 Evaluation

The performance of the prototypes shows a multitude of performance numbers relative to the intrusion detection mechanisms used for each network signature. Looking closely at the results in Table 1, it is possible to see that Gecode usually performs better than Adaptive Search, except in the SYN flood attack. This difference is explained by the fact that Adaptive Search needs a very good heuristic functions to improve its performance. The SYN flood attack performed better in Adaptive Search due to the fact that the network packets of the attack are close together and there aren't almost any other packets between the packets of the attack.

Even without a perfect heuristic of Adaptive Search, the results obtained are quite encouraging. As for Gecode, the results obtained are quite good.

As for NeMODe, it revealed be a success, allowing to to easily describe the network intrusions and generate valid code that could detect the desired network situation.

7 Conclusions and Future Work

The work presented in this paper presents NeMODe, a system for Network Intrusion detection, which provide a declarative Domain Specific Language that generates intrusion detection recog-

nizers based on Constraint Programming, more specifically, using Gecode and Adaptive Search. NeMODe presents a very expressive DSL that allows to describe network intrusion signatures by expressing relations between network packets simply by stating constraints over network packets.

This work shows that it is possible to use a single signature description based on CP to generate several recognizers, each one based on a different CP paradigms, and with that recognizers detect the desired intrusions. We proved that we can easily describe network signature attacks that spread across several network packets, which is somewhat tricky or even impossible to make using systems like Snort. Although the intrusions mentioned in this work can be detected with other intrusion detection systems, they are modeled/described with out relating the several network packets of the intrusion, much of the times using a single network packet to describe the intrusion, which could in some situations produce a large number of false positives.

Although the DSL allows to describe a broad range of attacks, it still needs more flexibility to cope with more types of signatures and include more back-ends.

There is plenty of room for improvement: we have reached an efficiency level that may be suitable to start performing network intrusion tasks on live network traffic link, allowing to apply this method in a real network to assess its performance.

References

[1] F. Rossi, P. Van Beek, and T. Walsh. *Handbook of constraint programming*. Elsevier Science, 2006.

[2] P. Van Hentenryck and L. Michel. *Constraint-based local search*. MIT Press, 2005.

[3] A. Van Deursen and J. Visser. Domain-specific languages: An annotated bibliography. *ACM Sigplan Notices*, 35(6):26–36, 2000.

[4] Gecode Team. Gecode: Generic constraint development environment, 2008. Available from http://www.gecode.org.

[5] Douglas Comer. *Internetworking With TCP/IP Volume 1: Principles Protocols, and Architecture, 5th edition*. Prentice Hall, 2006.

[6] Y. Zhang and W. Lee. Intrusion detection in wireless ad-hoc networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, page 283. ACM, 2000.

[7] H. Song and J.W. Lockwood. Efficient packet classification for network intrusion detection using FPGA. In *Proceedings of the 2005 ACM/SIGDA 13th international symposium on Field-programmable gate arrays*, pages 238–245. ACM New York, NY, USA, 2005.

[8] K.S.P. Arun. Flow-aware cross packet inspection using bloom filters for high speed data-path content matching. In *Advance Computing Conference, 2009. IACC 2009. IEEE International*, pages 1230 –1234, 6-7 2009.

[9] P. Codognet and D. Diaz. Yet another local search method for constraint solving. *Lecture Notes in Computer Science*, 2264:73–90, 2001.

[10] Salvador Abreu, Daniel Diaz, and Philippe Codognet. Parallel local search for solving constraint problems on the cell broadband engine (preliminary results). *CoRR*, abs/0910.1264, 2009.

[11] C. Schulte and P.J. Stuckey. Speeding up constraint propagation. *Lecture Notes in Computer Science*, 3258:619–633, 2004.

[12] Pedro Salgueiro, Daniel Diaz, Isabel Brito, and Salvador Abreu. Using Constraints for Intrusion Detection : the NeMODe System. In *PADL'11 - Thirteenth International Symposium on Practical Aspects of Declarative Languages*, 2011.

[13] Pedro Salgueiro and Salvador Abreu. A DSL for Intrusion Detection based on Constraint Programming. In *SIN 2010: Proceedings of the 3rd International Conference on Security of Information and Networks*, New York, NY, USA, 2010. ACM.

[14] Pedro Salgueiro and Salvador Abreu. On using Constraints for Network Intrusion Detection. In *INForum 2010 - Simpósio de Informática*, Braga, Portugal, 2010.

Utilização de um PLC para o desenvolvimento de um sistema de rega adaptativo

Shakib Shahidian, Ricardo Serralheiro, João Ramalho, João Serrano, Rui Machado

ICAAM, Herdade da Mitra, Universidade de Évora

Resumo

Foi desenvolvido um sistema de rega adaptativo com base num Controlador Lógico Programável (PLC) da Ibercomp. Este PLC é construído a volta de um processador 89c51RD2 de 8 bits, e a programação foi realizada utilizando o compilador de BASIC Bascom. Foram desenvolvidas diversas rotinas de cálculo da evapotranspiração e de gestão da rega tendo em atenção as capacidades reduzidas do processador. Foi desenvolvido um circuito para o sensor de temperatura baseado num IC LM35DZ e um amplificador operacional. O programa faz a leitura horária da temperatura e calcula a evapotranspiração com base na equação de Hargreaves Samani. O sistema realiza a rega na hora estabelecida pelo utilizador, regando sequencialmente todos os sectores do campo. O protótipo foi instalado num campo experimental, tendo-se obtido resultados muito encorajadores em termos de poupança de água. O custo do sistema é pouco superior ao de um programador de rega convencional.

Introdução

Nos últimos anos tem se assistido a um grande desenvolvimento tecnológico na área de Controladores Lógicos Programáveis, também conhecidos por PLCs. Inicialmente caros, esses aparelhos têm vindo a ficar cada vez mais acessíveis. Plataformas baseadas na família 8051, a AVR da Atmel, PIC e MAX são alguns exemplos de ferramentas que abrem novos horizontes para o desenvolvimento de sistemas de automação e controlo inteligentes de baixo custo.

A rega é o maior consumidor mundial de água e, se não for realizada com o devido cuidado pode originar erosão do solo e contaminação dos lençóis freáticos. Por outro lado, a água é um recurso escasso cujo consumo necessita de ser racionalizado, especialmente na agricultura.

A escassez do recurso água tem despertado uma grande preocupação com o seu uso racional na rega, quer de jardins e espaços verdes, quer na agricultura. Nos últimos anos tem-se assistido ao desenvolvimento de controladores de rega ditos inteligentes ou adaptativos que ajustam a dotação da rega às necessidades reais das plantas.

Existem três abordagens distintas para a criação de controladores inteligentes: os baseados no teor de água no solo (Campbell e Campbell, 1982), os baseados no stress hídrico das plantas (Jones, 2004) e os baseados no cálculo da Evapotranspiração (Shahidian et al, 2009). A última filosofia tem originado o desenvolvimento de diversos sistemas de rega com cálculo da Evapotranspiração. Dentro desta filosofia, existem basicamente duas abordagens: a centralizada, que consiste no cálculo centralizado e diário da Evapotranspiração e a sua posterior transmissão aos controladores locais através de internet ou sistemas wireless (Goodman, 2010), e a abordagem descentralizada que consiste no cálculo local da evapotranspiração pelo próprio controlador. O presente trabalho se insere nesta categoria.

Para a abordagem descentralizada há todo o interesse em medir um número reduzido de parâmetros climáticos, dado o elevado custo dos sensores. A equação de Hargreaves-Samani permite calcular a Evapotranspiração com base na medição regular da temperatura do ar, e

coeficientes dependentes da latitude do local e época do ano (Hargreaves e Samani, 1982; 1985). A equação de Hargreaves-Samani pode ser expressa da seguinte forma:

$$ET_{HS} = \alpha(T + 17.78)(T_{\max} - T_{\min})^{0.5} R_a$$

Onde, T é a temperatura média, T_{max} é a temperatura máxima diária, ºC, T_{min} é a temperatura mínima diária, ºC, R_a é a radiação extraterrestre (MJ m^{-2} dia^{-1}), e α é um constante de calibração que varia conforme o local, e que para as condições de Évora pode ser considerado como 0.0023 (Teixeira et al. 2008). A radiação extraterrestre pode ser calculado com base na época do ano e latitude, ou retirado de tabelas (Allen et al. 1998)

O objectivo deste trabalho foi desenvolver um sistema de rega adaptativo baseado num PLC comercial de baixo custo, que calcule diariamente as necessidades hídricas das plantas utilizando a equação de Hargreaves-Samani e realiza as regas de forma autónoma. O trabalho envolve o desenvolvimento do equipamento necessário e a programação do PLC.

Material e Métodos

Foram avaliadas diversas plataformas possíveis, tendo-se inicialmente utilizado uma placa Industrologic IC51. No entanto, as limitações dessa placa levaram à utilização do sistema uPLC IV da Ibercomp, construído a volta de um processador Atmel 89c51RD2+, com LCD e teclado integrados. Adicionalmente o sistema possui 8 relés para o controlo das electroválvulas de rega, um conversor analógico/digital TLC549 de 8 bits para a medição da temperatura, relógio em tempo real, 8 entradas opto-isoladas, 64K de memória para o programa e um bus I^2C. Possui ainda 8K de EEPROM no bus I^2C, o que é muito útil para armazenar as opções introduzidas pelo utente, como por exemplo a hora de rega. O sistema pode funcionar a 220 V, ou a 12V, o que o torna ideal para instalação em locais remotos e alimentados por energia solar. O custo unitário à inferior a 140 Euros, o que o coloca ao nível de um controlador de rega de gama média.

A Câmara Municipal de Évora disponibilizou um espaço verde próximo do Clube de Ténis de Évora para a realização dos ensaios.

Desenvolvimento do sistema

Um dos primeiros passos foi o desenvolvimento do sensor de temperatura. Inicialmente utilizou-se um termister, o PT100, devido à sua grande precisão e disponibilidade. Eventualmente optou-se por um circuito integrado LM35DZ que é um termómetro de precisão calibrado directamente em graus Celsius, com uma precisão de 0.5ºC. Como o ML35DZ tem um ganho de apenas 10mV por ºC, foi necessário construir um circuito amplificador. Para o efeito foi utilizado um amplificador operacional LM358N para se conseguir uma amplificação de 10 vezes. Foi construído um circuito impresso para o sensor, por forma a se conseguir um sensor fiável e compacto (Figura 1). O sensor fica alojado numa pequena caixa com ranhuras, para a circulação do ar e é colocada à sombra, por forma a se consegui um equilíbrio térmico com o ar.

No primeiro ano a programação foi feita em Tiny Machine Basic, no entanto esse compilador é pouco flexível em termos de cálculos, apenas possuindo funções para trabalhar com números de 8 bits. Deste modo, numa segunda fase a programação foi feita utilizando o Bascom (MCS electronics), um compilador BASIC para a família 8051, com programação estruturada.

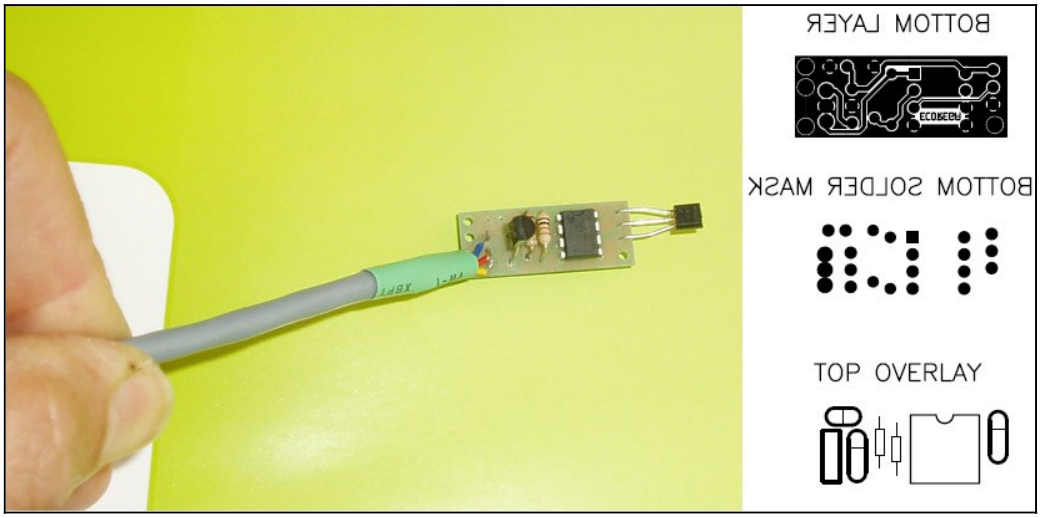


Figura 1. O circuito impresso e o sensor de temperatura.

Mesmo utilizando o Bascom, existem dificuldades em implementar este tipo de programas, nomeadamente o número limitado de variáveis e o facto do processador ser de 8 bits, o que exige algum cuidado na manipulação de números de 16 bits, ou caracteres alfanuméricos. Por outro lado o compilador não possui certas funções como a raiz quadrada, que teve de ser implementada através de uma tabela incluída no programa. A estrutura do programa está apresentada na Figura 2.

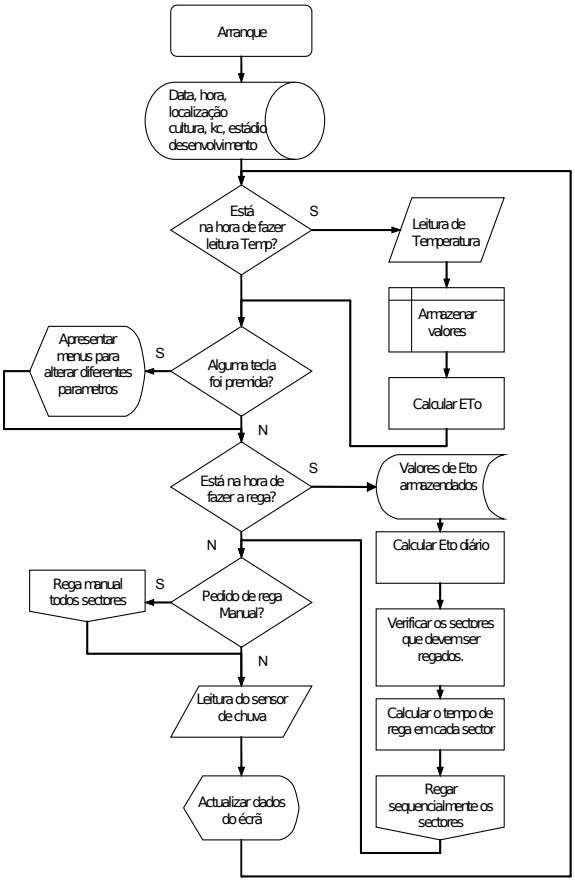


Figura 2. Fluxograma do programa de gestão da rega.

O controlador faz uma leitura horária da temperatura, e calcula a temperatura máxima, mínima e média das últimas 24 horas. Através dos menus existentes, o utilizador define a hora em que a rega deve ser efectuada (Figura 3) e a cultura. Quando chega a hora da rega, o controlador calcula a Evapotranspiração diária e calibrada para o local. Os valores mensais de radiação estão armazenados no programa em forma de DATA. Em seguida o controlador armazena em memória a hora de abertura de cada um dos sectores de rega, por forma a que os sectores sejam regados sequencialmente. As electroválvulas são então abertas e fechadas de acordo com os tempos armazenados em memória.

Figura 3. Aspecto geral do controlador de rega, indicando a hora da rega e a dotação calculada pelo método de Hargreaves-Samani.

Os variáveis introduzidos pelo operador, como por exemplo a hora da rega, o débito do sistema de rega e os sectores regados não devem ficar na memória volátil. Assim, esses dados são armazenados na memória EEPROM no bus I^2C. Da mesma forma, a Evapotranspiração, e a hora de início da rega ficam também guardados na EEPROM, para que no caso de uma falha de energia o sistema possa recuperar e funcionar sem interrupção.

Por forma a poder avaliar o desempenho do sistema, todas as leituras de temperatura e os resultados dos cálculos são também armazenados no EEPROM e acessíveis através da porta RS232.

Resultados

O sistema foi instalado em Janeiro de 2010 num relvado municipal próximo do Clube de Ténis de Évora. O sistema foi instalado directamente na caixa de rega da CMÉvora, substituindo facilmente o programador existente. O controlador registou a temperatura horária, bem como a rega realizada diariamente. Os dados foram retirados regularmente com recurso a um portátil. As temperaturas horárias registadas para um período de 30 dias (21 de Abril-20 de Maio) estão apresentadas na Figura 4. Verifica-se que a amplitude térmica apresentou uma grande variabilidade. A temperatura máxima diária só apresentou valores inferiores a 20 ºC nos dia 8, 9, 13 e 14.

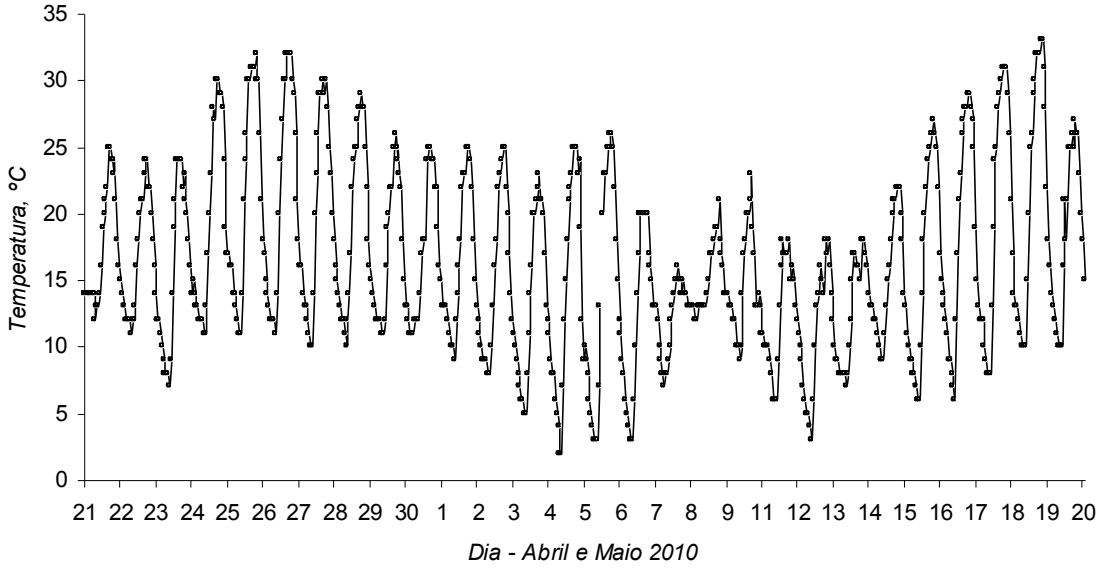


Figura 4. Temperatura horária de ar entre os dias 21 de Abril e 20 de Maio 2010

Na figura 5 apresenta-se a dotação aplicada pelo sistema adaptativo em conformidade com a evapotranspiração calculada através da equação de Hargreaves-Samani, procurando ir ao encontro das necessidades diárias da cultura. Verifica-se que as dotações diárias aplicadas variaram entre os 1.2 e 6.2 mm, tendo o sistema reduzido a aplicação de água nos dias mais frios, e aumentado a dotação no período de maior calor. Para efeitos de comparação, é também apresentada a dotação de rega que seria aplicada por um programador de rega normal, que diariamente aplica uma dotação fixada com base em valores médios históricos, como por exemplo 5 mm dia^{-1} no caso de Évora. Para o curto período estudado, a poupança de água em média foi de 0,58 mm dia^{-1}, ou seja 17 mm no período em causa, o que corresponde a uma poupança de 170 m^3 por hectare e mês.

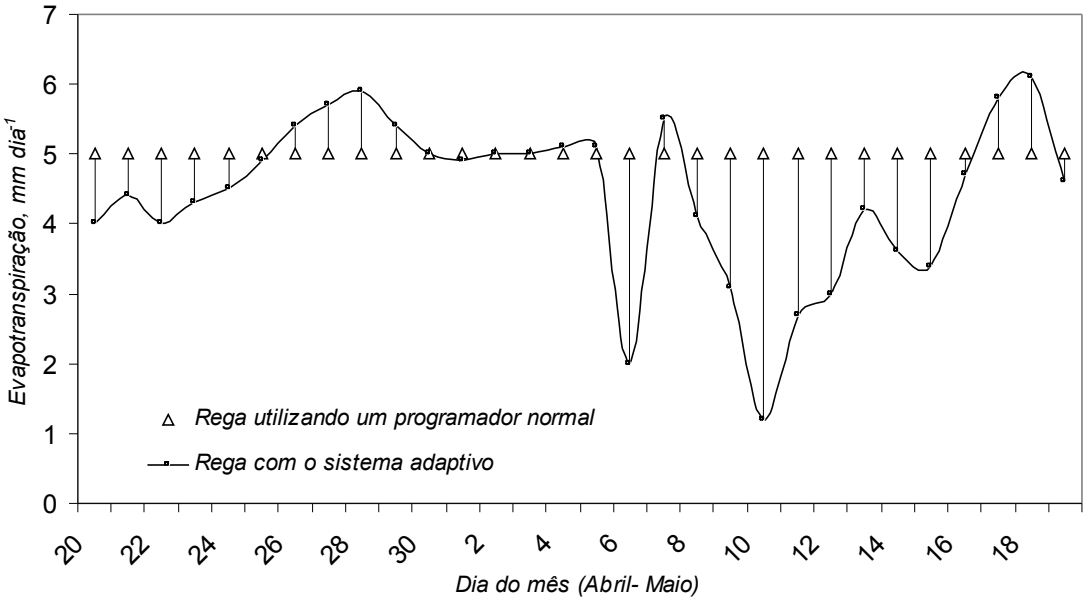


Figura 5. Dotação de rega obtida com o sistema adaptativo e comparação com um programador convencional.

Conclusões

Com base no comportamento e nos resultados obtidos pode-se concluir que cada vez mais os PLC podem trazer "inteligência" às tarefas comuns, substituindo com economia muitos sistemas de controlo convencionais.

O Controlador de rega desenvolvido neste trabalho calculou eficazmente a evapotranspiração diária e realizou as regas diárias, conseguindo alguma poupança no consumo da água. O que no futuro poderá ser melhorado através do estudo da relação entre a evapotranpiração calculada com este sistema e a real para proceder a reajustamentos nos coeficientes de calibração.

Agradecimentos

Os autores agradecem à Câmara Municipal de Évora pela disponibilidade demonstrada, especialmente ao Eng. Costa Diebe. Este trabalho foi realizado no âmbito de uma bolsa financiada pela FCT e Comunidade Europeia.

Bibliografia

Allen RG 1993 Evaluation of a temperature difference method for computing grass reference evapotranspiration. Report submitted to the Water Resources Develop. and Man. Serv., Land and Water Develop. Div., FAO, Rome. 49 p.

Campbell, GS, Campbell, MD, 1982. Irrigation Scheduling Using Soil Moisture Measurements: Theory and Practice Advances in Irrigation Volume 1. p 25-42.

Goodman, R. 2010. Wi-Fi Evapotranspiration Irrigation Controller. Electrical Engineering Department. California Polytechnic State University

Hargreaves, G.H., Samani, S. 1982. Estimating potential evapotranspiration. J. Irrig. Drain. Div., 108(3), 225–230.

Hargreaves, G.H., Samani, Z.A. 1985. Reference crop evapotranspiration from temperature. Appl. Eng. Agric. 1 (2), 96–99.

Jones, HG. 2004. Irrigation scheduling: advantages and pitfalls of plant-based methods. Journal of Experimental Botany Volume 55, Issue 407. p. 2427-2436.

Shahidian, S, Serralheiro, RP, Teixieira, JL, Santos, FL, Oliveira, MRG, Costa, JL, Toureiro, C, Haie, N., Machado, R.M. 2009. Drip irrigation using a PLC based adaptive irrigation system. WSEAS Transactions on Environment and Development. Issue 2, Volume 5.p 209-218.

Teixeira JL, Shahidian S, Rolim J. 2008. Regional analysis and calibration for the South of Portugal of a simple evapotranspiration model for use in an autonomous landscape irrigation Controller. WSEAS Transactions on Environment and Development. Issue 8, Volume 4. p.676-686.

Adaptive Interface for the Electronic School Notebook

Luis Alexandre
LabSI[2] / ESTIG,
Instituto Politecnico de Beja, Portugal
luis.alexandre@ipbeja.pt

Salvador Abreu
Universidade de Evora and
CENTRIA FCT/UNL, Portugal
spa@di.uevora.pt

Abstract

In this article we describe the construction of an adaptive interface for the Electronic School Notebook, with the ability to predict the most likely task that a user is about to perform, based on prior knowledge of actions already made in the system, combined with the action time frame. This adaptive capability will reduce the number of explicit interactions to accomplish a certain task. The system makes use of machine learning algorithms, based on decision trees and Markov chains.

1 Introduction

The Electronic School Notebook (CE-e) [1], has become a valuable tool for children with special needs, in the organization of notes in a classroom environment and also in the organization, in a logical manner, of electronic documents related to a given course.

The CE-e together with other assistive technologies, like Eugenio [10] actively support these children, allowing them to perform writing tasks within a very similar time period to the one spent by children without special needs. This tool promotes the adoption of a methodical process of collecting notes in classroom context, as a result of this process, remarkable benefits can be obtained in terms of academic success [4]. If the system becomes aware of the task at hand by the user, it will be able to better support users, by helping in accomplishing the task. In some situations it's very difficult to determine the intentions behind a user actions sequence, but in other situations using the domain knowledge [8] or observing the users behavior in similar situations [15], we are able to predict the goal of the user.

Not only chidrem with motor impairments can benefit from these technologies, also students with cognitive impairments can work in a more autonomous manner with the help of these systems. These kind of mecanisms have already been tried in other aplications, like email clients [15], in order to free the user from the execution of routine tasks by delegating the execution responsibility to the system. This approach has also been tried in the development of Eugenio's keyboard assistant [10, 19]. The project guiding principle is DWIM (Do What I Mean) [21]. This principle states that the system should behave exactly according to the user objective, instead of behaving according to a miss-executed instruction, not intended by the user. This is an unattainable goal; still it's guiding our intentions.

The main objective of this project, is to implement and evaluate an adaptive interface for the Electronic School Notebook that allows students with special needs a higher level of support in performing several tasks. To attain this objective, the system must possess application and user models [8] in order to identify the user intentions to provide the necessary assistance in the accomplishment of the task. In a first moment we gathered the required information to build a knowledge base that contains all the tasks performed by the user on the system, the different actions that lead to the task and finally the context in which they were performed. To characterize the context we gathered a timestamp composed of the date, day of the week, the current time and the associated activities. These records were gathered during normal classes,

in the student's personal computers, using CE-e. All the students have cerebral palsy, but this problem only affects them physically. None of these students uses any custom I/O devices, such as switches or adapted mice.

This data and knowledge base was used on the next phase of the project, were we built and evaluate support mechanisms that help users in the accomplishment of tasks, based on the knowledge base content. This is the main contribution of the project. If the system is able to correctly interpret all the context dimensions, it may evolve from a set of hardcoded rules that provide a limited set of adaptations, to a system that with its continued use, will be able to learn and create new adaptations, more appropriate for the user profile. This "intelligence" will only be possible with the use of machine learning algorithms.

On Section 2 of this article we will describe the sate of the art in this subject. The description of the system is made on Section 3 and finally on Section 4 we present a conclusion and propose directions for work to be done in the future.

2 State of the Art

Nowadays we are surrounded by computer systems (e.g. desktops, laptops, PDAs, among many others) and we become computer dependent, all aspects of our lives are determined by decisions or actions that took place on a computer. However these systems could provide a greater support, if they were able to provide it in an automatic and autonomous form. This support is only possible if the system is able to determine or recognize the context in which both the user and the machine are inserted [3].

Many researchers have reported the importance and benefits that context knowledge can bring to the usability of computer systems thereby increasing and enhancing the Human-Computer Interaction [5]. The main idea behind this knowledge of the context, known as context-awareness, states that computers can feel and react based on the environment. In order to enable this reaction, computer systems should have at their disposal a set of stimuli and prior knowledge that are combined with a set of hardcoded rules. The reaction is the logical result of these calculations [20].

A context-aware system can and should try to make assumptions about the surrounding environment. In [7] the context is defined as "*any information that can be used to characterize a situation of any one entity*".

Initially the context was understood has being the location, but more recently some researchers proposed that the location coulnd't be understood as the context of a situation, but rather as a component of context [7]. It was further proposed that the context could be used to build adaptive and intelligent interfaces, in which the interaction sould be as implicit as possible.

In the last years, context information has been combined with time; as result of this many researchers have proposed several types of "intelligent" interfaces (intelligent Human-computer Interaction). Examples of this new kind of interaction are the "Adaptive and Predictive Interfaces". These interfaces try to minimize the two major problems that affect the interaction between humans and computers: (i) The considerable quantity of information and widgets that populate the interfaces; and (ii) The fact that users must decide the next step or action in a very short time [17].

The idea behind these interfaces is the following: "*Instead of being the user to adapt to the system, the system adapts to the user*". Although based on a known premise, the inherent problems are in large number and complexity. An adaptive interface must be based on cognitive models and on the surrounding environment [17]. These models try to explain the user's level

of expertise and experience, through parameters such as: (i) commands made, (ii) error rates, (iii) typing speed, among others. An adaptive interface can make the adjustments in one of two forms:

- The adaption responsibility is splited between the system and the user, for example, the system makes a small adaptation and waits for its acceptance, if the adaptation is not accepted or not satisfactory, the system makes a new adaptation;
- Alternatively, the system suggests the most adequate adaptation for the current context in a totally independent mode.

However an adaptive interface also has its problems and among others we can mention the fact that the user can not create a mental model of the system, if the appearance of the system often changes. Another potential problem may result from the loss of control that a user may feel. Finally, and perhaps the biggest problem of adaptive interfaces, turns out to be the cost and complexity of the implementation. Another paradigm of adaptability and predictability derives from the attention and suggestion. In this case the interfaces are alert and automatically suggest actions to users.

An attentive interface automatically sets priorities and in accordance with these priorities presents to the user the most appropriate information or options regarding the current context and time, in order to optimize the resources of both user and system. With this optimization, the interaction actors never incur in overloading [22]. As in adaptive interfaces, attentive interfaces base their decisions on information and models, based on decisions previously made by the user.

On the other hand we have the suggestive interfaces, which only provide the user with clues about the next possible action, through the enhancement of certain interface components [12].

The interaction with a suggestive interface is very simple, in fact the user only has to choose one of the highlighted options or if none of the options matches the desired one, the user only has to choose the desired option. Usually suggestions are generated by a system often called "suggestions engine", which constantly observes the system state, generating a set of suggestions that matches the patterns closest to the current state.

3 Architecture of the Adaptive Interface

The interface of the CE-e prototype was been designed according to a design methodology for interactive systems with focus on the user and therefore, is quite simple and intuitive [1]. However as CE-e intends to provide as much help as possible in tasks of notetaking and information organization, it's intended that the interface can alleviate, as much as possible, the user of routine and repetitive tasks, which may be particularly useful for users with severe motor problems.

Thus, based on our experience, in the opinion of technicians and other experts, we decided to implement a suggestive interface on CE-e. We made this choice because these kind of interfaces are seen as a possible path for the future of interfaces, since they exploit context-awareness in order to reduce the number of explicit commands to be performed by users, required for any given operation [12] [6]. The adaptive interface, designed for the CE-e, meets all these requirements, extending the traditional notions of an interface.

3.1 The Knowledge Base

For some years, researchers have been working in Augmentative and Alternative Communication (AAC) systems. These efforts are only justified if the impact of these technological advances

cid	name	type	notnull	dflt_value	pk
0	ID	INTEGER	0		1
1	date	NVARCHAR(40)	0		0
2	weekDay	NVARCHAR(20)	0		0
3	time	INTEGER	0		0
4	form	NVARCHAR(50)	0		0
5	discipline	NVARCHAR(50)	0		0

Figure 1: CE-e log structure

can be measured. Thus, in order to facilitate the analysis of the collected data; researchers have defined a standard logging format that allows the analysis of data in a systematic way, called Universal Logging Format [14]. Under this format, we built a log system with the structure presented in Figure 1 that contains the following fields: date, day of week, time, system page, active discipline and resulting action.

But, a log mechanism it's useful if we have a fast, precise and powerful form to extract the information we want. Inspired in [11] and on features of relational DBMS we decide to use SQLite[1]. This relational database management system, besides allowing the construction of queries in standard SQL, doesn't need to have an active server in the system. This log mechanism, adds to the capabilities of SQL, a total independence of the system, since this RDMS is added to the application via a DLL built in C Programming Language.

3.2 The Encoding

For some time now, that researchers in the field of Human-computer Interaction, develop studies on the adaptability of interfaces to users, through learning systems based on machine learning, both for traditional desktop interfaces and for Web interfaces [13, 17, 9].

We can divide machine learning, through classification, in two major groups: Supervised Learning and Unsupervised Learning [2]. The collected logs for the formation of the knowledge base are constituted by the time frame, context and the performed action, i.e. the user's objective. In this scenario we face a case of supervised learning, where the user actions will be the class.

In the possession of real use records, it was possible to start working in the "intelligence" component of the system, in order to prove the feasibility of an intelligent interface for the Electronic School Notebook. This type of tests is known as "proof of concept".

As soon as possible, we wanted a prototype that demonstrates the usefulness of the Electronic School Notebook adaptive interface, so we decided to implement the inference engine of the CE-e with a Markov chains based algorithm [18].

4 Application Testing

The application tests where performed with two users. The first one, an adult without special needs, tested the system in a simulated environment, without any supervision and covering almost all possible actions. The second user, a nine years old student with special needs, tested the system in a classroom environment, with the supervision of a teacher; also, this user only

[1]Project Web site: http://www.sqlite.org/

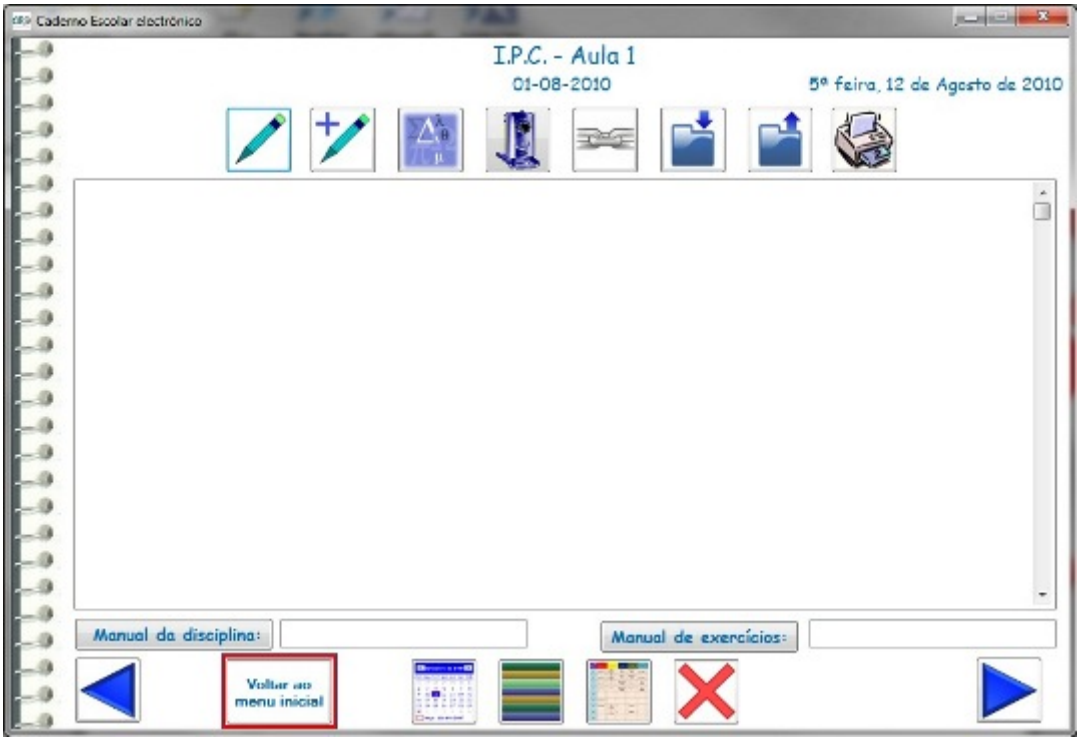


Figure 2: System prototype with the inference engine

used a limited set of tasks.

Holistically, we can say that the obtained results are satisfactory. In the case of user 1, the inference engine achieved a hit rate of 58% and a hit rate of 73% for user 2. These differences are justified, because user 1 covered almost all possible actions while user 2 only performed a limited set of actions.

The results are engaging; however these results can still be somewhat misleading, because the test period is too short and the number of users is low. Despite everything we may think, these preliminary results, are engaging and could serve as a base for profound tests with more users, covering a wider range of students across several schools and several educational levels.

5 Conclusions and Future Work

In this article we propose and present an adaptive interface for a tool that aims to be an alternative to traditional school notebooks, for students with special needs, particularly those who only have physical problems, that restricts the mobility and ability to manipulate traditional I/O mechanisms. Currently we have a fully operational prototype of the CE-e with the adaptive interface (Figure 2), that makes suggestions to the user using the knowledge base, which contains information about past actions in the system.

We want a system that continuously learns. This learning process can quickly push the number of lines, of the log file, to a few thousand. As a result, the inference engine can become very slow. A possible solution to this problem is the limitation of the log file, in number of lines or size. However this decision can only be taken after a period of use and testing with a duration not smaller than six months. These tests will determine what is the best structure for the knowledge base, ”with forgetfulness” or ”without forgetfulness”. A possible solution for the

continuous collecting of logs is an offline processing, which will lead to the construction of an index of actions. This technique is quite used in information retrieval systems [16].

The suggestion mechanism highlights the most likely option. This enhancement is combined with a keyboard shortcut, to avoid a mouse movement to confirm the option.

CE-e wants to be an effective alternative to traditional school notebooks. If the introduction of an adaptive interface proves to be an asset, students with physical and motor difficulties, will have at their disposal a tool that will help them further more in the organization of all writing tasks and in the organization of electronic documents. By now, the adaptive CE-e has an interface that can predict the most likely action for a certain context, which reinforces the lemma of the project: *Help and relieve students with special needs of routine tasks, that don't add capital gains to the work that they have to perform, in order to assist in their integration into regular education.*

Finally, we would like to thank the valuable collaboration, inputs and ideas of the faculty staff of the Polytechnic Institute of Beja assigned to the Information Systems and Interactivity Lab (LabSI^2).

References

[1] Luís Alexandre, Luís Garcia, and Luís Bruno. Development of an electronic scholar notebook for students with special needs. In *DSAI2007*, Vila Real, Portugal, 2007.

[2] Ethem Alpaydin. *Introduction to Machine Learning*. MIT Press, 2004.

[3] Mark Lawrence Blum. Real-time context recognition. Master's thesis, Department of Information Technology and Electrical Engineering, Swiss Federal Institute of Technology Zurich (ETH), 2005.

[4] Joseph R. Boyle and Mary Weishaar. The effects od strategic notetaking on the recall and comprehension of lecture information for high school students with learning disabilities. *Learning Disabilities Research & Practice, 16(3)*, pages 133–141, 2001.

[5] Nicholas A. Bradley and Mark D. Dunlop. Toward a multidisciplinary model of context to support context-aware computing. *Human Computer Interaction*, 20:403 – 446, 2005.

[6] A. Van Dam. Post-wimp user interfaces. *Communications of the ACM*, 2:63–67, 1997.

[7] Anind K. Dey. Understanding and using context. *Personal Ubiquitous Computing*, 51:4 – 7, 2001.

[8] Alan Dix, Janet Finlay, Gregory Abowd, and Russell Beale. *Human Computer Interaction, 3rd Edition*. Prentice Hall, 2004.

[9] Enrique Frias-Martinez, Sherry Y. Chen, and Xiaohui Liu. Survey of data mining approaches to user modeling for adaptive hypermedia. *IEEE Transactions on Systems, Man and Cybernetics - Part c: Applications and Reviews*, 36, NO. 6:734 – 749, 2006.

[10] Luís Garcia. Concepcão, implementacão e teste de um sistema de apoio à comunicacão aumentativa e alternativa para o português europeu. Master's thesis, Universidade Tecnica de Lisboa, Instituto Superior Tecnico, 2003.

[11] Marcos Andre Goncalves, Ganesh Panchanathan, Unnikrishnan Ravindranathan, Aaron Krowne Edward A. Fox Filip Jagodzinski, and Lillian Cassel. The xml log standard for digital libraries: analysis, evolution, and deployment. *Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, 1:312 – 314, 2003.

[12] Takeo Igarashi and John F. Hughes. A suggestive interface for 3d drawing. *Computer Science Department, Brown University.*

[13] Pat Langley. *Machine learning for adaptive user interfaces*, volume 1303/1997 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2006.

[14] Gregory W. Lesher, Bryan J. Moulton, Gerard Rinkus, and D. Jeffery Higginbotham. A universal logging format for augmentative communication. 2000.

[15] Pattie Maes. Agents that reduce work and information overload. *Communications of the ACM 37(7)*, 1994.

[16] C.D. Manning, P. Raghavan, and H. Schtze. *An Introduction to Information Retrieval*. Cambridge University Press, 2008.

[17] Anthony F. Norcio and Jaki Stanley. Adaptive human- computer interfaces: A literature survey and perspective. *IEEE Transactions on Systems, Man and Cybernetics*, 19, No.2:399 – 408, 1989.

[18] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE 77*, 2:257 – 286, 1989.

[19] Nelson Rodrigues. Desenvolvimento de mecanismo de interaccão predictiva para aumentar o desempenho de tarefas. Escola Superior de Tecnologia e Gestão do Instituto Politecnico de Beja, 2007. Projecto de Licenciatura.

[20] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'94)*, 1:89 – 101, 1994.

[21] Warren Teitelman. A tour through cedar. In *ICSE '84: Proceedings of the 7th international conference on Software engineering*, pages 181–195. IEEE Press, 1984.

[22] Roel Vertegaal. Designing attentive interfaces. *ETRA'02 New Odeans Louisiana USA*, 1:23 – 30, 2002.

WAACT - Widget Augmentative and Alternative Communication Toolkit

Gonçalo Fontes
LabSI^2 - ESTIG
Instituto Politécnico de Beja
Portugal
goncalo.fontes@ipbeja.pt

Salvador Abreu
Universidade de Évora e
CENTRIA FCT/UNL
Portugal
spa@di.uevora.pt

Resumo

Neste artigo descreve-se a proposta de implementação de uma plataforma de desenvolvimento de sistemas de Comunicação Aumentativa e Alternativa para programadores, com o objectivo de melhorar a produtividade e diminuir os tempos despendidos na implementação deste tipo de soluções. Esta proposta assenta numa estrutura composta por *widgets* configuráveis por código, integráveis em novas aplicações, numa filosofia de reaproveitamento de objectos e funcionalidades. Esta plataforma pretende ainda dar flexibilidade aos programadores, através da possibilidade de introdução de novas funcionalidades e *widgets*. A implementação em tecnologias *open source* independentes da plataforma, permitirão utilizar os objectos deste *toolkit* em vários sistemas operativos.

1 Introdução

A comunicação é uma prática e uma necessidade fundamental para todos os seres humanos. No entanto, por diversos motivos, muitos de nós sofrem de condições físicas que tornam a comunicação tradicional difícil ou até mesmo impossível.

Certas perturbações sensoriais, cognitivas ou motoras podem comprometer total ou parcialmente as capacidades comunicativas humanas. As estimativas apontam para que cerca de 10% da população mundial seja portadora de um qualquer tipo de deficiência, sendo que nesse grupo, uma percentagem bastante significativa é afectada por deficiências ao nível da comunicação [9]. Nestas circunstâncias pode-se recorrer á Comunicação Aumentativa e Alternativa (CAA).

Segundo a *American Speech-Language-Hearing Association* (ASHA), esta área de prática clínica tenta compensar de forma temporária ou permanente, incapacidades de comunicação por parte de pessoas com dificuldades ao nível da fala ou escrita [11]. Ainda segundo a ASHA um sistema de CAA é, *"um grupo integrado de componentes, incluindo símbolos, ajudas, estratégias e técnicas usadas por indivíduos para melhorar a comunicação"* [9].

Os sistemas de informação e as tecnologias actuais podem ser um importante auxílio para os sistemas de CAA, tendo sido desenvolvidos nos últimos anos esforços nesse sentido com o desenvolvimento e a implementação de diversas soluções. Algumas destas soluções de apoio a CAA têm vindo a ser desenvolvidas pelo Laboratório de Sistemas de Informação e Interactividade (LabSI^2), em conjunto com o Centro de Paralisia Cerebral de Beja (CPCB) e o Instituto de Engenharia de Sistemas e Computadores: Investigação e Desenvolvimento em Lisboa (INESC-ID), entre as quais se destaca o "Eugénio - O Génio das Palavras"[5].

Este sistema é uma ferramenta de apoio à escrita de textos em Português Europeu que recorre a modelos estatísticos baseados em n-gramas [4] para sugerir um conjunto de palavras prováveis na sequência do texto já escrito. O Eugénio funciona no ambiente MS Windows e apoia a escrita de mensagens em qualquer aplicação deste sistema operativo.

No desenvolvimento de outros sistemas de apoio à CAA, como é o caso do Caderno Escolar - Electrónico (CE-e) [1], tem-se verificado a necessidade de reutilização de componentes de software já desenvolvidos para outros sistemas. Por exemplo, no CE-e, que pretende ser uma alternativa digital

aos tradicionais cadernos de papel para alunos com necessidades especiais, verificou-se a utilidade de incorporação de algumas funcionalidades já desenvolvidas para o Eugénio, como a predição de palavras ou o teclado de ecrã. Além disso, se estas ferramentas adoptarem uma estrutura modular que facilite a substituição dos seus componentes, então poder-se-ão experimentar e avaliar de forma mais eficaz novas abordagens a determinadas técnicas de CAA, como é o caso da predição de palavras.

Contudo, a integração de componentes entre as duas aplicações anteriormente descritas não é tarefa fácil, visto que tais sistemas não foram desenvolvidos de forma modular e nem sequer na mesma tecnologia. Neste contexto, este artigo propõe uma plataforma de desenvolvimento transversal ao sistema operativo que permita a reutilização de componentes de software, numa linguagem multiparadigma e de uso geral.

De forma a introduzir este trabalho, na secção 2 apresentamos uma revisão do estado da arte, sendo na secção 3 apresentado e descrito o sistema. Na secção 4 será explicada a avaliação e o trabalho relacionado, e por fim, na secção 5, concluímos e propomos direcções para trabalho futuro.

2 Estado da Arte

É comum, no que diz respeito ás tecnologias de informação, que no desenvolvimento de um software ou sistema, o público portador de um qualquer tipo de deficiência cognitiva ou motora, acabe por não ser contemplado pelos requisitos das várias fases do projecto. Duas das razões para esta situação prendem-se com a especificidade que os sistemas devem ter neste tipo de casos e a variedade de situações entre cada utilizador e o seu estado.

No entanto, o desenvolvimento de software de apoio à CAA é imprescindível para as pessoas que conseguiram ganhar parcial ou total autonomia graças às tecnologias, bem como para futuros utilizadores.

2.1 Comunicação Aumentativa e Alternativa

Um dos principais objectivos desta área, é fornecer a ajuda necessária às pessoas incapazes de satisfazer as suas necessidades diárias de comunicação através de meios convencionais.

Ao contrário do que normalmente é sugerido, a Comunicação Aumentativa e Alternativa abrange todas as formas de comunicação, não sendo esta apenas um exclusivo da forma oral, sendo assim usada para a expressão de pensamentos, necessidades, vontades ou ideias. Todos usamos CAA, mesmo que inconscientemente, quando fazemos expressões faciais ou gestos, usamos símbolos ou imagens, ou ainda quando escrevemos. Todas estas utilizações de CAA permitem-nos usufruir de uma prática fundamental, a comunicação.

As pessoas com graves dificuldades ao nível da fala ou expressão têm nestes métodos de comunicação, uma forma de complementar o seu discurso existente, ou, se este for inexistente, uma forma de o substituir. Para se exprimir, estas pessoas dispõem de diversas ajudas aumentativas, como os quadros de comunicação por imagens ou símbolos, ou ainda equipamentos electrónicos.

Contudo, se estas ajudas permitem uma melhoria significativa da comunicação dos seus utilizadores, estas também apresentam alguns problemas, tais como o facto de a sua utilização ser bastante lenta comparativamente com os métodos de expressão mais utilizados, a fala ou escrita. Este é um dos principais problemas ligados ao desenvolvimento de soluções de auxílio a comunicação, sendo que os dois valores mais importantes expressados por pessoas que utilizam este tipo de sistemas são: (i) Dizer exactamente o que querem dizer; (ii) Dizê-lo o mais depressa que consigam [6].

2.2 Software de apoio a CAA

Como já foi referido, ao longo dos últimos anos têm sido desenvolvidos muitos esforços na construção de soluções tecnológicas com o intuito de auxiliar os utilizadores com dificuldades comunicativas. A grande maioria dos *softwares* de CAA desenvolvidos tem por base, de uma forma ou de outra, um teclado de ecrã[1], tanto na sua forma tradicional funcionando nos tradicionais computadores, como em interfaces físicas adaptadas (dispositivos do tipo *"handheld"*). Este *softwares* são utilizados para a obtenção de frases, podendo estas ser formadas por letras, palavras ou por sequências de símbolos pictóricos. Um exemplo deste tipo de sistemas é o Eugénio [5]. Esta ferramenta, que dispõe de uma funcionalidade de predição, recorre a modelos estatísticos baseados em n-gramas [4] para sugerir um conjunto de palavras prováveis na sequência do texto.

A ideia por detrás das n-gramas ou modelo de *Markov* é prever a probabilidade da letra seguinte num determinado texto. Para isso são examinadas amostras de textos denominadas de textos de treino que, com base numa estimativa de parecenças, irão identificar a letra com maior probabilidade de ser a seguinte.

Para além da predição de palavras, o Eugénio dispõe ainda de outros mecanismos de apoio a CAA tais como, o seu sistema de varrimento da aplicação, que permite aos utilizadores portadores de qualquer deficiência motora, utilizar mecanismos de I/O alternativos aos mecanismos tradicionais ou, o seu mecanismo de extensão de abreviaturas, que permite ao seu utilizador a escrita de palavras ou frases utilizando a sua forma abreviada previamente definida.

Isolando todas estas funcionalidades, podem-se extrair componentes dos mais diversos tipos, como a predição de palavras, o teclado de ecrã, o varrimento e a expansão de abreviaturas. Todos eles seriam de uma grande utilidade em sistemas similares.

Para além do Eugénio, existem algumas outras ferramentas de CAA que permitem a expressão de palavras ou frases, quer de uma forma escrita, quer de uma forma falada, por intermédio de sintetizadores de voz. Um desses casos é o Talkactive [12] que recorre a modelos de *High Frequency Vocabulary* ou *Core Vocabulary*: Este tipo de modelo define que um número relativamente pequeno de palavras pode constituir a grande maioria do que é dito em comunicações normais. Com algumas centenas de palavras uma pessoa pode dizer cerca de 80% de tudo o que é necessário em comunicações diárias [13].

Este é um sistema de utilização simples que recorre a símbolos pictóricos para identificar palavras, que conjugadas entre si, permitem formar as frases que o utilizador pretende comunicar por intermédio de imagens representativas, tendo assim a vantagem de num único *click* poder transmitir uma palavra, frase ou ideia.

Devido em grande parte a sua simplicidade, estes sistemas de símbolos ganharam grande popularidade em vários países, sendo usados com grande sucesso por pessoas portadoras de paralisia cerebral [3][10]. Contudo, o Talkactive também possui claras desvantagens, como o facto de, devido a sua extensão, não ser possível disponibilizar num único ecrã todo o vocabulário de utilização diária ou de não estar disponível em português.

No entanto, à semelhança do que se passa com o Eugénio esta aplicação também é composta por diversos componentes distintos, tais como o teclado, o *corpus* de símbolos pictóricos e um editor de texto, que poderiam por sua vez também ser de grande utilidade em aplicações do mesmo género.

Outro software de apoio a pessoas com necessidades especiais, mas desta vez essencialmente direccionado para estudantes, é o Caderno Escolar electrónico (CE-e) [1]. Este sistema que pretende ser um substituto dos tradicionais cadernos escolares em papel, disponibiliza aos seus utilizadores uma interface e variadas funcionalidades de *notetaking* organizado, permitindo a escrita de texto, inserção de imagens e hiperligações, *upload* de ficheiros e aplicação de fórmulas matemáticas, entre outras.

[1] Componente que apresenta uma matriz contendo os vários caracteres disponíveis para a composição de mensagens

2.3 Sistemas com Paradigmas Semelhantes

A ideia da criação de uma plataforma de auxílio ao desenvolvimento de aplicações de CAA não é propriamente nova, tendo sido nos anos 90 desenvolvidos esforços nesse sentido por parte do Consórcio *Comspec*. Este consórcio que reuniu uma equipa multidisciplinar constituída por educadores, engenheiros e programadores provenientes de diversos países europeus [8] tinha por objectivo a criação de uma plataforma que permitisse plena transversalidade entre quatro tipos de utilizadores (programadores, integradores de sistema, facilitadores e utilizadores finais).

Contudo, devido a necessidade de garantir o cumprimento dos requisitos necessários a todos estes utilizadores, acabaram por serem impostas grandes limitações na interface bem como na configuração de componentes, o que levou ao esquecimento do projecto, alguns anos mais tarde.

O projecto *Ulysses* tentou ser menos ambicioso que o *Comspec* tendo sido definidos apenas três tipos de utilizadores principais (programadores, integradores e utilizadores finais), no entanto o desenvolvimento do sistema acabou por não conhecer grandes avanços, tendo acabado por ser abandonado [7].

3 Apresentação e Descrição do Sistema

Como foi dito na secção 2, a maior parte dos sistemas de apoio à CAA foram desenvolvidos de forma monolítica, sendo a sua implementação, a sua alteração ou integração noutros sistemas bastante difícil. Este problema acaba por ter diversas origens, tais como o desenvolvimento de aplicações proprietárias, a falta de estrutura para reaproveitamento de código, ou ainda, a utilização em diferentes estruturas ou linguagens durante a sua implementação.

São estes os problemas que este projecto denominado de *Widget Augmentative and Alternative Communication Toolkit* (WAACT) pretende resolver, permitindo a criação de novas ferramentas de CAA, utilizando uma variedade de componentes gráficos predefinidos e configuráveis, comuns neste tipo de aplicações, tendo como principal objectivo de aumento da produtividade no desenvolvimento e a possibilidade de experimentar e avaliar novas abordagens aos sistemas durante a investigação.

3.1 Análise e Decisões

Durante a análise deste projecto foram surgindo diversas questões, tais como, a quem seria o sistema disponibilizado e quem o poderia desenvolver.

Assim, como o principal objectivo é a uniformização da estrutura das aplicações em questão bem como a reutilização de componentes já desenvolvidos e validados, verificou-se que a melhor forma de atingir esse objectivo seria através da participação no desenvolvimento de todos aqueles que utilizarão o sistema. Para isso, foram escolhidas ferramentas e linguagens de uso geral e abrangidos por licenciamento aberto, permitindo a utilização e possível desenvolvimento de todos.

Contudo, o uso de sistemas *Open Source* não era de todo suficiente, tendo em conta que a maior parte dos profissionais destas áreas acabam por trabalhar em sistemas e *softwares* proprietários, tendo sido assim necessário a utilização de tecnologias multi-plataforma, bem como a criação de uma estrutura modular que permitisse um desenvolvimento contínuo e flexível.

O desenvolvimento do *WAACT* é feito em *C++* e apoiado na *User Interface Framework, QT* [2], o que permite resolver alguns dos problemas atrás referidos.

Por outro lado, a necessidade de garantir a gestão dos eventos enviados por equipamentos de Input, processando-os e, se necessário, enviando-os para algum equipamento de Output, originou a criação de uma interface *Event Manager* que tem por missão tornar os módulos independentes dos eventos, permitindo o isolamento de cada *Widget*.

Sendo o *QT* também desenvolvido em *C++*, este *middleware* permite uma grande liberdade e interoperabilidade entre sistemas operativos, podendo assim ser desenvolvidas aplicações para as diversas plataformas existentes no mercado, com recurso á versatilidade que o paradigma da programação orientada a objectos oferece.

O *QT* dispõe ainda de um mecanismo de *SLOTs* e *SIGNALs* que se podem definir como uma alternativa às técnicas de *callback*. Estas são as funcionalidades centrais deste sistema e o maior aspecto diferenciador de outros *Frameworks*. Esta funcionalidade permite ligar funções a eventos, ou funções a outras funções, o que permite despoletar acções em qualquer tipo de evento predefinido ou em qualquer evento programado.

3.2 Utilização

Como já foi dito, foi necessária a utilização de um *middleware* que fosse multi-plataforma, de forma a não restringir a utilização de *software* desenvolvido a um determinado sistema, tendo sido o QT a plataforma escolhida. Esta escolha permite alguma flexibilidade ao programador, pois permite o desenvolvimento utilizando varias tecnologias. Contudo, pensamos ser importante proporcionar aos utilizadores do WAACT uma metodologia de implementação mais direccionada e que não obrigue a utilização de todas as tecnologias disponíveis na plataforma, como se pode ver no excerto de código abaixo apresentado.

Assim, numa óptica de programação orientada a objectos, foram criadas, para além dos *widgets*, funções que permitissem absorver as bibliotecas do QT, para que o programador apenas programe em C++ com recurso ao WAACT, libertando o utilizador de mais uma tecnologia.

```
W_KeyBoardKey *k = new W_KeyBoardKey();
k->keySized("B", "b", 50, 50);
W_TextPad *t = new W_TextPad();
t->textPad();
EventManager *e = new EventManager();
e->clickedKeyToTextpad(k, t);
```

4 Avaliação e Trabalho Relacionado

Numa primeira fase do projecto, foi importante avaliar a real mais-valia que poderia ser proporcionada por um *toolkit* deste tipo no desenvolvimento de software específico. Normalmente este tipo de avaliação é feita numa fase final de implementação, no entanto, sentimos a necessidade de validar as escolhas e caminhos percorridos; a decisão para esta validação passou pela implementação de algumas ferramentas já desenvolvidas noutras tecnologias de forma a daí tirar algumas ilações.

Uma dessas ferramentas foi o CE-e, da qual implementamos um protótipo bastante simples que apenas dispunha das principais funcionalidades de *notetaking*.

Para além da portabilidade entre vários sistemas operativos, tendo este protótipo sido compilado, executado e testado em Sistemas Microsoft, Linux e Mac (Figura 1); este teste demonstrou-nos que estas aplicações poderiam ser implementadas com um relativo baixo esforço de programação, desde que os *widgets* pretendidos estejam disponíveis no WAACT. Verificamos assim, que com poucas linhas de código, poderemos obter uma aplicação gráfica, mesmo que rudimentar, mas funcional.

Como já foi referido anteriormente, este projecto assentou no pressuposto da reutilização de componentes existentes noutras aplicações, bem como no teste de novas abordagens de integração dos mesmos, tendo sido adicionado ao protótipo um teclado de ecrã bastante semelhante ao do Eugénio (Figura 1). Neste caso fizemos a integração de um teclado QWERTY [2], no entanto, poderão ser utilizados ou criados

[2]Disposição de teclado adoptada pelos países ocidentais

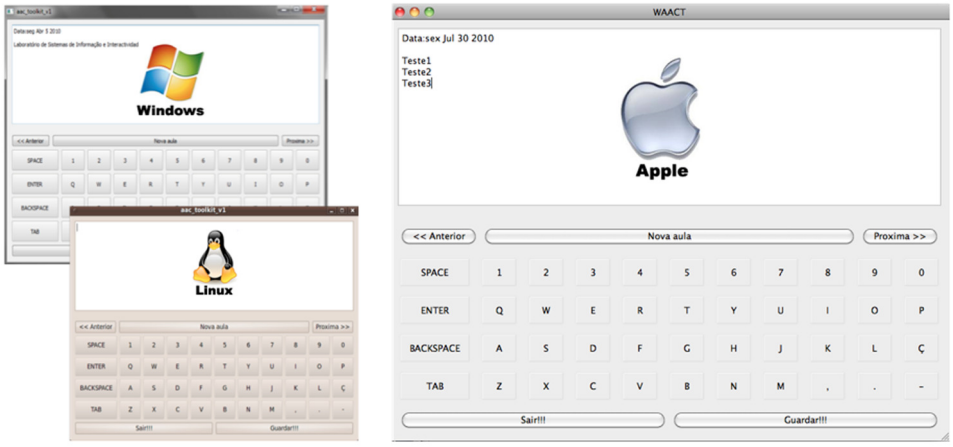


Figura 1: Protótipo de um sistema de *notetaking* desenvolvido em WAACT

outros, sendo que, o WAACT está preparado para qualquer disposição de teclas, podendo inclusivamente ser criado um teclado de raiz.

Baseado no protótipo anterior, a integração de um teclado virtual, que é um componente comum e bastante utilizado, necessitou apenas do registo de mais duas linhas no código já implementado. Assim, com base nestes dados, verificamos que desde que alguns dos componentes necessários ao desenvolvimento de determinada aplicação estejam implementados no WAACT, o aumento de produtividade e eficiência é visível no desenvolvimento de soluções deste tipo.

Por outro lado, é possível garantir flexibilidade para a investigação, podendo ser integrados ou trocados diversos componentes, ou ainda para o desenvolvimento, podendo ser avaliadas novas perspectivas por parte do programador.

5 Conclusões e Trabalho Futuro

Neste artigo propusemos e descrevemos o WAACT e qual o seu objectivo, passando ainda pelas necessidades existentes ao nível de *softwares* de CAA.

Actualmente dispomos de variados *widgets* implementados, tais como, teclados de ecrã e seus componentes, caixas *rich text*, paletes de botões, botões gerais e específicos, entre outros. Para além dos *widgets* propostos, dispomos ainda de um módulo chamado de *Event Manager* que gere os eventos de *input*, processando-os e despoletando um evento de *output*, caso seja esse o objectivo, permitindo ainda o desenvolvimento de componentes dependentes ou independentes dos já existentes.

Contudo, este é um trabalho que deve ser contínuo e atento às necessidades nas áreas de CAA, devendo ainda ser implementados vários componentes que permitam completar, pelo menos para já, o leque das necessidades actuais.

Pretendemos ainda, numa fase mais avançada, disponibilizar o *WAACT* em regime de *"Open Source"* bem como uma documentação adequada que permita uma fácil utilização e expansão por parte dos programadores.

No entanto, nesta fase, podemos desde já validar o QT como uma escolha bem sucedida no que diz respeito as questões de portabilidade, bem como, no que diz respeito a qualidade gráfica fornecida no desenvolvimento de aplicações.

Por fim, gostaríamos de agradecer a colaboração do corpo docente do Instituto Politécnico de Beja afecto ao Laboratório de Sistemas de Informação e Interactividade (LabSI^2), pelo contributo e ideias dadas para esta plataforma.

Referências

[1] Luís Alexandre, Luís Garcia, and Luís Bruno. Development of an electronic scholar notebook for students with special needs.

[2] Nokia QT Cross Platform Application and UI Framework. `http://qt.nokia.com`, disponível em Outubro de 2010.

[3] Serenella Besio and Lucia Ferlino. Blissymbolics software worldwide: from prototypes towards future optimized products. In *In Proceedings of the 2nd ECART Conference*, 1993.

[4] Peter F. Brown and Vincent Pietra. Class-based n-gram models of natural language. association for computational linguistics. *MIT Press*, 18(4), 1992.

[5] Luís Garcia. Concepção, implementação e teste de um sistema de apoio á comunicação aumentativa e alternativa para o português europeu, 2003.

[6] AAC Institute. `http://www.aacinstitute.org`, disponível em Outubro de 2010.

[7] Georgios Kouroupetroglou and Alexandros Pino. New generation of communication aids under the ulysses component-based framework. In *The 5th International ACM SIGCAPH Conference on Assistive Technologies*, 2002.

[8] Mats Lundalv. Comspec - the advent of an integrated modular communication system. In *Proceedings of the Future Integrated Solutions Conference*, Oxford, 1994. ACE Centre.

[9] Manuel Gericota. Ajudas técnicas á comunicação para pessoas com paralisia cerebral. `http://portal.ua.pt/bibliotecad`, disponível em Outubro de 2010, 1995.

[10] Ralf Schlosser, R. Koul, P. Raghavendra, and L. L. Lloyd. State-of-the-art in blissymbolics research: Implications for practice. In *In Proceedings of the 6th ISAAC Conference*, pages 121–123, 1994.

[11] Rose Scvcik and MaryAnn Romski. Aac: More than three decades of growth and development. 2000.

[12] Lda. Sensory Software International. `http://www.sensorysoftware.com/talkative.html`, disponível em Outubro de 2010.

[13] Gregg Vanderheiden and David Kelso. Comparative analysis of fixed-vocabulary communication acceleration techniques. *AAC Augmentative and Alternative Communication*, 3 (4):196–206, 1987.

Índice de Autores