Actas das 4as Jornadas de Informática da Universidade de Évora

JIUE 2014

Miguel Barão
Francisco Coelho
Salvador Abreu

u évora
ESCOLA DE CIÊNCIAS E TECNOLOGIA

Évora, Portugal
27–28 de Fevereiro de 2014

Actas das 4as Jornadas de Informática da Universidade de Évora — JIUE2014
Escola de Ciências e Tecnologia
Universidade de Évora
2014

http://www.di.uevora.pt/jiue2014/

Prefácio

As Jornadas de Informática da Universidade de Évora são uma iniciativa do Departamento de Informática, que tem este ano a sua quarta edição. É destinada a divulgar e promover os trabalhos na área da Informática realizados na Universidade ou contando com a participação dos seus elementos. Nestes trabalhos inclui-se, fundamentalmente, a investigação, básica ou aplicada, efectuada no âmbito de projectos ou no contexto de teses de mestrado e de doutoramento.

As Jornadas oferecem, aos alunos de pós-graduação da Universidade, um fórum amigável e alargado onde discutir o seu trabalho. O ambiente e a audiência, diferentes dos com que contactam habitualmente, permitem-lhes aumentar a confiança no resultado dos seus esforços e ganhar experiência na sua apresentação pública. Para os alunos da Licenciatura em Engenharia Informática, é a oportunidade de um primeiro contacto com a investigação realizada no Departamento e uma hipótese de encararem a vida académica sob uma perspectiva de futuro.

Esta edição das Jornadas conta com uma sessão especial "COBIT®Sessions", promovida pelo ISACA Lisbon Chapter, e dedicada às boas práticas de governança e gestão de TI, em particular as boas práticas da ISACA como o COBIT 5.

As presentes actas integram os artigos escolhidos de entre todos os submetidos aos dois tracks JIUE2014 e PhDI. A comissão organizadora agradece a todos os seus autores, que manifestaram o interesse em participar nas Jornadas submetendo-os e apresentado-os, e aos seus colegas do departamento e alunos de doutoramento que colaboraram no processo de selecção.

Miguel Barão
Francisco Coelho
Salvador Abreu

Évora, 27 de Fevereiro de 2014

Comissão de programa

Responsáveis

Miguel Barão
Francisco Coelho
Salvador Abreu

Membros

Carlos Caldeira
Irene Rodrigues
José Saias
Lígia Fernandes
Luís Arriaga
Luís Rato
Paulo Quaresma
Pedro Salgueiro
Teresa Gonçalves
Vasco Pedro
Vitor Nogueira
Alexandra Moedas
David Maia
Dora Melo
Francisco Guimarães
João Sequeira
Matheus Silveira
Nuno Miranda
Pedro Roque
Prakash Poudyal

JIUE2014

5ª-feira, 27 de Fevereiro de 2014

Sessão 1

Sessão 2

6ª-feira, 28 de Fevereiro de 2014

Sessão 3

Sessão 4

Seamless connection between question answering systems

Pedro Fialho*, Paulo Quaresma, and Luísa Coheur

[1] Universidade de Évora, Portugal
[2] L2F/INESC-ID Lisboa, Portugal

Abstract. Dialogue systems are envisaged as one of the main components of future human-computer interaction, allowing natural language communication to and from computers. Currently, Question Answering (QA) systems approach such a link, but lack a coherent dialogue. This PhD is aimed at the development of a modular/configurable dialogue system, covering multiple QA modules, semantic tools and paraphrase detection to allow a coherent dialogue.

1 Introduction

Dialogue systems are essential to future human-computer communication due to their ability to perform computational tasks (such as retrieving information) from natural human language inputs, eventually generating a human language response. Their aim is typically that of a Question Answering (QA) system with added dialogue capabilities that provide a continuous interaction experience. This PhD is aimed at a platform to generate text based dialogue systems, that includes multiple QA modules, covering open and closed domains.

Development of this platform implies that a coherent language and knowledge linkage among answers from the various QA modules is achieved, by implementing an history of interactions (for each user) and detecting paraphrases, using general purpose ontologies and synonym sets, among other semantic tools.

2 Baseline resources

This PhD includes understanding the basic usage/deployment of pre-existing QA related modules (without further research on their internal details – as a *black box*), namely SAYSOMETHINGSMART, EDGAR, JUSTCHAT and TALKPEDIA – all developed at the Spoken Language Laboratory of INESC-ID (L2F/INESC-ID) –, and connecting them so that an input is supplied to multiple modules and the chosen output is based on the confidence value reported by these modules.

SAYSOMETHINGSMART[3] (SSS) is an open domain QA having film subtitles as knowledge source, organized/extracted into pairs of questions and answers, using Lucene[4] to compare user inputs with questions and retrieve the respective answers. TALKPEDIA[5] is also open domain, where an answer is started by one of a set of predefined templates (such as "All I know is,") followed by a phrase from Wikipedia[6] containing words found in the input.

EDGAR[7] is a domain specific QA requiring knowledge to be defined through pairs of questions and their answers, which already exist for a basic agent/character profile (such as age and hobbies) and for some facts about the Monserrate palace in Sintra (from project FALACOMIGO, where the author participated), mostly applying distance metrics to decide which question is closer to the input. JUSTCHAT is not a QA module, instead it provides a semi-automatic form of building QA pairs used by EDGAR, by searching chatbot's logs (among others), which enables EDGAR to be open domain and easier to configure (such as for the profile).

* Corresponding author. Email: prpfialho@gmail.com
[3] http://www.l2f.inesc-id.pt/~pfialho/sss/
[4] http://lucene.apache.org/core/
[5] http://www.l2f.inesc-id.pt/~pfialho/talkpedia/
[6] https://www.wikipedia.org/
[7] https://edgar.l2f.inesc-id.pt/etutor.php

3 Planned tasks

Currently, this PhD is focused on improving SSS, since each user question in this system triggers an isolated search in the subtitles corpus, without considering previous user questions. Thus, some answers are contradictory and the dialogue may lack coherence, since no context processing is performed.

The ability to detect paraphrases is essential to ensure that answers to similar or related user inputs are not contradictory, detecting when a user repeats the same question in a different terminology. Currently, the questions "are you married" and "do you cook" yield contradictory answers, where the first is "no" and the second is "my wife cooks", which reveals the most difficult problem to solve, that will be approached with general purpose ontologies, such as YAGO[8]. Other problems can be solved with language normalization techniques (such as for questions with "what's" and "what it", which currently yield different answers) and synonyms sets, such as WordNet[9].

The planned approaches for these problems include typical semantic similarity techniques such as WordNet's word sense disambiguation and Latent Semantic Analysis – as found in SEMILAR[10], a toolbox for paraphrases – or more complex approaches such as mapping each sentence to a graph (such as by using Discourse Representation Structures, as retrieved by Boxer[11]) and comparing graph distances, thus reducing the search space in the subtitles corpus.

Initially, these tasks will be performed for English, where SSS is already functional and where more relevant resources can be found. However, a Portuguese version is also envisaged, depending on the complexity of adaptation of current resources and availability of relevant information.

Some QA modules don't implement a confidence value (such as TALKPEDIA) and, for those who do, confidence values are not comparable across modules, due to the lack of a common metric. As such, they rely on the confidence reported by internal mechanisms (as from ranking algorithms, as in SSS), whether isolated or by combining several values. One of the planned tasks is aimed at the development of a form of normalization for the various confidences of the available modules and a form of obtaining a confidence from a text output (for modules without confidence value), with the latter being more important since confidences may not be comparable.

4 Conclusion

From this stage it is expected to result a baseline development platform, to be used for the seamless connection among modules. The final platform allows connection of other modules, by specifying a REST based communication protocol which will be prepared in the following stage, being the main form of connection between currently available modules.

[8] http://www.mpi-inf.mpg.de/yago-naga/yago/
[9] http://wordnet.princeton.edu/
[10] http://www.semanticsimilarity.org/
[11] http://svn.ask.it.usyd.edu.au/trac/candc/wiki/boxer

Inteligência empresarial suportada em metadados vista como ontologia organizacional

Francisco Guimarães[1], Carlos Caldeira[2], Paulo Quaresma[3]

[1]Departamento de Informática, Universidade de Évora,
Francisco.santana.guimaraes@gmail.com
[2]Departamento de Informática, Universidade de Évora, ccaldeira@di.uevora
[3]Departamento de Informática, Universidade de Évora, pq@di.uevora

Resumo. O contexto da actividade económica actual é caracterizado por um grande volume de dados, estruturados e não-estruturados, com origem em processos e sistemas internos ou externos das organizações, envolvendo várias pessoas, com funções distintas, em entidades que fazem parte da cadeia de valor, mas em colaboração sobre os mesmos dados. Esta realidade leva a que existam várias interpretações sobre o significado dos dados e sua criticidade, o que associado ao volume e criticidade da informação pode levar a entropia. Neste contexto, torna-se fundamental conhecer a localização, estrutura e semântica dos dados, para melhor gerir e tomar decisões. Este conhecimento sobre os próprios dados corresponde ao conceito de metadados. Como as organizações são compostas por vários níveis de arquitectura interrelacionados, que se podem resumir em arquitectura de negócio e arquitectura técnica (aplicações e tecnologia de suporte), também os metadados necessitam de manter as duas perspectivas e devido relacionamento. Na perspectiva técnica, os metadados são amplamente utilizados em soluções de *Business Intelligence* enquanto suporte ao seu próprio funcionamento. No caso da perspectiva de negócio, onde é necessário manter descritores de indicadores de negócio baseados em dados, o conceito de metadados de negócio não tem sido tão amplamente utilizado. Por outro lado, a multiplicidade de sistemas operacionais e soluções de *Business Intelligence* na mesma organização, torna complexo o controlo e centralização destes metadados. Neste domínio de problema, existem várias abordagens de solução para gestão de metadados assentes em princípios de interoperabilidade, reutilização e vistos como ontologias. Este artigo, explora o conceito de implementação de metadados enquanto ontologia com a semântica de negócio definida por metamodelos por sector de actividade, modelados enquanto arquitectura empresarial. Esta abordagem é o que designamos por inteligência empresarial no contexto da tese de doutoramento .

Palavras-Chave: *business intelligence*, *datawarehouse*, metadados, semantic *web*, arquitecturas empresariais, ontologias, agentes inteligentes

1

Jornadas de Informática da Universidade de Évora, 2014

1 Introdução

As organizações modernas coexistem num mundo dinâmico de informação com integração entre processos internos e externos, criando uma rede virtual de empresas suportada em informação. Por outro lado, estas organizações têm uma realidade ao nível de sistemas de informação que resultam de uma amálgama histórica de implementação de vários tipos de aplicações, bases de dados e infra-estruturas.

Neste contexto, proliferam vários tipos de informação, com estruturas diferentes, em sistemas heterogéneos. Esta informação tem padrões de utilização assentes em diversos processos, com vários tipos de utilizadores, em diversas entidades internas e externas, com necessidades diferentes, sobre os mesmos dados.

Esta realidade, mais de que um problema de aplicações informáticas, é um problema de conhecimento sobre os próprios dados, na base da informação. Esta informação sobre as próprias estruturas e semântica dos dados, é designado por metadados.

Para uma gestão adequada deste tipo de dados, torna-se fundamental considerar que existe uma perspectiva técnica e uma perspectiva de negócio nas organizações, que deve ser vista de forma centralizada e interligada. Numa perspectiva técnica, consideram-se as aplicações e tecnologias de suporte às mesmas em termos de bases de dados, *software* e infra-estruturas de redes e comunicações, que fazem parte da arquitectura de sistemas de informação. Numa perspectiva de negócio consideram-se os processos que suportam uma semântica associada à lógica de negócio, muitas vezes codificada em programas informáticos, mas que faz parte do que designamos por arquitectura de negócio.

Na perspectiva metadados técnicos, estes sempre existiram nas próprias aplicações, e em particular em soluções de *business intelligence* para suportar o seu funcionamento. Neste domínio existe uma tendência de reutilização deste tipo de informação para melhorar os controlos de segurança, ao nível de *data discovery* e *data classification* para efeito de gestão de risco, integrado no conceito de *data governance*. Deve-se ainda destacar as tendências de soluções de *master data management* para controlar redundância de informação entre aplicações, implementado em soluções comerciais como a SAS [SASMDM] e Oracle [ORACLEMDM].

Na perspectiva de metadados de negócio, os sistemas de Business Intelligence são os que mais consideram a sua necessidade enquanto registo de transformações entre bases de dados origem e bases de dados como datawarehouse ou datamart, utilizando aplicações integradores como ETL (extracção, transformação e carregamento). São igualmente consideradas em descritores de lógica de cálculo de indicadores e detalhe de informação associada a relatórios apresentados aos utilizadores.

No entanto, estes metatados de negócio, são igualmente capturados e modelados em sistemas de arquitectura empresarial. Estes sistemas permitem a modelação da realidade da organização, e do ecossistema onde se insere, considerando a captura de modelos de estratégia, negócio, processos e mesmo de sistemas de informação (aplicações tecnologia e informação). Estes modelos são representados em cada perspectiva, mas igualmente na inter-relação de conceitos entre cada nível.

Ao criarem estes modelos, os sistemas de arquitecturas empresariais, criam na realidade uma visão da linguagem da organização, que em certa medida pode ser vista como uma reutilização de uma linguagem do sector de actividade onde se inseres. Por essa razão, existem algumas tendências de relacionar estes modelos de arquitectura empresarial com o domínio das ontologias como em [Kanga] e [Rajabi].

Esta conjugação entre arquitecturas empresariais e ontologias é o que pretendemos igualmente investigar, mas considerando igualmente que esses modelos vistos como ontologias empresariais, correspondem igualmente ao conceito de metadados. Ao considerarmos como metadados, queremos investigar a sua reutilização em sistemas de *business intelligence*, para definirmos um conceito de inteligência empresarial.

Neste contexto, este artigo, apresenta o estado da arte no contexto de metadados, arquitecturas empresariais e ontologias, associado ao problema de gestão de dados. Com estes domínios identificados, pretendemos enquadrar uma possível solução não intrusiva, que permita ter uma visão integrada de metadados técnicos e de negócio enquanto ontologia que servirá de base para criarmos um modelo de inteligência empresarial.

2 Metadados

Metadados corresponde ao conceito de dados sobre dados, visto como descrição de informação sobre as próprias de estruturas de dados numa organização incluindo bases de dados, programas ou classificação de arquivos documentais físicos. Pela definição de alguns autores de referência, podemos definir Metadados da seguinte forma:

- [Marco] define Metadados como conhecimento existente em aplicações ou em colaboradores, que representam aspectos internos ou externos à organização, incluindo informação sobre processos de negócio, regras e estruturas de dados;
- [Tozer] define o conceito de Metadados como algo relacionado com o comportamento e descrição de outros dados;
- [Chisholm] define Metadados como algo que se traduz nos dados que descrevem todos os aspectos activos de informação de uma empresa, permitindo-lhe utilizar e gerir de forma efectiva esses mesmos activos;
- [Blechar] define o conceito de Metadados como algo que consiste em informação sobre as características de um qualquer artefacto organizacional, tal como o seu nome, localização, importância atribuída, qualidade ou valor para a organização, assim como os seus relacionamentos com outros artefactos da organização.

Apesar de várias definições, existe um conceito comum associado a identificação de objectos, atributos desses objectos e relações entre os mesmos, sendo que os objectos representam dados. Por outro lado, face às características dos vários tipos de metadados, [Marco] agrega os mesmos em dois grandes grupos de acordo com a Figura 1:

- **Metadados Técnicos:** Relatórios, frequências de execução, tempo de execução, mapeamentos operacional/analítico, conversões de dados, modelos lógicos/físicos de dados, identificação campos/tabelas/índices/programas e gestão de versões;
- **Metadados de Negócio:** Estruturas de dados numa perspectiva de utilizador, definições de negócio mapeadas em campos, atributos de negócio mapeados em campos, estatísticas de qualidade de dados, regras de navegação pelos dados, localização/agregação temática dos dados.

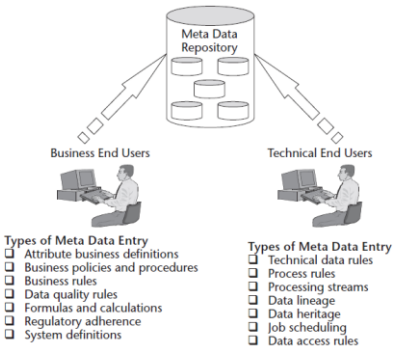


Figura 1. Tipos de metadados e utilizadores [Marco]

Existem várias normas para estruturar os Metadados, como é o caso de *Meta-Object Facility* [OMG], *Common WareHouse Metamodel* [CWM], *Basic Interoperability Data Model* [R.L.I Group] e o *Dublin Core Metadados Initiative* [DCMI]. Todas estas normas baseiam-se no princípio de que os activos de sistemas de informação como projectos, *software*, *hardware*, pessoas, processos e estruturas de dados, são fundamentais numa organização e têm estruturas próprias descritivas e de relação entre elas, que podem ser generalizadas.

Para a implementação de metadados recorre-se a extractores com base em várias fontes de dados, para consolidar o repositório e permitir a sua utilização por vários tipos de processos, aplicações e utilizadores. Este modelo referido por [Marco] na Figura 2 permite identificar o ecossistema e importância dos metadados vistos como repositório de conhecimento organizacional, em vez da visão de simples repositórios de metadados técnicos aplicacionais.

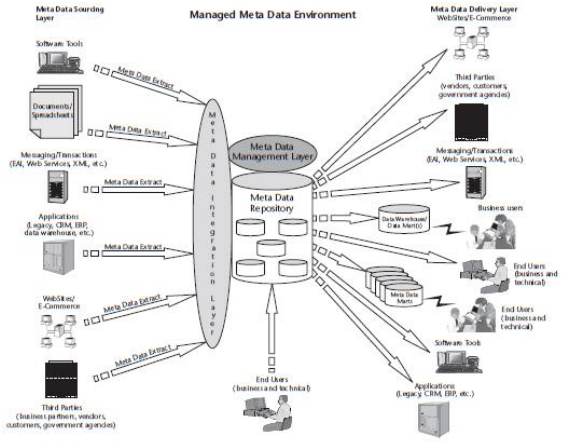


Figura 2. Ecossistema dos metadados [Marco]

3 Arquitectura empresarial

De acordo com a norma [IEEE Std1471] as arquitecturas empresariais podem ser definidas como "a organização fundamental de um sistema composta por componentes, suas relações internas e com o ambiente envolvente, e os princípios de governação do seu desenho e evolução".

Por outro lado, [Towers] refere que a arquitectura empresarial é utilizada com os seguintes objectivos:

- Suporte à implementação da estratégia;
- Clarificação de responsabilidade, tendo por base o conhecimento do posicionamento de cada função, das suas tarefas e da relação entre todos os componentes da organização;
- Definir uma *Framework* que permita identificar onde estamos, e para onde queremos ir, detalhando o caminho e monitorizando o processo de transformação.

Em termos de decomposição por níveis de arquitectura, como referido por [Rittgen] e suportado por uma metodologia e estrutura da arquitectura empresarial em [TOGAF], uma arquitectura empresarial pode ser decomposta pelo seguinte:

- **Arquitectura Organizacional:** Estruturas orgânicas e funcionais em termos de teoria organizacional. Pode-se considerar igualmente a definição da visão, missão, princípios e políticas gerais da organização. No entanto, deve-se considerar ainda a definição da oferta de produtos e serviços, e definição clara de canais de

distribuição e segmentação de clientes, que neste conceito pode ser designada como arquitectura estratégica, ou separando a arquitectura orgânica da arquitectura estratégica;

- **Arquitectura de Negócio:** Modelo de funcionamento detalhado em processos, actividades e tarefas para atingir os objectivos estratégicos e operacionais da organização;
- **Arquitectura Informacional:** Estruturas de informação que são utilizados pela organização, com preocupações de captura da sintaxe e semântica, independente das aplicações e processos que as utilizam e da forma como estão implementados;
- **Arquitectura Aplicacional:** Aplicações informáticas, em termos de serviços de suporte ou que permitem integrar e inovar no modelo de prestação de serviços ou integrados em produtos comercializados;
- **Arquitectura Tecnológica:** Infra-estrutura tecnológica considerando o suporte às aplicações informáticas, mas igualmente as redes, comunicações e serviços base tecnológicos para garantir a fluidez da comunicação em suporte digital.

Existem várias *Frameworks* com destaque para as seguintes:

- **Zachman:** [Zachman] Utiliza duas dimensões para definir os dados (*What*), as funções (*How*), a localização geográfica (*Where*), as pessoas (*Who*), o tempo (*When*) e a motivação (*Why*) associados aos artefactos. Esta dimensão é cruzada com o nível de detalhe em termos de âmbito (*Scope*), modelo de negócio (*Business Model*), modelo de sistemas (*System Model*), modelo de tecnologias (*Technology Model*) e representação detalhada (*Detailed Representations*);
- **MEAF:** *Metis Enterprise Architecture Framework*, criado pela empresa *Troux Techonologies* a partir de modelos de várias empresas, e que está na base da sua ferramenta de arquitecturas empresariais [Troux];
- **TOGAF ArchiMate**: A *framework* denominada *The Open Group Architecture Framework*, apresenta um modelo de arquitecturas de negócio, aplicações e tecnológicas, uma metodologia para a sua implementação e uma linguagem de notação que pode ser reutilizada por várias ferramentas, e que é designada por *Archimat* [TOGAF].

Apesar da sua relevância, este conceito tem sido descurado nas organizações derivado do esforço de actualização dos modelos. Existem no entanto algumas abordagens de implementação vendo esta arquitectura como ontologia e implementada com conceitos de semântica, nomeadamente ao nível do *Archimate* [TOGAF] enquanto linguagem de modelação [Azevedo]

4 Ontologia

Se considerarmos alguns sectores de actividade, existem termos que têm um significado estruturante enquanto modelo de negócio e objectos envolvidos. É o caso a título de exemplo do sector financeiro, onde conceitos de "cliente", "conta", "produto bancário", "rentabilidade", "segmento de cliente", entre outros, têm uma sintaxe e semântica própria. Esta linguagem está incorporada nos sistemas e no modelo de funcionamento de todas as entidades do sector financeiro.

A sua consolidação e visão comum pelas organizações são a base dos metadados de negócio. Qualquer desalinhamento de definição tem impacto enquanto problema de comunicação nas organizações ao nível de processos, pessoas e tecnologia, considerando possíveis automatismos que se queira implementar.

Estes metadados de negócio têm como raiz filosófica o conceito de ontologia. Isto porque a ontologia corresponde a uma descrição formal de uma área de conhecimento, ou domínio de conhecimento, a partir da qual se montam processos de comunicação com base no significado. Este formalismo permite a criação de automatismos nos processos.

O conceito de ontologia apesar de ter origem no domínio da filosofia, é utilizado igualmente em ciência da computação. Neste domínio, as ontologias são definidas como termos utilizados para descrever e representar áreas de conhecimento.

Para [Gruber] a ontologia é a especificação formal ou definição conceptual de ideias, conceitos, relações e outras abstracções no contexto de um domínio ou discurso. Como tal, é um vocabulário para utilização na linguagem nesse domínio, permitindo a comunicação e reutilização. A questão da formalização na definição da ontologia, torna-se fundamental no domínio da computação, pois permite legibilidade pelas máquinas. O conceito de ontologias é referido igualmente por [Heflin] para descrever vários tipos de artefactos, incluindo taxonomias e esquemas de metadados do DCMI.

Os componentes básicos de uma ontologia de acordo com [Corcho], são os seguintes:

- **Classes:** Correspondem a conceitos e utilizado em taxonomias, representando tarefas, acções, estratégias, processos de raciocínio, entre outros conceitos;
- **Relações:** Representam tipos de interacção entre as classes de um domínio, determinando uma classificação dessas relações enquanto "subclasse de", "conectada a";
- **Funções:** Tipo especial de relação onde o último elemento da relação, pode ser visto como um elemento determinístico face aos elementos precedentes;
- **Axioma:** Definem o significado e restrições, que permitem modelar expressões sempre verdadeiras;
- **Instâncias:** Representam elementos específicos, ou seja, os próprios dados.

Para tipificar as ontologias, existem várias formas, sendo que ressalvar a definição de [Gruber] que publicaram uma Framework específico no domínio da ciência da computação e Informação, agrupada por dimensões:

- **Dimensão semântica:** Relacionada com a especificação do vocabulário. Pode ser decomposta em níveis de estrutura (formalidade), expressividade da linguagem e granularidade dos conceitos;
- **Dimensão pragmática:** Relacionada com a finalidade e contexto da ontologia. Pode ser decomposta em níveis de intenção de uso, automação lógica para inferência, rigidez na caracterização e declaração de objectos/classes, ou por último, ao nível de metodologia para criar as próprias ontologias.

Com o advento da Web Semântica [Berners-Lee], utilizaram-se linguagens como o *Web Ontology Language* [OWL] e *Resource Descritpion Framework* [RDF] para implementação de ontologias na navegação World Wide Web. No entanto, existem outras linguagens para implementação de ontologias, como as seguintes:

- ***Topic maps:*** É um formalismo de representação de recursos organizados em tópicos, Associações enquanto relações entre tópicos, e Ocorrências enquanto relacionamento de tópicos com outros recursos relevantes [Librelotto];
- ***Ontology interchange language* (OIL):** Combina formalismo de *frames* com semântica formal e inferência lógica descritiva para classificação e taxonomia de conceitos, tendo evoluído para **DAML+OIL Agent Markup Language**;
- ***Simple knowledge organization system* (SKOS):** Aplicação do RDF para representação de *thesaurus* e tipos de sistemas de organização de conhecimento.

As abordagens de modelação de conhecimento baseadas em ontologias têm uma aplicação imediata a conceitos de arquitectura empresarial como forma de criar um modelo de metadados semântico em determinado domínio de conhecimento. Se considerarmos o domínio de conhecimento como sector de actividade como Banca, Seguros, Corretagem, Infra-estruturas, entre outros, a sua aplicabilidade permitirá consolidar uma linguagem comum dentro da organização e entre organizações do mesmo domínio de conhecimento.

Desta forma ficará facilitada comunicação interna e entre entidades, tendo por base o conhecimento, permitindo a criação de automatismos na comunicação suportada em sistemas de informação, e caminhando para modelos de inteligência empresarial.

5 Proposta de investigação

A proposta de investigação considera a necessidade das organizações terem uma visão integrada de metadados técnicos mas igualmente de negócio, num único repositório devidamente articulado entre todos os componentes. Este repositório deverá ter não só a estrutura e relação de conceitos, mas igualmente a semântica dos mesmos, numa perspectiva de negócio.

Para a encontrar uma forma de implementação destes repositórios no domínio da ciência da computação têm-se realizado várias investigações, que se podem agrupar da seguinte forma:

- Modelos de metadados que sejam reutilizáveis ou interoperáveis [CWM], [DCM]. Tornam-se complexos de integrar linguagens específicas de negócio de um sector de actividade e para uma entidade específica desse sector de actividade. Por outro lado, implicam alterações em sistemas já em funcionamento com custos de adaptação elevados;
- Modelos de metadados de negócio e técnicos ao nível de ferramentas de *business intelligence* para mapeamento de indicadores face a campos em bases de dados, mas igualmente para captura de transformações e cálculos com impacto nos indicadores. Utiliza-se igualmente para controlo de qualidade de dados, como é o caso de conceitos *master data management*, *data governance*, *data profiling e data lineage*. No entanto, têm fraca incorporação da visão de semântica de negócio integrada nos metadados, sendo mais focado para qualidade de dados com algum potencial ao nível de definição de regras de negócio, mas sem reutilização por vários tipos de agentes;
- Modelos de metadados com incorporação de semântica e suportada em ontologias para permitir não só a centralização de dados, mas dotar igualmente este repositório de inteligência organizacional, para adaptação dinâmica ao contexto interno e externo onde se insere. Nesta linha, tem-se destacado uma aproximação entre metadados e arquitecturas empresariais, nomeadamente através da linguagem padrão *Archimate* [TOGAF]. Algumas preocupações a este nível estão relacionadas com encontrar técnicas e linguagens mais adequadas para cada nível de arquitectura (negócio, aplicacional, informacional, tecnológica), respectiva integração e análise dos modelos assim criados, como referido por [Morais] e [Kanga].

No entanto, estas preocupações, não endereçam as seguintes questões:

- Como criar uma ontologia alinhada com modelos por sector de actividades, que se constituam numa linguagem específica de negócio para facilitar a gestão de conhecimento interno e interoperabilidade entre sistemas, ou agentes, comprometidos com esta linguagem?
- Como instanciar uma ontologia automaticamente com base em vários metadados técnicos e funcionais, derivado da inevitabilidade da existência de vários sistemas aplicacionais, como *packages* ou desenvolvimento à medida, em áreas aplicacionais ou de informação de gestão, criando uma arquitectura informacional, vista como ontologia ou metadados organizacionais?
- Como ajustar os metadados organizacionais, com base em informação de especialistas internos, para garantir que corresponde à realidade específica da cada organização?
- Como reutilizar este modelo pelas aplicações, vistas como agentes comprometidos por estes metadados ou ontologia organizacional, com particular ênfase em sistemas de *business intelligence* em processos de ETL (Extracção, Transformação e

Carregamento) e na navegação através de sistemas de exploração de dados (*reporting, dashboard*), usufruindo da semântica assim construída?

Para responder a estas questões, propomos uma visão de solução por incorporação de semântica em ontologias suportadas em linguagens de arquitecturas empresariais com objectivo de reutilização em contexto de metadados em aplicações de *business intelligence*.

A nossa proposta de investigação, acrescenta níveis de extensão destes conceitos, já parcialmente aplicados ao repositório de arquitecturas empresariais. No entanto, a nossa proposta tenta garantir que igualmente os extractores de informação e os utilizadores desta arquitectura de informação sejam agentes com inferência. Por outro lado, tentamos que seja possível uma edição dos metadados para ajuste específico para cada organização, com base num metamodelo por sector de actividade.

Para tal, a estrutura da nossa proposta, de acordo com a Figura 3, considera o seguinte:

- ETL Semântico enquanto extractor subordinado às próprias regras de negócio dos metadados e com inteligência ao nível de captura por eventos, por padrões de produção dos dados e com potencial de acrescentar aprendizagem. Desta forma, a nossa solução é não intrusiva para as fontes de dados e sistemas de informação existentes;
- Utilização de agentes inteligentes para suporte a utilizadores no contexto de navegação por informação de gestão, ou transaccional no contexto de processos de negócio. Desta forma, consegue-se posicionar os metadados no contexto de utilização.

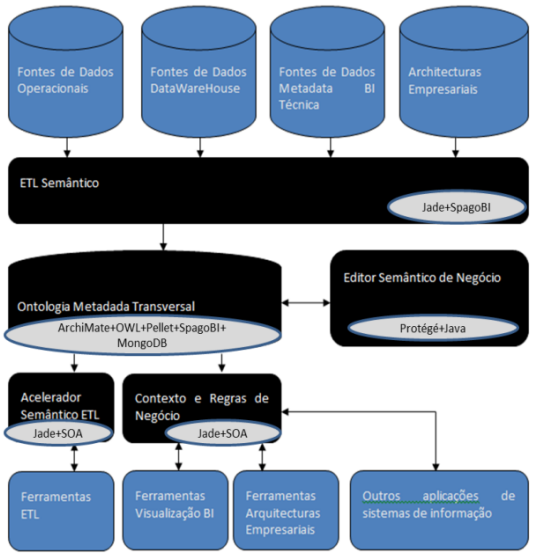


Figura 3. Modelo Proposto

Na base da solução, serão implementados metadados com base em metamodelos por sector de actividade, instanciados por reverse do conhecimento destes metadados via comportamento da própria organização. Este conhecimento será capturado por extractores, sendo ajustados com o conhecimento tácito através de um editor semântico. Estes metadados assim modelados, permitiram por sua vez retroalimentarem o consumo, mas igualmente a aprendizagem da extracção e consolidação de metadados vistos como ontologia organizacional.

Para a implementação do modelo serão utilizadas tecnologias de suporte a ontologias como o RDF no repositório de metadados, a partir do qual se pode desenvolver agentes suportados em JADE [Jade] e Pellet [Pellet] para implementar a inferência, e motores de ETL suportados em SpagoBI [Spago].

6 Conclusão

Este artigo apresenta o estado da arte de metadados, ontologias e arquitecturas empresariais enquanto domínios do problema da implementação de metadados como ontologia transversal numa organização. Este conceito de ontologia transversal, visto como metadados organizacional, tem por base o pressuposto de existir um problema nas organizações com proliferação de vários tipos de sistemas com metadados heterogéneos.

O artigo detalha as abordagens de implementação de metadados técnicos e de negócio, com as desvantagens inerentes a alguns modelos, optando por seguir uma abordagem de investigação baseada em metadados como ontologias organizacionais derivados de modelos de arquitecturas empresariais.

No entanto, apresenta uma solução de construção dinâmica da ontologia com base em metamodelos específicos por sector de actividade, a partir do qual as organizações podem ajustar o seu próprio domínio de conhecimento.

Com esta abordagem é possível construir dinamicamente os metadados como ontologia, e usufruir deste universo de discurso para retroalimentar os extractores de metadados e os consumidores, em vários processos. Este conceito assim implementado, é o que designamos por inteligência empresarial, com base em metadados vistos como ontologias.

Este modelo será montado em três casos de estudo em três sectores de actividade diferentes (Banca, Infra-estruturas e Corretagem), testando a construção dos metamodelos, a inferência para captura dos metadados e a inferência como indutor de conhecimento no contexto de navegação em sistemas de *business intelligence*.

Referências

1. [Azevedo] Azevedo,Carlos, Almeida,João, Van Sinderen, Marten, Quartel, Dick, Guizzardi, Giancarlo. An Ontology-Based Semantics for the Motivation Extension to ArchiMate. Disponível em http://nemo.inf.ufes.br/files/an_ontology_based_semantics_for_the_motivation_extension_to_archimate.pdf. Último acesso em 12/12/2013.
2. [Blechar] Blechar, M. Metadata Management Technology Integration Strategues, 2H05 to IH06. Gartner Research.
3. [Berners-Lee] Berners-Lee, T, Hendler, J, Lassila, O. The Semantic Web. Scientific American: the Semantic Web. Disponível em www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21. Último acesso em 12/12/2013.
4. [Corcho] Corcho, O, Fernandez-Lopes, M, Gomez-Perez, A. Methodologies, Tools and Languages for Buidling Ontologies. Where is their meeting point? Data & Knowledge Engineering.
5. [Chisholm] Chisholm, M. Repositories Build or Buy. MPO Newsletter.
6. [CWM] Common Warehouse Metamodel Specification, Volumes 1 & 2. Disponível em http://www.omg.org/. Ver também http://www.cwmforum.org/. Último acesso em 12/12/2013.
7. [DCM] DCMI. The Dublin Core Metadata Initiative. Disponível em http://dublincore.org/. Último acesso em 12/12/2013.
8. [Gruber] Gruber, T. Toward Principles for the Design of Ontologies Used for Knowledge Sharing, In:Guarino, N., Poli, R.(Eds.):Formal Ontology In Con-ceptual Analysis and Knowledge Representation, Klwer.
9. [Guimaraes] Guimarães, F. Utilização de sistemas multi-agentes em processos de negócio e sistemas de workflow no domínio da memória organizacional. Tese Mestrado. ISCTE.
10. [Heflin] Heflin, J. OWL Web Ontology Language: Use cases and requirements. W3C Recommendations.
11. [IEEE Std1471] Systems and Software engineering - Recommended practice for architectural description of software-intensive systems: ISO/IEC 42010:2007(E).
12. [Jade] Java Agent Development Plataform. Disponível em http://jade.tilab.com. Último acesso em 12/12/2013.
13. [Kanga] Kanga, Dongwoo, Leeb, Jeongsoo, Choib, Sungchul, Kim, Kwangsoo. An ontology-based Enterprise Architecture. Disponível em http://www.sciencedirect.com/science/article/pii/S0957417409006368. Último acesso em 12/12/2013.
14. [Librelotto] Librelotto, G. Topic Maps: Da Sintaxe à Semântica.
15. [Marco] Marco, D, Jennings, M. Universal Meta Data Models. Wiley Computer Publishing.
16. [Morais] Morais, André. Aplicação de Ontologias à Representação e Análise de Modelos de Arquitectura Empresarial. Dissertação para Obtenção de Grau de Mestre em Engenharia Informática e de Computadores.Instituto Superior Técnico. 2013.
17. [OMG] OMG. Object Management Group – MetaObject Facility (MOF). Disponível em www.omg.org/mof/. Último acesso em 12/12/2013.
18. [OWL] OWL Web Ontology Language. Disponível em www.w3.org/TR/owl-features/. Último acesso em 12/12/2013.

19. [ORACLEMDM] Oracle Master Data Management. Disponível em http://www.oracle.com/us/products/applications/master-data-management/overview/index.html. Último acesso em 12/12/2013.
20. [Pellet] Motor de Inferência. Disponível em http://clarkparsia.com/pellet/features. Último acesso em 18/12/2013.
21. [Rajabi] Rajabi, Zeinab, Minaei, Behrouz, Seyyedi, Mir Ali.Enterprise Architecture Development Based on Enterprise Ontology. Disponível em http://www.scielo.cl/pdf/jtaer/v8n2/art07.pdf. Último acesso em 12/12/2013.
22. [Ritgen] Rittgen, P. Enterprise Modeling and Computing with UML. IDEA Group Publishing.
23. [RDF] RDF Vocabulary Description Language 1.0: RDF Schema. Disponível em www.w3.org./TR/rdf-schema/. Ultimo acesso 12/08/2013.
24. [R.L.I Group] Group R.L.I. The basic interoperability data model. Disponível em https://kspace.cdvp.dcu.ie/respository/doc/bidm.html. Último acesso em 12/12/2013.
25. [Spago] Open Source Business Intelligence Suite. Disponível em http://www.spagoworld.org/xwiki/bin/view/SpagoBI. Último acesso em 12/12/2013.
26. [SASMDM] SAS Data Management Software. Disponível em http://www.sas.com/en_us/software/data-management.html#master-data-management. Último acesso em 12/12/2013.´
27. [Tozer] Tozer, G. Metadata Management for Information Control and Business Success. Archet House, Inc.
28. [Tolbert] Tolbert, D. CWM: A Model-based Architecture for Data WareHouse Interchange. Disponível em www.cwmforum.org/uciwesas2000.htm. Último acesso em 12/12/2013.
29. [Towers] Towers, S, Burlton, R. In Search of BPM Excellence: Straigth from the Thougth Leaders.
30. [Togaf] TOGAF v9, The Open Group Architecture Framework. Disponível em www.Togaf.org. Último acesso em 12/12/2013.
31. [Turco] Turco, C. Enterprise Architecture & Metadata Modelling: A Guide to Conceptual Data Model, Metadata Repositoty, Business and System Re-Enginerring. ISBN-0-7414-5304-5.
32. [Troux] Troux Enterprise Architecture. Disponível em http://www.troux.com/. Último acesso em 12/10/2013.
33. [Zachman] Zachman, J. The Zachman Framework™ Evolution. Disponível em http://www.zachmaninternational.com/index.php/ea-articles/100#maincol. Último acesso em 12/12/2013.

Software Transactional Memory implementation for PaCCS (Parallel Complete Constraint Solver)

David João Maia

University of Évora, Portugal
d11331@alunos.uevora.pt
February 23, 2014

Abstract. Parallel execution has been gaining a predominant place in the computing world. In a few years, we have seen tremendous progress in processors execution power, leaving areas such as memory management and processes synchronization in parallel execution systems to explore.
PaCCS is a "Parallel Complete Constraint Solver" and was designed with the aim of exploiting parallel execution in hierarchical systems through constraint solving technics. Although load balance and effective process synchronization tasks by the worker threads are crucial, also the memory organization and scalability are equally important factors when it comes to parallelism. This paper aims to present an implementation of the current PaCCS, in a context where the organization and memory management model are based on software transactional memory (STM), having as target symmetric multiprocessors (SMP) and distributed architectures.

Keywords: Software Transactional Memory (STM), Constraint Satisfaction Problem (CSP), Constraint Optimisation Problem (COP), Parallel Execution, Symmetric Multiprocessors (SMP);

1 Introduction

In concurrent systems, many processes may compete for the right to access shared resources. In order to enable coherent data access, it is required the use of synchronization mechanisms. These mechanisms employ techniques such as locks or critical sections. Its operation implies that when an object is locked by a process or thread, no further process can modify that object while the lock is found active.
On the other hand we have critical sections, that forces that only one thread can execute a particular block of code at a given time. These mechanisms have some flaws, presently well identified. During application development, it is the developer's responsibility to ensure that during the threads synchronization no problems occur, for instance [HM]: deadlocks, convoying or priority inversion.

According to [ARAtS], programs that use conventional lock mechanisms or mutual exclusion are not easily scalable. It is easy to understand that the greater the number of threads disputing access to resources, the greater the number of threads that will be queued to access the locked resource or enter the critical section. Thus leading to an implementation bottleneck.
The transactional memory mechanism was born in order to take advantage of the rapid evolution of multicore processors and multithreaded hardware techniques, such as SMP. It is important to clarify that transactional mechanisms are not new, as they are already used for several years in database management systems. The concept of transactional memory has been gaining greater importance since it has an amazing potential in supporting parallel execution. A major objective of this mechanism is the removal of the programmer's responsibility for managing locks and processes synchronization. Passing this responsibility almost exclusively to the software transactional memory.

An important factor for the parallel execution is the concept of scalability, in which transactional memory, theoretically, has a great development potential. Programs using unrefined conventional lock mechanisms clearly suffer from scalability problems. As the number of threads grows, so does the race for resources, which in turn increases the waiting times.
For programs using refined locks, according to [HM,ARAtS], they can scale better, but under the penalty of an increase in the program's complexity, amplifying its maintenance difficulty.
Systems like PaCCS, aim to take advantage of hardware techniques such as multithreading and distributed architectures. In either case, scalability is an important factor that theoretically helps to improve performance. However, currently, this is not entirely true. There's a limit in which the use of more threads will not introduce a higher performance, possibly leading to a worse overall application performance.
Through a software transactional memory implementation, it is possible to optimize the scalability factor of PaCCS like systems. This idea is not entirely original, as there are already several implementations of transactional memories, not for PaCCS like systems, but for generic libraries. However, the main motivating factors of this work were:

1. "How can we optimize scalability factor in multithreaded systems using a software transactional memory without performance penalty?"

2. "How can we take advantage of a software transactional memory whose variables are constantly and concurrently blocked for writing?"

The second statement is clearly the worst case scenario where a transactional memory can be. Let's take as an example the operation dynamics: the transactional memory is designed to allow multiple threads to be able to modify subsets of shared variables, typically different among them. It was not designed to have all threads blocking the shared variables permanently.

When applied the transactional memory technique to the PaCCS system, one can, however, foresee some gains, both in terms of memory management, lock mechanisms, network traffic and scalability. The next sections will be devoted to describing the general operation of a software transactional memory as well as the current PaCCS architecture. It will also be proposed a modification in the PaCCS architecture, in which it will manage its memory using a software transactional memory, with the aim of achieving a greater performance without compromising its scalability.

2 PaCCS - Parallel Complete Constraint Solver

PaCCS is a constraint solver designed to solve problems through restrictions (CSP/COP) on systems with [Ped12] shared or distributed memory. It aims to search the solutions space of the formulated problems in parallel. May it be through local multithreading (SMP), remotely (distributed) or a mixture of the two.

2.1 Present architecture

In the implementation reported by [Ped12], PaCCS is organized into teams, each of which has two layers: one for controller and another for the workers. In the first layer we have the controller, whose objective consists in managing the communication and distribution of work between teams (remotely) or workers (locally). In the second layer we have the workers, whose function is to explore the entire search space by looking for solutions. Figure 1 gives an overview of its architecture.

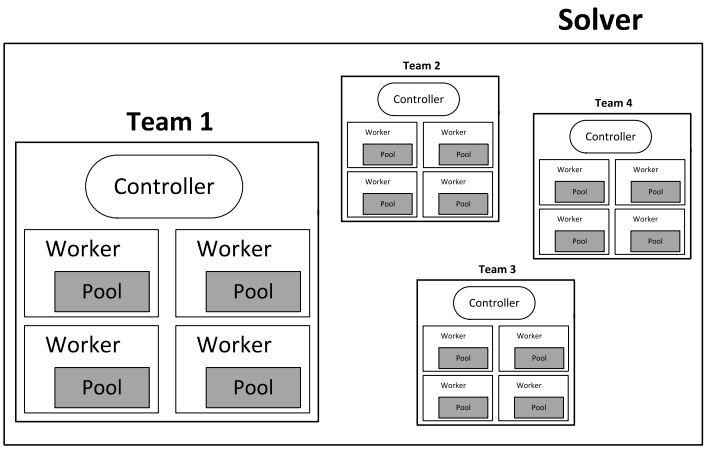


Fig. 1. Solver architecture

2.2 PaCCS execution dynamics

During PaCCS execution process, one can follow three different targets:

1. Find a solution to the problem (CSP);
2. Count the number of existing solutions to the problem (CSP);
3. Find the best solution to the problem (COP).

In any of the objectives, the operation has two phases: problem formulation and resolution. During the formulation phase all the variables, their domains and constraint sets between variables are created. After the problem formulation, the resolution phase begins, which starts by dividing the search space by the teams. In turn, each team will again divide the problem into subproblems sending them for each worker. The division of the search space is a powerful feature of PaCCS, since through this technique it is possible to parallelize all the search.

2.3 Memory Organization

The search of the solution space is done in parallel, which means that at certain key times, a worker may try to access shared resources owned by another worker or team. Thus, in the memory model adopted, it was chosen that the workers would share some features, mainly:

1. Problem representation (variables, domains and constraints);
2. Pools with search spaces;
3. Variables for control locks;
4. Best current solution.

Note that each worker has a shared pool, which is only accessed by other workers when there is need to steal work to another worker. Figure 2 shows the organization of a pool where all search spaces are stored.

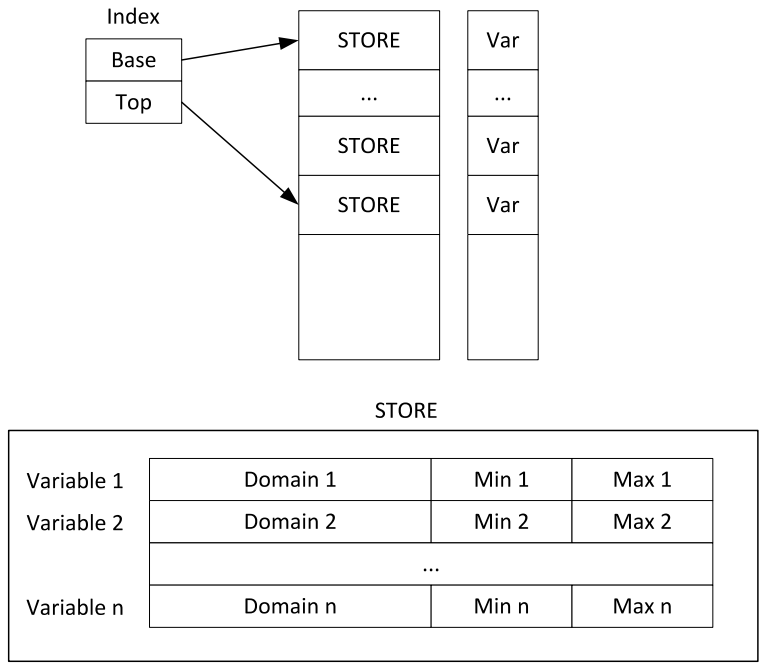


Fig. 2. Solver Pools

However, each worker has a set of private variables used exclusively for themselves, namely: current search space, dynamic information about restrictions or internal validations [Ped12].

3 Proposal

Systems that work with massive volumes of data, such as, OLTP (Online Transaction Processing) or OLAP (Online Analytical Processing) systems, with a huge set of changes in shared variables, clearly take advantage of transactional memory mechanisms. May they be: hardware transactional memory (HTM), software transactional memory (STM) or hybrids. These type of scenarios do not apply to PaCCS reality, since the amount of data and their changes are not massive enough to require a transactional memory approach. However, nothing prevents that its memory management from being done by a transactional mechanism in order to support parallel execution.
Through STM, it is possible to introduce an efficient memory management capable of managing read and write operations to a smaller set of variables, but highly requested. Given this scenario, the objective of this paper is to propose a software transactional memory model, applied to the execution context of PaCCS, capable of maintaining or improving current performance and scalability.

3.1 Transactional Memory

A transaction is a block of code executed by one or more threads whose execution is atomic. During its execution, the thread attempts to write the results to the shared memory. In case possible, it is said that the transaction has COMMITED, updating the value of the shared variable (globally visible) and making the value visible to all other transactions. If the transaction fails to COMMIT, it is said that the transaction as ABORTED or did a ROLLBACK.
There are two basic attributes that define the essence of the transaction mechanism, namely: atomicity and serialization. The serialization introduces the concept, in which no transaction is aware of the order in which they are executed, knowing only that are executed sequentially.

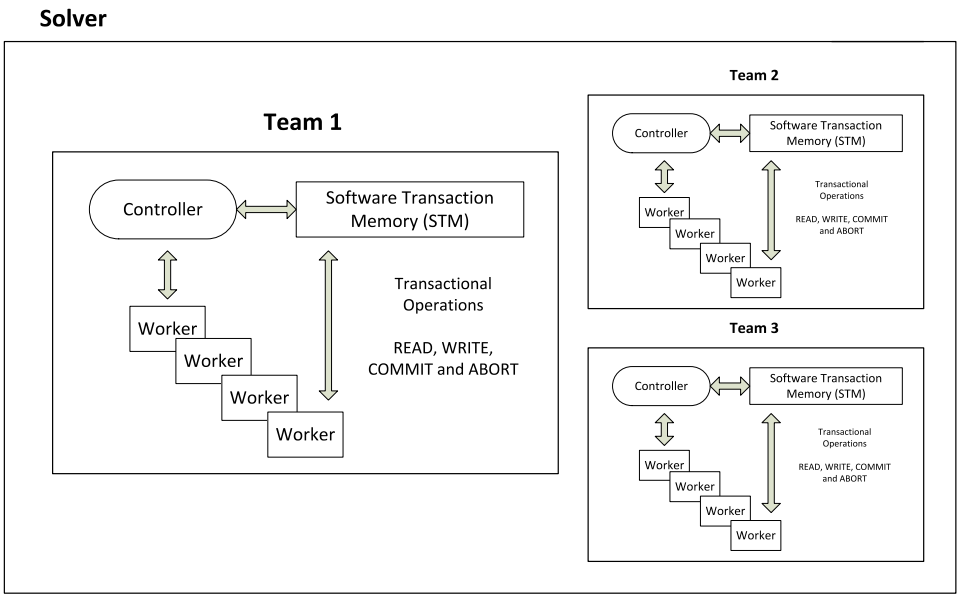


Fig. 3. New solver architecture

Atomicity states that when a transaction COMMITS, the modified variables are provided in a comprehensive and atomic form. Associated with the basic structure of transactional memory are two mandatory mechanisms: the lock engine (logical locks) and a version control mechanism.
The logical lock mechanism is used to manage access to shared resources, allowing one or more transactions to access the same resource at the same time. It is commonly found in implementations, different types of locks, examples: read lock, write lock or exclusive lock. Let's look to the following example, multiple transactions may share a read lock on a shared variable, but only one transaction can hold a write lock. The exclusive lock or optimistic locks (some applications use these naming) usually are associated with transactions that begin by requesting a read lock, knowing in advance that they will request a write lock later.
The second mechanism is responsible, for each write lock, keeping versions of shared variables modified by the transaction. Thus, when a transaction COMMITS, the value of all the changed variables are updated atomically, making it visible to other transactions. On the other hand, when the reverse operation is generated, ABORT or ROLLBACK, the changes made by the transaction on the shared variables are discarded from the version control mechanism (CVS), having no impact on any other executing transaction.

3.2 Proposed Architecture

The current architecture uses a local, but shared pool of search spaces for each worker thread, reducing contention when accessing to shared structures to a minimum. However, every time a worker local search space pool becomes empty, the worker will try to steal work from another co-worker or from another team, by accessing its shared structures.
With the new architecture, the scenery changes completely, leaving out the local pool (which does not mean that there are no local variables in each worker). Thus, all pools and global variables are now being managed by the STM. Figure 3 presents the new architecture of PaCCS featuring software transactional memory.

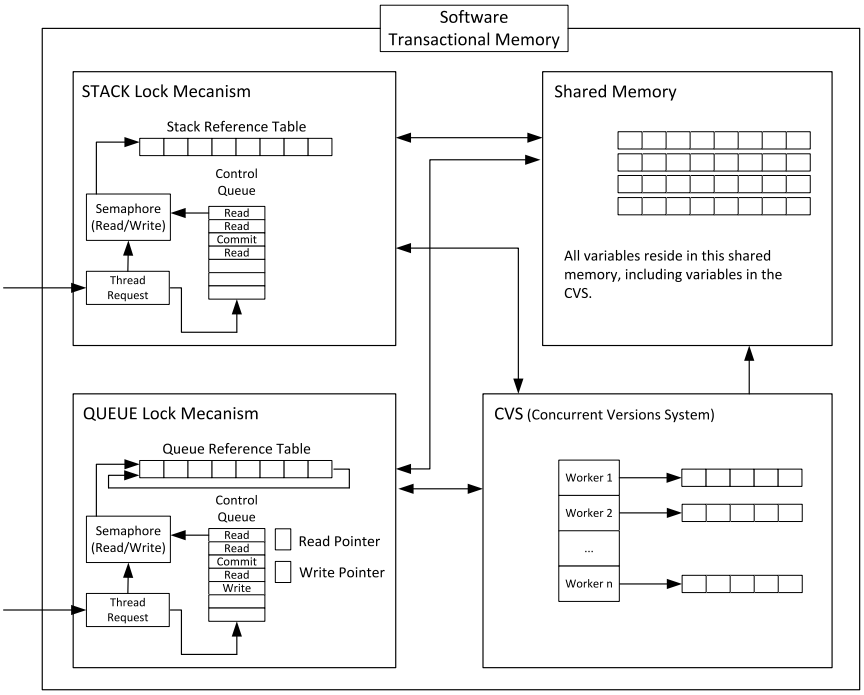


Fig. 4. STM architecture

With this organization, to some extent, we tried to preserve the original two-layer model. The bottom layer consisting of workers keep their original function, which is the search engine for the solver. The most noticeable modification is applied to the top layer, responsible for the internal communication of the workers and between teams. Initially this layer was composed only by the controller. Now it has a new member, the STM, which will aim to feed all workers and the controller with information.

Notice that in this new model, all workers in a team will access directly the same STM memory, increasing contention and data race to access shared resources, possibly leading to worst overall performance. This model is only viable if the memory operations granularity is as thin as possible. One of the primary objectives of the STM is exactly this, reduce to a minimum possible the blocking operations in any memory access.
In the following section we'll discuss an implementation that enables the use of this architecture.

3.3 Implementation details

The memory design proposed in this work was influenced by the nature of PaCCS variables. The way STM internally manages these variables affect the overall performance of PaCCS. Thus, we choose to classify the type of existing variables as follows: global variables to hold the combination of values with the solution of the problem and dynamic variables, considered local stores in the old architecture (for each worker), with the search spaces yet to be processed.
The first type of variables has a more static or reactive nature, for instance, they are only modified when the solver finds a solution, otherwise they are just used to query the objective function. The variables that keep the search spaces, according to [Ped12], called STORES, have a more dynamic nature, since they are constantly being required for reading and writing, being discarded after processing.
For this scenario, two lock mechanisms were created for internal STM memory management purposes: one for the memory management similar to a STACK and another for memory management working as a processing QUEUE, as shown in Figure 4. Both lock mechanisms,

STACK and QUEUE, have only references to memory addresses, of existing variables in the shared memory of the STM.

The handling of complex data structures during read and write operations are undesirable in highly concurrent systems, since their blocking actions can compromise the parallel execution. To avoid such problems, it was decided to create a shared area, whose aim is to maintain and make available, through an access API its content to all transactions (threads). All READ, WRITE, COMMIT or ABORT operations only manipulate references (integers and pointers), being considerably lighter handling, since no variables with active locks will be copied. The access or copy of the shared memory information is always made after the release of any holding lock. Only the transaction (thread) in question will wait for loading the contents of its variables.

The only identified bottleneck, will occur on every STM access operation, by team, by passing through a semaphore (POSIX pthread_mutex_lock), to assign a logical STM blocking mask, LOCK_READ or LOCK_WRITE, to a variable identified in the control structure of the mechanism associated memory (STACK or QUEUE). The locking mechanism, STACK and QUEUE, aims to manage the logical blocks requested by transactions. Allowing each transaction to attempt to block a variable with a logical read or write lock.
This implementation only supports two types of locks: LOCK_READ and LOCK_WRITE. Since the access patterns and blockages are relatively similar, there is no need to create additional types of locks. In this context, access patterns follow a policy of all or nothing, with the exception of the search spaces STORE variables, which are:

1. CSP – When a worker finds a solution, he sends a request for a LOCK_WRITE lock on all variables (present in the solution) on the STACK mechanism for update;

2. CSP – Counting the number of solutions, whenever a worker finds a solution, it sends a lock request LOCK_WRITE to update counter variable (in this option the counter is the only variable attended by workers in the STACK);

3. COP – When a worker finds a potential solution, he sends a request for a LOCK_READ on all variables (present in the solution) on the STACK mechanism to validate the best solution. If the solution is not better, the lock is released;

4. COP – When a worker finds a potential solution, he sends a request for a LOCK_READ on all variables (present in the solution) on the STACK mechanism to validate the best current solution. If the solution is found better, the transaction will issue a COMMIT which will generate a LOCK_WRITE locking all variables.

With this type of policy, the workers are constantly competing to access the same shared data. This is clearly the worst state in which a transactional memory can be found. However, its transactional design allows multiple threads to continue its execution concurrently, even if the data is modified. The access patterns of the search spaces have a different search pattern, the access is always exclusive, but variables order is irrelevant, since the objective is to search all search spaces.
Please note that during the execution of a worker, instantiation and propagation, search spaces that do not satisfy the constraints are discarded from memory. Although the lock mechanisms are similar, their internal operation is quite different. Let us take a look to the QUEUE lock mechanism, as shown in Figure 4. This mechanism possess semapahore (POSIX pthread_mutex_lock) whose objective is to control access to logical STM blocking structures performed by each transaction. Internally, this mechanism has an array designed to keep the history of transactions and their holding lock. There is another array, whose function is to map the logical lock type to a variable reference, in the CVS or shared memory. Access to the second array is controlled by two variables that point to the next available read or write index.

After the current search space is fully processed, new search spaces are generated and added to the QUEUE memory, being the current search space discarded, even if it has valid solutions or not. Thus, the idea for this table is to be less complex to manage as possible without keeping search loops or table reordering. This is where the two pointers are used, whenever a transaction locks an object in the QUEUE table mappings, the "ReadPointer" value is updated to the next available index, based on the value at which it is. Once the value of the pointers exceeds the array size, the counting is reset back to the beginning of the table. If the number of objects (generated search spaces) exceeds the size of the available array, a new array is created with the double size of the current array while preserving all existing mappings (a similar approach to hash tables).

The STACK lock mechanism has a similar structure, except the pointers. There are also two arrays, one to store the transaction history and their active locks, and another one to store the references to the variables in the CVS or shared memory.
The CVS module is simply a mechanism for versioning control. Its operation consists in indexing variable references per transaction, keeping all the modifications on global variables. Whenever a READ request is executed, the STM validates if the variable in question has been modified by the transaction, if so, the returned value is read from CVS. If not, the return value is read directly from its active value in the shared memory. When the STM receives a WRITE request (not to be confused with COMMIT), memory creates a new version in the CVS module, becoming the variable used for this transaction.
Regarding to the STM shared memory process, it is still under close study, however, it is thought that the best approach is: through implementation of a POSIX shared memory or by using a programming language based on PGAS (Partitioned Global Address Space), as the UPC (Unified Parallel C). It is assumed that a good implementation of a UPC will be the best approach, enabling the creation of a STM synchronization mechanism, on different teams, similar to existing ccSTM (cache coherence STM). For resource integration of STM with PaCCS was defined an API reference:

Function	Description
STM_DEQUEUE	Get next value of QUEUE memory. Logical LOCK_READ, non-blocking action.
STM_ENQUEUE	Adds a new value in the CVS associated to QUEUE memory. Logical LOCK_WRITE, non-blocking action.
STM_ADD	Creates a new variable in shared memory and adds a STACK reference in the locking mechanism.
STM_REMOVE	Removes a variable from the shared memory and removes the reference in the STACK locking mechanism.
STM_READ	Get data from memory STACK (reads from global memory the first time, all other remaining times reads from CVS). Logical LOCK_READ, non-blocking action.
STM_WRITE	Write value of a shared variable in the CVS associated transaction. Logical LOCK_WRITE, non-blocking action (writes only in CVS).
STM_COMMIT	Locks the selected memory (STACK or QUEUE) and updates all reference variables in the selected memory existing in the CVS. Blocking action.
STM_ABORT	Locks the selected memory (STACK or QUEUE) and invalidates all references variables and locks on the memory selected existing in the CVS. Blocking action.
TRANSACTION_BEGIN	Begin a new transaction and prepare associated structures in the CVS.
TRANSACTION_END	End transaction and start the cleaning process. Variables validation declared in the CVS and their locks.

In this implementation, the STM memory does not possess a concept of ROLLBACK, which implies that if a transaction ABORTS, the entire transaction fails and needs to be

reprocessed. In these cases, all the references in CVS are discarded and recreated again during the reprocessing. On the other hand, there's the inverse function, COMMIT, that whenever is executed, only the lock mechanism passed as argument is blocked for update (STACK or QUEUE). In case of the QUEUE memory, the impact is minimal, since all variables written are new and are allocated by the CVS in the creation process. Thus, the COMMIT operation modifies only references. In the case of the STACK memory, the COMMIT operation is a little bit more tricky. This requires updating the addresses indicated by the variables in STACK mechanism.
A secondary effect generated by a COMMIT in the STACK memory (possibly problematic) is the eventual possibility to force an ABORT on all running threads, forcing the reprocessing of the current search space. This is not necessarily true, but in the worst case, all remaining threads may fail to process the current search space and become forced to restart processing. Let's take a look at the following example: a team with 16 workers, of which 15 are processing their current search space. The remaining worker finds the solution and issues a COMMIT. All the workers that are still in the instantiation or propagation phase will not feel any impact on their execution. Only the workers who have locked the same variables in the STACK memory, usually to execute the objective function, will fail. The reason for the failure is due to the fact that when the value of the solution variables were read by a set of transactions, were being changed, at the same time by a COMMIT of another transaction. So these running transactions should fail and be reprocessed.
As opposed to traditional models, this model favors writing to reading operations. Allowing some variables with a read lock active, to receive lock request for writing. The only negative effect is when a COMMIT is executed, several aborts are generated for running transactions. Hence, the larger the number of solutions generated, the higher the performance impact.

4 Related Work

The original version of PaCCS was designed to work on SMP and distributed systems. The model presented in this paper focuses on SMP systems, but also allows its operation in distributed context (with some limitations such as synchronization between STM's).
There are currently several implementations of STM [RLBJC,HM] for parallel execution systems or clusters [RLBJC]. In this paper, was proposed an implementation of a software transactional memory, inspired by existing works in the area. The difference lies in the focus of the problems, which are derived almost exclusively from a parallel execution oriented to constraint programming reality.
During the development of this research, some emerging techniques and models in the area of parallel execution on distributed context, were identified as useful to improve both the performance and scalability of such systems. Examples of such techniques are:

- PGAS (*Partitioned Global Address Space*);
- RDMA (*Remote Direct Memory Access)*;
- GPI (*Partitioned Global Address Space Programming Interface*).

The PGAS model enables defining a physical address space shared between multiple processors. That is, two or more processors can map to the same physical addresses of an physical memory shared device. This shared memory needs not be at the same physical location, and may be distributed over a network. According to [RLBJC], programming languages based on PGAS, such as UPC or X10, can achieve better performance and scalability in distributed context. In [RLBJC], the author has proposed an implementation of an STM environment based on GASNet.
The model proposed in [RLBJC] is very interesting, as base for exploring the synchronization of the STM's, currently deficit, using an implementation by UPC as opposed to the current MPI.

However, there are other approaches, such as GPI [Mac13] (based on PGAS but oriented to clusters), also applied in the context of parallel execution in a declarative programming oriented reality. These programming models, exploit the use of RDMA for process synchronization. This is an equally interesting model, with enormous potential for optimization of process synchronization in distributed STM context.

5 Conclusions and future work

With the present work, it was possible to expose an implementation of a software transactional memory, where the execution context was oriented to constraint solving problems. Fundamental aspects have been covered on the structure and organization of the PaCCS constraint solver, as well as aspects of software transactional memory. Another goal of this work was also trying to achieve a better performance for PaCCS execution in SMP architectures, which is still in progress.

Currently, PaCCS supports distributed execution, however, with this new model, there are still some areas that need improvement, in particular the synchronization support between STM's. In the near future, would be quite interesting to support automatic distributed synchronization between STM's teams. Thus, allowing features such as "remote steal" or load balancing, could become implemented in a transparent manner.
Techniques such as PGAS, GPI or RDMA are considered strong candidates for future implementations, adding support for distributed operation of the STM's. There are, however, other interesting techniques such as synchronization via cache coherence algorithms, in which, theoretically, can also achieve high performance on architectures with NUMA (Non-Uniform Memory Access).

Bibliography

[ARAtS] Christos Kozyrakis Ali-Reza Adl-tabatabai and Bratin Saha. Unlocking concurrency.

[FRW] Peter van Beek Francesca Rossi and Toby Walsh. *Handbook of Constraint Programming*. Elsevier.

[HM] Maurice Herlihy and J. Eliot B. Moss. Transactional memory: Architectural support for lock-free data structures. In *Proceedings of the 20th annual international symposium on Computer architecture (ISCA '93)*.

[Mac13] Rui Machado. *Massively Parallel Declarative Computational Models*. PhD thesis, Universidade de Évora, Julho 2013.

[MMS] Neelam Goyal Luke Yen Mark D. Hill David A. Wood Michael M. Swift, Haris Volos. Os support for virtualizing hardware transactional memory.

[Ped12] Vasco Pedro. *Constraint Programming on Hierarchical Multiprocessor Systems*. PhD thesis, Universidade de Évora, Maio 2012.

[RLBJC] Vikram S. Adve Robert L. Bocchino Jr and Bradford L. Chamberlain. Software transactional memory for large scale clusters.

[RR] Thomas Rauber and Gubula Runger. *Parallel Programming for Multicore and Cluster Systems*. Springer.

[RRL] Maurice Herlihy Ravi Rajwar and Konrad Lai. Virtualizing transactional memory.

The influence of Training data on the performance of the SVMCMM algorithm

Prakash Poudyal, Teresa Gonçalves, and Paulo Quaresma

Department of Informatics, University of Evora, Portugal
prakashpoudyal@gmail.com,tcg@uevora.pt,pq@uevora.pt

Abstract. Supervised machine learning algorithms need training data to infer a function but the ultimate performance of the inference may be impacted with the use of quality of data instead of quantity only. In this paper, we focus on using training data in an iterative process to train a Support Vector Machine Conditional Markov Model (SVMCMM) and measure its performance. The proposed methodology is applied to a corpus of legal documents from the Euro- Lex site in which instances of plaintiff were analyzed. The performance is analyzed through precision, recall and f-measure. Result indicates that the quantity of training data is not only sufficient to get the best results but also a need of good quality data sets.

Keywords: Plaintiff, Training data, machine learning algorithm, legal domain

1 Introduction

A human being gets to know many things of the world after getting exposed to the information. His perception on those things depends upon on the magnitude of data and facts present in the information. Without getting exposed to the information a person cannot learn much. Similarly, supervised machine learning algorithm uses sets of training data to create a constraint and pattern to determine new data sets. Based on the nature of data sets balanced or unbalanced, quality and quantity vary the performance of supervised machine learning algorithm. So, to achieve optimal performance supervised machine learning algorithms must depend on the large number of balanced and qualitative training data sets.

On the basis of above assumption, an experiment was conducted to analyze plaintiffs which divided into 3 categories (government, organization and person) from 890 file cases of Euro-Lex site[1]. From previous work, we found Support Vector Machine Conditional Markov Model (SVMCMM) algorithm [4] as the best algorithm to determine named entity type person and organization so in this experiment we use it.

[1] http://eur-lex.europa.eu/en/index.htm

The paper is organized as follows: section 2 briefly describes related work; section 3 describes the data set used for the experiment; section 4 presents the experimental setup for the iteration process and section 5 the methodology implemented; section 6 describes the results and discussion finally, section 7 presents some conclusions and points out possible future work.

2 Related Work

Some related works are discussed below:

P. Quaresma and T. Goncalves [2] described a mixed approach using linguistic information and machine learning techniques to identify entities in the legal documents. Documents were available in four languages viz English, German, Italian and Portuguese.

Straneiri and Zeleznikow [3] described several approaches of applying data-mining technology in legal documents; they also discuss the trends of solving legal information extraction problem from machine learning techniques to natural language processing methodologies.

TinTin et. al. [1] proposed a system called Legal TRUTHS (TuRning Unstructured Texts to Helpful Structure) that extracts relevant information from Philippine Supreme Court decisions, specifically on criminal cases in English version. All together 25 training and 50 test documents were used. The system reuses existing tokenizer and named entity recognizer of Balie which is an open-source project [8]. Balies NER tagger is a dictionary-based tagger, which relies on lexicons to determine the entities. Automatic filtering of the data was involved in drawing out relevant information from the texts. The result shows that, precision and f-measure values are above 90% where recall is 84.3%.

Moen et. al. [5] conduct a project name SALOMON that automatically summaries Belgian criminal cases in order to improve access to the large number of existing and future court decision. In Sparck Jones [6] two strategy were applied in SALOMON: the first was to shallow processing and the other is a deeper processing. The extracted summary of the case is inductive and information and is 20% of the full cases.

3 Dataset Description

In this work, we used legal documents from the European Lex web portal that consists of 2815000 documents from the year 1951 till now. The database is updated daily and almost 12000 judicial documents appended every year. The website has 23 official languages of the European Union. The documents are freely accessible from the European Union web portal.

The documents are in static pages and have similar pattern of structure divided into three parts: first section consists of the CaseID reference and the title of the decision; second section provides the information about court staffs; and finally third section consists of the judgment decision. For this experiment, we

took only plaintiff which lies in the first section of the document. In average there are 2 to 5 plaintiff instances per document, but in some of the cases more than 10 are found. Plaintiffs are divided into three categories which are *government*, *organization* and *person*. In total, out of the analyzed 890 legal documents, 1322 where organization instances, 539 government and 272 person instances only. Most of the instances are found in swerve pattern. It is classified as good quality and low quality. If the instance are much closer to each other and has some common words in each phrase, than it is considered as good quality whereas, if the instance are in different language and has special character in the words than it is considered as a low quality.

4 Experimental Setup

Minorthird. The machine-learning framework Minorthird [7] is an open source software tool collection of Java classes for storing and annotating text, extracting entities and categorizing text. Minorthird is user friendly and supports both graphical interface and terminal base while conducting the experiments. It is quite different from other framework because it supports many machine learning algorithms. Minorthird can take several formats of text document, but it prefers plain text with XML tags. There are all together ten ML algorithm, but we selected the best algorithm to classify plaintiff according to the paper [4]: Support Vector Machine Conditional Markov Model (SVMCMM). Experiments were conducted with the default parameters of Minorthird.

Support Vector Machine Conditional Markov Model. SVMCMM is the combination of two very powerful machine-learning algorithms: Support Vector Machine (SVM) and Conditional Markov Model (CMM). SVM is trained from the feature provided and CMM is trained from the output of the SVM, which results on spans of text rather than simply the individual tokens. The working principle of SVMCMM is shown in figure 2.

Performance measures. Precision [9] is defined as the number of relevant documents retrieved divided by the total number of documents retrieved of the positive class; recall is defined as the number of relevant documents retrieved divided by the total number of elements that actually belong to the positive class. F-measure is the harmonic mean of precision and recall and belongs to a class of functions used in information retrieval. F_β can be written as

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall}$$

5 Methodology

Figure 4 explained about the methodology of the iteration process of the experiment. At first, we took 100 training documents which are manually tagged

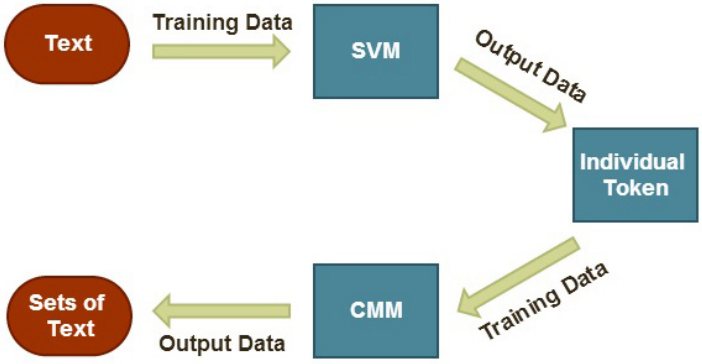


Fig. 1. Working principle of SVMCMM

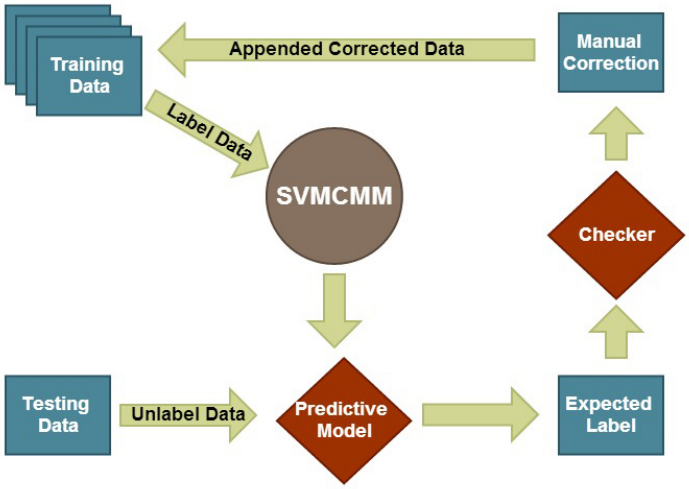


Fig. 2. Methodology of the experiment

with person, organization and government and trained SVMCMM algorithm. For testing data an average of new 100 raw legal documents was taken and provided to SVMCMM algorithm to determine its performance through the predictive model. SVMCMM classifies the instances, which are manually evaluated (counting true positive, false positive and false negative). If the instance of plaintiff is scored for true positive, it is left as it is, but if false positive and false negative are scored it is corrected. After that, the corrected testing data is appended to the previous training data, obtaining a new larger training set for the next iteration. The process continues until the 8^{th} iteration. The number of training documents and testing documents is given in the table 1. Similarly, in table 2 we can find the number of training instance and testing instances of each entity with its iteration number.

Experiment	Training documents	Testing documents
1^{st}	100	97
2^{nd}	197	98
3^{rd}	295	100
4^{th}	395	100
5^{th}	495	100
6^{th}	595	100
7^{th}	695	99
8^{th}	794	96

Table 1. Number of training data and testing data with its iteration number

	Training Gov	Testing Gov	Training Org	Testing Org	Training Person	Testing Person
1^{st}	50	62	172	197	42	14
2^{nd}	112	90	369	148	56	21
3^{rd}	202	67	517	168	77	37
4^{th}	269	45	685	159	114	32
5^{th}	314	99	852	167	146	40
6^{th}	413	54	1038	186	186	71
7^{th}	467	58	1259	221	257	39
8^{th}	525	64	1480	144	296	38

Table 2. Number of training instance and testing instances of each entity with its iteration number

6 Result and Discussion

After analyzing 890 legal documents, we found total numbers of 2218 instances of plaintiff out of which 13.20 % are person, 24.30 % are government and 62.50 % are organization. The European lex court site consists in 23 official languages and for this experiment we selected only English. But most of the plaintiffs' instances are in their own respective national language which causes the structure to be in diverged pattern. Along with this, there are other several circumstances for the result to be in disparity which is discussed below according to the class wise.

6.1 Government

There are all together 539 instances of government entity. In first iteration, f-measure is 0.687 which is very low than other iteration. As the training data increases the performance of SVMCMM of determining instance of government increases as we can observe from the graph in figure 3. However, in 3rd iteration the graph decline a bit from the 2nd, because in this data set, instances are found in several national languages such as Portuguese, Italian, German, French which were not appear in training data before but first time appear in testing data. Some of the such instance that are not determine by the SVMCMM are in case "C-140/99" plaintiffs which are in Italian language "Presidenza del Consiglio dei Ministri" and "Fallimento Traghetti del Mediterrano Spa"; similarly, in case "C-194/08", "Bundesminister f r Wissenschaft und Forschung" is in German language and in case "C-526/08", "Grand Duchy of Luxembeurg" is French.

Nevertheless, most of the instance of government has common words in the noun phrases such as "Ministry", "Republic", "Government", "Kingdom". Some of the instance are illustrated such as in "Case C-521/07" the instance of the government is determined as <gov>Kingdom of the Netherlands</gov>, similarly in case of "C430/07" <gov>Minister van Landbouw</gov> and case "C-397/07" has <gov>Kingdom of Spain</gov> and<gov> Hellenic Republic </gov>.

Overall the graph of government is in progressive order and finally in 8th f-measure value is 0.968 except in some exceptional case such as in 3rd iteration.

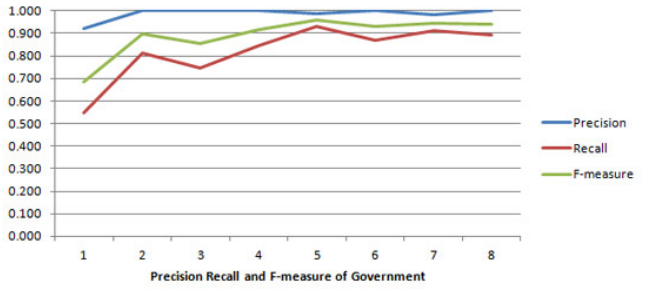


Fig. 3. Graphical representation of precision, recall and F-measure of Government

Finally, we can conclude that, the performance of SVMCMM over government is satisfactory.

6.2 Organization

From the table 2, we can find that instances of organization is much more large number than the instance of government and person which provoke of class imbalance problem. There are several demerits of having class imbalance. For example, in 4^{th} iteration, out of 152 instances, the algorithm was able to tag 134 instance of organization, which is correct, but 25 instances of person and government are tagged as organization but only 18 were not tagged as organization. As a consequence, false positive is higher than false negative. There are several other reason for organization scoring bit low than government as well, such as language barrier in the detection process, presence of acronym in the instance, lengthy name in instance. Few of the examples of the error are illustrated.

Error type 1: False Positive (Instance tagged of person and government as organization)

- In case "C586/08" instance <org> Angelo Rubino</org> tag as organization but "Angelo Rubino" is a name of person.
- In case "C376/08" instance <org> Comune di Milano</org> tag as organization but "Comune de Milano is the municipality of the Milan, Italy which must have government tag.
- In case "C403/09 PPU" instance <org>Jasna Detiček</org> tagged as organization but "Jasna Detiček" is a name of person.

Error type 2: Language barrier for the detection. Some of the instance of organization are written Portuguese, German, Spanish, French, Slovenian, Icelandic, Italian, Danish etc with their special character which are not common to other language. Few of them are illustrated

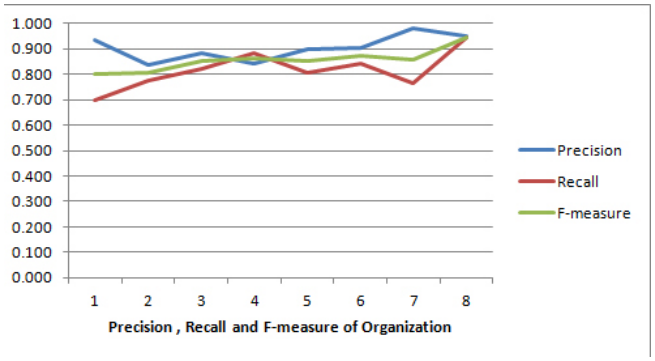


Fig. 4. Graphical representation of precision, recall and F-measure of Organization

- In case "C262/10" has plaintiff name "Döhler Neuenkirchen GmbH" was not able to detect because of the German character " ö " in the instance which is quite different to any other language.
- In case "C137/08" has plaintiff name "VB Pénzügyi Lzing Zrt, is is in Hungarian language which was not able to detect by SVMCMM algorithm.

Error 3: False Negative (Very long instance which were not able to detect)

- In case "C145/10" the German company "Verlag M. DuMont Schauberg Expedition der Kölnischen Zeitung GmbH & Co KG" has long name which was bit different than the other instance of the organization.
- Similarly, in case "C567/10" "Atelier de Recherche et d'Action Urbaines ASBL.

Error 4: False Negative (existence of acronym in instance which create problem in the detection.)

- In case number "C124/10P", the plaintiff "EFTA Surveillance Authority" is the Iceland authority company that monitors compliance with European Economic Area rules in Iceland, Liechtenstein and Norway.
- In case "C179/11" there is a French organization "Groupe d'information et de soutien des immigrés (GISTI)".

On the other hand, there are several instances of the organization which have some common words such as "Commission", "Gambh", "Ltd", "SpA", "Inc". If the above words are available in the name of the organization, SVMCMM's accuracy rate will be higher. For example, in case "C487/07" has instance <org>Malaika Investments Ltd<org> and <org>Starion International Ltd<org>) and similarly, in case "C558/07" has instance <org>Hercules Inc.<org>.

We conclude that, even having a large number of instance than any other class, the entity organization has a mix of good quality as well as low quality data. The reason why the performance of the SVMCMM at the 8^{th} iteration is f-measure 0.947 which is quite satisfactory result.

6.3 Person

Large variation is found in the instance of person categories. Out of 2218 instances only 292 of instance are person class. Some of the names are written in the national language of plaintiff's origin which consists of several characters in the words which do not exist in other languages. For example in Case "C-42/11", the plaintiff name is "João Pedro Lopes Da Silva Jorge" – in the first name the character ã in the word João which is a Portuguese character that is not found in any other country's language. Similarly, in case "C-515/08", there are all together 3 person names which are "Vtor Manuel dos Santos Palhota", "Mário de Moura Gonçalves", "Fernando Luis das Neves Palhota" and "Termiso Limitada"; instance has character "á" in word Mário and similarly "ç" in word "Gonçalves". Similarly, some of the name of the person are in short form, for

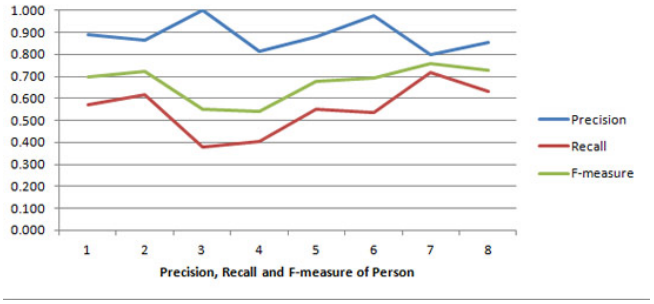


Fig. 5. Graphical representation of precision, recall and F-measure of Person

example in case of "C-345/09", "J.A. van Delft" and "O. Fokkens"; here, the first name has a shortcut form.

Therefore, the graph in figure 5 is up and down. In several iteration, we cannot find improvement. The f-measure value of 3rd iteration decline than of 2nd reached to 0.549. In this iteration , there are 30 instances in which 15 are tagged but rest 15 are not tagged. Similarly in 4th iteration the f-measure score 0.542 in which out of 33 instances of person only 15 were tagged and 2 instance of organization are detected as person and 18 were not detected. Similar kinds of correlation are found in 5th and 6th iteration.

Finally, we can conclude that most of the instances have no common words and also no hints as the phrases of government and organization used to have and also number of instances of person was very low. Therefore, if we take note of f-measure at the 8th iteration, we still found f-measure at the level of 0.719 which is very low in comparison to other class which indicates that there is plenty of space to improve it.

7 Conclusion and Future Work

From the results and discussion, we can conclude experiment till 8th iteration process which has consequence impact in the quality and quantity of the data. So, we conclude that optimal performance of supervised machine learning algorithm can be achieved by the availability of large quantity of quality training data. In future work, there is need to increase the performance of SVMCMM to detect person instances. To accomplish the work, external datasets of pronoun must be used to train supervised machine learning algorithm helping to find proper data.

8 Acknowledgment

We would like to thank Prof. Dr. Salvador Abreu and Mr. Mohammad Moinul Hoque, Mr. Roy Bayot of University of Evora for several suggestion and Prof. Daniel Diaz of the University of Paris-1, France for providing very sophisticated server to conduct experiments.

References

1. Tin Tin Cheng, J.L. Cua, M.D. Tan, K.G. Yao, and R.E. Roxas. Information extraction from legal documents. In Natural Language Processing, 2009. SNLP 09. Eighth International Symposium on, pp 157 162, (Oct 2009)
2. Paulo Quaresma and Teresa Goncalves. Using linguistic information and machine learning techniques to identify entities from juridical documents, 2010.
3. Andrew Stranieri and John Zeleznikow. Knowledge Discovery from Legal Databases (Law and Philosophy Library). Springer, September 2005.
4. Prakash Poudyal and Paulo Quaresma: An hybrid approach for legal information extraction. Proceedings of the 25th International Conference on Legal Knowledge and Information Systems, Amsterdam, Netherlands, JURIX, pp. 115–118. IOS Press,(2012)
5. Marie-Francine Moens, Caroline Uyttendaele, and Jos Dumortier. Abstracting of legal cases: the salomon experience. In Proceedings of the 6th international conference on Artificial intelligence and law, ICAIL 97, pages 114122, New York, NY,USA, 1997. ACM.
6. Paul S. Jacobs. Using statistical methods to improve knowledge-based news categorization. IEEE Expert: Intelligent Systems and Their Applications, 8(2):1323, April 1993.
7. William W. Cohen. Methods for identifying names and ontological relations in text using heuristics for inducing regularities from data. Available: http://minorthird.sourceforge.net.
8. Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. Scientific American, 284(5):3443, May 2001.
9. Salton, Gerard and McGill, Michael J. Introduction to modern information retrieval : McGraw-Hill,1983

Sistema de análise de sentimentos em mensagens Web

Jorge Miguel Ferreira Letras

Universidade de Évora, Portugal
m8844@alunos.uevora.pt

Resumo As redes sociais são plataformas em larga escala onde pessoas de todo o mundo se podem conhecer, partilhar imagens e vídeos ou trocar opiniões. Hoje em dia são gerados diariamente pelos utilizadores, milhões de opiniões ou comentários relativamente a produtos ou serviços. O objetivo deste artigo é apresentar um sistema com a capacidade de determinar, através de técnicas de aprendizagem automática, o sentimento geral demonstrado por uma mensagem, classificando-a como sendo positiva, negativa ou neutra. As caraterísticas principais e inovadoras deste sistema são os valores da entropia das palavras presentes na mensagem.

Keywords: Reputação online, análise de sentimentos, entropia

1 Introdução

O sucesso das redes sociais juntamente com as novas ferramentas e utilizações da *Web 2.0*, causaram algumas alterações na forma como as pessoas comunicam e partilham informação entre si. Atualmente existem diversas plataformas *online* que permitem a partilha de todo o tipo de informação entre utilizadores de todo o mundo como os *blogs*, *micro-blogs*, redes sociais ou serviços de análises a produtos e serviços.

Com a quantidade e diversidade deste tipo de conteúdo podem-se desenvolver análises diversificadas acerca da opinião de utilizadores sobre vários domínios como político, económico ou social. As análises retiradas sobre essa informação podem ter bastante utilidade quer para delinear políticas de mudança numa empresa, esclarecer consumidores insatisfeitos ou analisar o sucesso de eventos, de produtos ou até de personalidades políticas em altura de eleições.

O objetivo deste trabalho é apresentar um sistema inovador de deteção de sentimentos em mensagens baseado em técnicas de aprendizagem automática. A partir de um conjunto de mensagens, obtidas no *micro-blog Twitter*[1], o sistema analisa a estrutura morfológica das mensagens e determina o sentimento geral que as mesmas demonstram. As caraterísticas principais e inovadoras deste sistema são os valores da entropia das palavras presentes na mensagem. Com a implementação destas caraterísticas foi registada uma melhoria substancial na eficácia geral em relação ao resto do conjunto de caraterísticas das mensagens. O sistema apresentado representa uma parte do projeto desenvolvido para a tese de final de curso do Mestrado em Engenharia Informática.

A ordem do trabalho é a seguir descrita. Na secção 2 vão ser apresentados exemplos de trabalhos efetuados que refletem o estado da arte em relação à mineração de dados em contexto de análise de sentimentos. A descrição de todas as técnicas e abordagens utilizadas no sistema estão descritos na secção 3. Neste capítulo também será descrito o conjunto de dados utilizado bem como os passos de pré-processamento a que foi sujeito. Na secção 4 vão ser apresentados e analisados os resultados que foram obtidos pelo sistema. Finalmente, em 5 é apresentada a conclusão e abordadas sugestões de trabalho futuro com o objetivo de melhorar a eficácia do sistema.

[1] Mais informação em: http://www.twitter.com/

2 Trabalho relacionado

A resolução de trabalhos nesta matéria utilizam maioritariamente dois métodos para análise do texto: o método baseado em regras, como em [12], ou através de técnicas de aprendizagem automática supervisionadas.

Método de aprendizagem automática As técnicas de aprendizagem automática têm como objectivo estabelecer um modelo de classificação através de um conjunto de dados que represente a informação a ser alvo de análise.

Em [5] são analisados três dos métodos mais utilizados em aprendizagem automática: *Naive Bayes*, *Support Vector Machines* e *MaxEnt* (máxima entropia). As caraterísticas utilizadas foram baseadas em *unigram* (conjuntos únicos de palavras), com deteção da negação. Em *Naive Bayes* o melhor desempenho foi obtido com recurso a *unigrams* diferenciando as categorias das palavras através da junção da etiqueta POS às palavras, teve uma exactidão de 81.5%. Em *MaxEnt* a melhor exactidão obtida foi com recurso a *unigrams*, e obteve-se cerca 81%. No cômputo geral, tanto para *unigrams*, *bigrams*, ou *unigrams+POS* o método utilizando *SVM* registou quase sempre a melhor eficácia e através de *unigrams* registou uma exactidão de 83%. De facto, os métodos que retornam melhores resultados são normalmente com a utilização de *unigrams* [5][7].

Chauvalit et al [1] comparou a exactidão dos dois tipos de abordagem, o modelo baseado em regras e modelo de aprendizagem automática. Os resultados mostraram uma melhor eficácia utilizando técnicas de aprendizagem automática, conseguindo aproximadamente 85% contra 77% conseguidos através de métodos de classificação através de modelos baseados em regras. Turney [10] utilizando métodos de aprendizagem automática conseguiu cerca de 66%.

2.1 Trabalhos em RepLab 2013

Algumas das caraterísticas e abordagens implementadas no sistema apresentado têm como base ideias e sugestões presentes nos *RepLab*[2] 2012 e 2013.

O sistema *SZTE* [3], na sua versão número 8, foi aquele que obteve melhor eficácia geral com 69%, 48% de cobertura , 34% de abrangência e 38% de Medida F. No pré-processamento eram aplicados alguns dos processos mais utilizados como a redução das palavras à sua raíz (lematização), deteção e atribuição de valores de polaridade em *emoticons*, remoção de carateres estranhos e normalização de números, URL, *usertags* e sinais de pontuação. Era também aplicada a deteção de termos utilizados em redes sociais e substituído pela sua forma extensa, por exemplo, o termo "*LOL*"passa a "*laughing out loud*". Estes termos, muitas vezes, exprimem um sentimento e em dicionários comuns os mesmos não são referenciados.

O sistema *diue* [6], desenvolvido pelo Departamento de Informática da Universidade de Évora foi um dos representantes portugueses a participar na tarefa de classificação de polaridade. Muitas das suas ideias serviram como base para a elaboração do sistema desenvolvido. Tal como neste trabalho, para o processamento e análise da informação, foi utilizado o pacote de ferramentas NLTK para Python. O processamento da informação começa por separar as palavras através de pontuação ou de espaços em branco, em seguida aplica-se a lematização através de WordNet. Para a determinação de sentimento o *diue* utilizaram-se 3 léxicos de sentimento: AFINN, SentiWordNet e um léxico utilizado em [4], treinado a partir de críticas a produtos.

O sistema submetido foi treinado com recurso ao algoritmo de aprendizagem automática baseado em árvores de decisão presente no NLTK. O modelo de dados era composto por 18 caraterísticas. As mais relevantes relacionavam a posição da entidade (empresas da industria automóvel, entidades bancárias, escolas ou artistas) com as palavras que demonstravam sentimento, como por exemplo as presenças de negação e termos polarizados antes e após a entidade. O resultado do sistema submetido foi de 55% de eficácia com 25% de Medida F.

[2] http://www.limosine-project.eu/events/replab2013

O outro sistema português, POPSTAR [2], veio por intermédio do INESC-ID (Instituto de Engenharia de Sistemas e Computadores: Investigação e Desenvolvimento em Lisboa)[3]. Como os anteriores sistemas, recorre também a uma abordagem de aprendizagem automática com recurso a regressão logística.

O modelo de dados que serviu para treino e testes englobava a presença de palavras (*bag-of-words*) com pesos atribuídos através da abordagem Delta-TF.IDF. As palavras com maior valor de entropia eram excluídas pois não forneciam um valor discriminatório entre as classes. Para além disso foram utilizadas outras caraterísticas como valor geral de polaridade, número de palavras negativas e positivas, número de sinais de pontuação, *emoticons* ou palavras constituídas por maiúsculas. Uma caraterística que melhorou os resultados foi através da adição do título da página à mensagem sempre que no texto estivesse presente um URL. O POPSTAR obteve 64% de eficácia e 37% de Medida F.

3 Trabalho desenvolvido

Este sistema utiliza um conjunto de dados alvo de pré-processamento e representado através das caraterísticas seguidamente referidas. Os mesmos foram exclusivamente treinados e testados através da ferramenta NLTK[4] para Python. O NLTK é uma ferramenta para Python desenvolvida com o objetivo de facilitar e tornar o trabalho com linguagem natural mais eficaz. O seu pacote é composto por variados recursos lexicais e bibliotecas que permitem classificar e identificar as principais caraterísticas em textos.

3.1 Conjunto de dados

O corpus do RepLab 2013 consiste num conjunto de mensagens retiradas da rede social Twitter. O conteúdo destas mensagens poderá ser objetivo quando o seu teor não representa uma opinião, ou subjetivo, quando o conteúdo da mensagem indica um estado de espírito, uma opinião sobre um determinado produto ou entidade. Neste corpus estão representadas 61 entidades, desde empresas da industria automóvel, entidades bancárias, escolas ou artistas.

Após uma análise ao conteúdo do corpus foi detetado que havia algum ruído e que poderia distorcer o resultado do sistema. Posto isto foram filtradas as seguintes mensagens:

- Entradas em branco - mensagens que tinham sido apagadas pelos seus utilizadores;
- Mensagens repetidas - não seria eficiente estar a analisar duas vezes o mesmo conteúdo;
- Mensagens em diferentes línguas - apenas foram contabilizadas as mensagens escritas em inglês.

	Número de mensagens
Positivo	13462
Negativo	3164
Neutro	6282
Total	22908

Tabela 1. Número de mensagens por cada classe e o seu total.

Depois de feita uma filtragem das mensagens, a contabilização do conjunto de dados utilizado está descrito na Tabela 1. A partir da análise da tabela pode-se conferir que a quantidade de mensagens com conotação positiva é significativamente superior e as mensagens de classe negativa estão muito pouco representadas.

[3] http://www.inesc-id.pt/
[4] http://nltk.org/

3.2 Pré-processamento do texto

As mensagens por intermédio das redes sociais são muito descritas como difíceis de analisar não só pelo seu tamanho reduzido mas também pelo uso frequente de acrónimos, palavrões, *emoticons*, *Uniform Resource Locator* (URL), *hashtag*, etc. As mensagens originais foram alteradas de forma a que seja feita uma análise mais eficaz através dos seguintes métodos:

- Substituição de *emoticons* - Foi utilizado um dicionário de *emoticons* de forma a que fossem substituídos por "happy"ou "sad"cada vez que fosse encontrado um numa mensagem. Os *emoticons* foram categorizados por positivos ou negativos, por exemplo: ":)"seria substituído por "happy"e ":("por "sad".
- Remoção de URL - Foram identificados e removidos os endereços para páginas. Para efeitos de análise do texto esta informação não é relevante e poderia causar ruído. Os mesmos foram gravados à parte para posterior análise.
- Tratamento de *hashtags* - Estas são umas das caraterísticas mais comuns em mensagens via Twitter. Foi verificado durante este trabalho que frequentemente estas referências poderiam expressar um sentimento. Desta forma, foi retirado o carater # presente no início de cada uma e a *hashtag* tratada como se de uma palavra normal se tratasse.
- Remoção de *Usertags* - As *Usertags* são referências a outros utilizadores da rede social e por isso não acrescentam valor relevante ao conteúdo.
- Alteração de siglas e termos comuns da *Web* - Termos como *"LOL"* ou *"BRB"* são siglas já bastante conhecidas e utilizadas em qualquer mensagem via Web. Tendo como base uma ideia em [3] foi utilizado um dicionário de siglas e termos mais utilizados. Por exemplo os termos como *"LOL"* (*Laughing out loud*) ou "BRB"(*Be right back*) são substituídos pelos seus correspondentes significados por extenso. A lista foi retirada de *chatslang.com* [5].
- Remoção de caracteres estranhos - Caracteres como $, %, & ou são removidos do texto por não oferecerem nenhum tipo de informação relevante.
- Divisão de palavras pelas maiúsculas - É bastante comum em mensagens curtas se encontrarem palavras com algumas letras maiúsculas pelo meio. Essas palavras utilizam-se muito quando são referenciadas *hashtags* ou então apenas para poupar espaço. O sistema procura essas palavras e divide-as por intermédio dos seus caracteres maiúsculos, por exemplo, a palavra *"GreatService"* vai passar a ser duas palavras, *"Great Service"*.

3.3 Técnicas e caraterísticas utilizadas

Após a normalização das mensagens procedeu-se a um conjunto de técnicas de forma a ser possível retirar informação através da sua estrutura sintática, tal como a sua polaridade e caraterísticas principais.

Categorias gramaticais As categorias gramaticais, também conhecidas e doravante denominadas como *Part-Of-Speech* (POS), servem para identificar a categoria de cada palavra com base na sua definição e contexto dentro de uma frase. A relação de cada palavra com as adjacentes são determinantes para atribuir corretamente a categoria a cada uma das palavras.

A abordagem utilizada para a classificação de palavras é o módulo `pos_tag` do NLTK. Este é um classificador treinado através do algoritmo *Maximum Entropy* com recurso ao corpus Treebank [6]. Este classificador está treinado para textos em Inglês.

Na maior parte das vezes são os verbos, adjetivos e advérbios que são representativos do sentimento demonstrado. As palavras que não sejam catalogadas dentro nestas 3 categorias serão descartadas, uma vez que na maior parte das vezes não representam informação quanto ao sentimento.

Tomando como exemplo as duas frases [5], *"This is a love story!"* e *"I love this story."*. A primeira frase é neutra em relação a sentimentos, pois neste caso a palavra "*love*"é um nome e não está a exprimir nenhum sentimento ou opinião sobre determinada matéria.

[5] http://www.chatslang.com/terms/common
[6] http://www.cis.upenn.edu/~treebank/

Lematização de palavras A técnica seguinte à deteção de categoria é a lematização de palavras, seguindo a metodologia do trabalho [11]. Para isso, foi utilizado o recurso WordNet. O processo é reduzir cada palavra à sua raíz o que se traduz numa diminuição de palavras diferentes mas com significados iguais, por exemplo, verbos em diferentes modo. Esta também é uma forma de corrigir alguns erros gramaticais.

3.4 Classificação do sentimento através de aprendizagem automática

O modelo de dados é composto por 24 caraterísticas que representam as diferenças nas mensagens presentes no conjunto de dados.

- Percentagem de palavras neutras
- Percentagem de palavras com polaridade positiva
- Percentagem de palavras com polaridade negativa
- Valor máximo de polaridade negativa
- Valor máximo de polaridade positiva
- Presença de palavra com polaridade positiva
- Presença de palavra com polaridade negativa
- Quantidade de palavras com polaridade positiva
- Quantidade de palavras com polaridade negativa
- Quantidade de palavras neutro
- Soma das polaridades de todas as palavras
- Presença de intensificadores positivos
- Presença de intensificadores negativos
- Quantidade de pontuação na frase
- Presença de ponto de exclamação
- Presença de ponto de interrogação
- Presença de negação
- Presença de entidades
- Presença de palavras com caracteres maiúsculos
- Presença de URL
- Polaridade do título do URL
- Valor entropia de palavras com sentimento positivo
- Valor entropia de palavras com sentimento negativo
- Valor entropia de palavras com sentimento neutro

Para o cálculo da polaridade (positiva, negativa ou neutra) de uma mensagem é utilizado um dicionário de palavras com polaridade. O dicionário AFINN é composto por 2477 palavras manualmente anotadas de -5 a 5 conforme o seu valor de negatividade ou positividade respetivamente.

Os **intensificadores** são palavras que podem significar um reforço a uma palavra com sentimento, quer positivo ou negativo. Exemplo dessas palavras são "mais"ou "pouco". A presença destas palavras está relacionada com frases com teor subjetivo.

As caraterísticas que identificam as pontuações estão implementadas para identificar variadas situações. A caraterística *quantidade de pontuação* adquire um valor numérico que representa a quantidade de sinais pontuação, por exemplo: `!"#\$%&'()*+,-./:;<=>?@[\]^_‘{\}~`.

Palavras como "não"ou "nunca"são palavras que demonstram uma negação e que podem inverter tanto a polaridade de um adjectivo ou verbo com sentimento conseguinte ou até o sentimento geral de uma frase. Foi utilizado uma base de dados com palavras como "not", "never"ou "neither"e a presença de uma destas palavras é identificada. O método adotado foi retirar ou adicionar 2 valores sempre que o sentimento de a palavra a seguir à negação fosse maior do que 2 ou menor que -2 e inverter caso fosse compreendida entre -2 a 2 [9].

O método de **deteção de entidades** utilizado é o método `NE_Chunk` do NLTK, treinado com o algoritmo de aprendizagem automática *Maximum Entropy* através do corpus *Automatic Content*

Extraction (ACE) [7] desenvolvido pelo *Linguistic Data Consortium*[8]. A deteção de uma entidade na mensagem pode significar a subjetividade da mesma.

As mensagens têm a particularidade de conter ligações *URL* para outras páginas. Estas ligações também podem fornecer informação dado que o conteúdo das mesmas está relacionado com o conteúdo da própria mensagem. A abordagem seguida neste sistema foi calcular o **valor da polaridade do título** dessas mesmas páginas, como em [2].

O conceito de **entropia** pode ter vários significados consoante as suas utilizações. Neste sistema, a entropia é utilizada para medir a distribuição de cada palavra em relação à sua presença nas várias classes de sentimentos. Por exemplo, a palavra "*happy*"tem menor entropia do que a palavra "*the*"pois a sua utilização, normalmente, é feita em frases onde o sentimento global é positivo, ao contrário da última em que pode aparecer diversas vezes em frases nas diferentes classes.

A utilização da entropia neste sistema teve como base o trabalho desenvolvido em [2] e foi calculado utilizando a definição feita por *Shannon* em 1948 [8]. A função de cálculo da entropia utilizada é:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \tag{1}$$

[8]

O X é a palavra objeto de cálculo; n representa as 3 classes onde a palavra pode estar presente (positiva, negativa e neutra); e $p(x_i)$ a probabilidade de a palavra surgir numa determinada frase dentro da classe.

O resultado da entropia não é indicativo do grau de sentimento de uma palavra, apenas é dada uma indicação da distribuição da sua ocorrência ao nível das diferentes classes. Dando como exemplo os valores H(one) = 0.114 bits e H(nicely) = 0.0007 bits, a palavra "*one*"está presente mais vezes e provavelmente em mais classes e por isso menos relevante em termos de informação que a palavra "*nicely*".

Considerou-se a entropia apenas de palavras com valor de entropia menor que 0.004 bits. Este valor foi escolhido após uma análise dos valores em palavras mais frequentes. Com este limite têm-se a certeza que as palavras que aparecem com maior frequência são excluídas.

Para cada uma das palavras que estão presentes na mensagem o sistema vai calcular os valores das caraterísticas respeitantes à entropia. Se o valor de entropia da palavra for menor que 0.004 bits é feita uma pesquisa no conjunto de palavras que ocorrem em mensagens das classes Positivo, Negativo ou Neutro, por esta ordem. Caso ela esteja presente numa frase da classe Positivo é adicionado o respetivo valor de entropia à caraterística *"positive_entropy_value"* e o sistema passa automaticamente para a próxima palavra da frase. Se não, estiver presente nas palavras que aparecem na classe das frases com polaridade positiva é feita uma pesquisa do valor de entropia nas palavras de classe Negativo e em último caso, na classe Neutro.

4 Resultados

A Tabela 2 apresenta o resultado obtido. Estes valores dizem respeito à classificação por intermédio do algoritmo de aprendizagem automática Naive Bayes com as caraterísticas e técnicas anteriormente referenciadas. Os resultados apresentados foram obtidos dividindo o mesmo conjunto de dados em dois subconjuntos de quantidades iguais, um para o treino e o outro para os testes.

Estes valores são bastante satisfatórios. Uma taxa de acerto elevada em conjunto com os valores altos de precisão, cobertura e Medida F. A classe Positivo, regra geral, foi a que teve melhores resultados nas 3 medidas de desempenho, esse resultado pode estar relacionado com o facto de haverem mais amostras para essa classe. A classe Neutro teve o melhor resultado de todos em

[7] http://catalog.ldc.upenn.edu/LDC2005T09
[8] https://www.ldc.upenn.edu/

	Precisão	Cobertura	Medida F	Taxa de Acerto
Positivo	0.8039	0.9131	0.8550	-
Negativo	0.6203	0.6227	0.6215	-
Neutro	0.872	0.6116	0.719	-
Total	-	-	-	0.7918

Tabela 2. Resultado do sistema com seleção de polaridade com base em verbos, adjetivos e advérbios.

termos de precisão dos resultados. Os 79,2% obtidos de taxa de acerto demonstram um resultado ao nível ou até melhor em comparação com alguns dos sistemas apresentados no estado da arte, que foram apresentados no RepLab e utilizam o mesmo tipo de dados.

5 Conclusões

Através do resultado obtido pode-se concluir que esta abordagem e técnicas utilizadas neste sistema garantem uma boa fiabilidade em detetar o sentimento geral de uma frase nas 3 classes distintas. O cálculo do valor de entropia de uma palavra relativamente à sua classe demonstra ser uma caraterística com bons resultados na distinção entre as várias classes.

Apesar de este sistema ter sido apenas testado em mensagens provenientes da rede social Twitter, estas mensagens são geralmente difíceis de analisar dada a sua complexidade. A pouca extensão, a diversidade de carateres especiais e referências URL, *hashtags*, entre outras, exigem que seja feito um conjunto de técnicas de processamento do texto. Dada esta complexidade, o sistema desenvolvido adquiriu uma maior versatilidade, prevendo-se que seja capaz de garantir bons resultados a partir de um outro tipo de conjunto de mensagens.

Com a capacidade de detetar entidades em mensagens, futuramente a este sistema, poderá ser implementada a funcionalidade de relacionar o sentimento geral da mensagem com uma ou várias entidades presentes. Apesar disso, este é um sistema que ainda não está totalmente finalizado e que poderá ser melhorado com a introdução de treinos e testes a partir de um diferente conjunto de dados.

Referências

1. Chaovalit, P., Zhou, L.: Movie review mining: a comparison between supervised and unsupervised classification approaches. In: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 4 - Volume 04. HICSS '05, IEEE Computer Society (2005)
2. Filgueiras, J., Amir, S.: Popstar at replab 2013: Polarity for reputation classification. In: To appear in: Fourth International Conference of the CLEF initiative. CLEF 2013 (2013), http://www.clef-initiative.eu/documents/71612/1ee57b21-3296-4371-96cd-2dae6367d2ff
3. Hangya, V., Farkas, R.: Filtering and polarity detection for reputation management on tweets. In: To appear in: Fourth International Conference of the CLEF initiative. CLEF 2013 (2013), http://www.clef-initiative.eu/documents/71612/51490ac1-b1fa-4ea2-a520-6b52ef98e862
4. Liu, B., Hu, M., Cheng, J.: Opinion observer: Analyzing and comparing opinions on the web. In: Proceedings of the 14th International Conference on World Wide Web. pp. 342–351. WWW '05, ACM, New York, NY, USA (2005), http://doi.acm.org/10.1145/1060745.1060797
5. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques (2002), http://dx.doi.org/10.3115/1118693.1118704
6. Saias, J.: In search of reputation assessment: experiences with polarity classification in replab 2013. In: To appear in: Fourth International Conference of the CLEF initiative. CLEF 2013 (2013), http://www.clef-initiative.eu/documents/71612/10fcd949-e5f0-4f00-8e01-cbd2a213e147
7. Salvetti, F., Reichenbach, C., Lewis, S.: Opinion Polarity Identification of Movie Reviews (2006)
8. Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal 27, 379–423, 623–656 (July, October 1948), http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf

9. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., Stede, M.: Lexicon-based methods for sentiment analysis. Comput. Linguist.
10. Turney, P.D.: Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. CoRR cs.LG/0212032 (2002)
11. Villena-Román, J., Lana-Serrano, S., Moreno, C., García-Morera, J., Cristóbal, J.C.G.: Daedalus at replab 2012: Polarity classification and filtering on twitter data. In: Forner, P., Karlgren, J., Womser-Hacker, C. (eds.) CLEF (Online Working Notes/Labs/Workshop) (2012), http://dblp.uni-trier.de/db/conf/clef/clef2012w.html#Villena-RomanLMGC12
12. Yang, C., Bhattacharya, S., Srinivasan, P.: Lexical and machine learning approaches toward online reputation management. In: Forner, P., Karlgren, J., Womser-Hacker, C. (eds.) CLEF (Online Working Notes/Labs/Workshop) (2012), http://dblp.uni-trier.de/db/conf/clef/clef2012w.html#YangBS12

Constraint programming for parallel systems: An overview

Pedro Roque

Universidade de Évora, Portugal
d11735@alunos.uevora.pt

Abstract. The increasing multi-core architectures make current parallel systems capable of much more processing power then any single-core system. The challenge is how to develop software capable of harnessing that power in such a simple and fast manner as if it were developed for single-core architectures.
One of the most studied areas for multi-core architectures programming is the development of constraint based systems capable of solving constraint satisfaction problems.
This article describes the main features, the techniques and the programming languages applied to the most currently implemented systems for constraint satisfaction problems solving in parallel environments.

Keywords: Parallel systems, constraint programming, Constraint Satisfaction Problems

1 Introduction

Currently, almost every new personal computer, laptop and even some smartphones use multi-core architectures. The problem is that some of the software currently in use is not able to gain performance in those systems when compared to single-core architectures.

This problem is even bigger if we consider the level of parallelism and processing power of some systems as the current fastest supercomputer in world Tianhe-2, which comprises 3,120,000 cores and 1,375 TiB of memory [20].

Some researchers are studying and developing software systems capable of solving complicated problems through the increased processing power of parallel computer architectures. Much research in this area is focused in solving Constraint Satisfaction Problems (CSP).

Definition 1. *A CSP can be defined as a triple* $P =< X, D, C >$*, where:*

- $X =< x_1, x_2, ..., x_n >$ *is a n-tuple set of variables;*
- $D =< D_1, D_2, ..., D_n >$ *is a n-tuple set of finite domain values, where* D_i *is the domain of the variable* x_i*;*
- $C =< C_1, C_2, ..., C_m >$ *is a set of relations between the variables in* X*, called the constraints.*

A solution for the CSP P is a n-tuple $A =< a_1, a_2, ..., a_n >$ where $a_i \in D_i$ and every C_j is satisfied.

The most current researches in CSP parallel solving aimed at testing new implementations in some academic problems like the n-Queens[1] and the Sudoku [2]. Other current researches in this area were applied to solving scheduling problems based on precedence constraints between the jobs that define the scheduling problem. As these two branches of CSP solving are currently the most researched, these will be the main topics of this article. The used techniques, algorithms and some implementation specifications will be described in the following sections.

In the next section some basic notions about parallel search problems are described and is presented a parallel search with parallel consistence system. In section 3 is described a work developed on how to adapt the parallel search while it is running. Section 4 introduces some works done about work stealing in parallel search and in section 5 are described some systems developed for scheduling problems solving in parallel environments. Finally, in section 6 are presented the conclusions retrieved while developing this article.

2 Parallel search with parallel consistency

Parallel search and parallel consistency are two basic notions of problems solving in parallel systems. Parallel search consists on searching for the problem solution using multiple parallel instances of solvers, and parallel consistency is used to maintain a coherent search between the multiple solvers.

In [12] is stated that the CSPs are usually solved through the combination of backtracking search with consistency checking that removes inconsistent values:

- In backtracking, each search tree node corresponds to a value assigned to a variable. That value is chosen from the domain of each variable. When one assignment causes a restriction violation, that value is removed from the domain set of its respective variable, and that branch of the search tree is abandoned;
- In consistency check, when a value assignment violates any restriction, the branches from the search tree whose root node correspond to that assignment are removed from the search tree.

In [12] is shown that the benefit of using constraint programming in parallel search is affected by the position of the solution. Two examples of this occurrence are represented in figure 1. In these examples the problems A and B are divided by three processors (P1, P2 and P3), but the solution of each problem is found in different locations of the search tree:

[1] The n-Queens problem consists on placing n-queens on a $n \times n$ chessboard such that no two queens are able to attack each other.

[2] The Sudoku problem consists on filling a $n \times n$ square grid with integer numbers. The grid is also divided with n squares. The numbers can't be repeated inside each of the n squares, neither on each main square horizontal or vertical lines.

- In problem A, the solution is located at the bottom left of the search tree. This means that the work done by processors P2 and P3 was unnecessary, resulting only in increased communications. In this case, it would be more fruitful to use P2 and P3 to maintain parallel consistency;
- In problem B, the solution is located at the portion of the search tree processed by P3, which means that all the work done by P1 and P2 was unnecessary, also resulting only in increased communications. In this case, parallel search would mean to decrease the total amount of explored nodes for about two-thirds, but parallel consistence would also speed up the process.

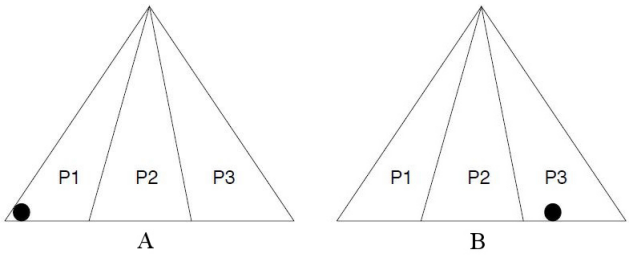


Fig. 1. Two examples of the possible solution position in search trees [12].

In [12] is also concluded that the level of performance gained by parallel search and parallel consistence directly depends on the type of problem to solve:

- If a problem is highly constrained, it will be difficult to gain any performance by adding parallel search due to the increased communications caused by parallel consistence;
- By the contrary, if a problem has only a few constraints, it will have to many inconsistent branches for an effective parallel consistence.

Rolf et al in [12] used JaCoP solver (Java Constraint Solver [14]) for solving Sudoku and n-Queens problems using parallel search with parallel consistence. According to Rolf et al in [12], JaCoP is fully implemented in Java allowing to easily add multithreading and distribution over network sockets.

According to Rolf et al in [12], the JaCoP execution is split in three phases. This phases are represented in figure 2, and can be described as:

- Initialization - Prepare the solvers to receive work;
- Search - Data-parallel depth-first search;
- Termination - Detects when the search has finished.

Rolf et al in [11] states that to apply JaCoP to a distributed cluster, the only change that has to be done is to replace a single reference to the search object type on the original constraint code. After that change is done, JaCoP allows

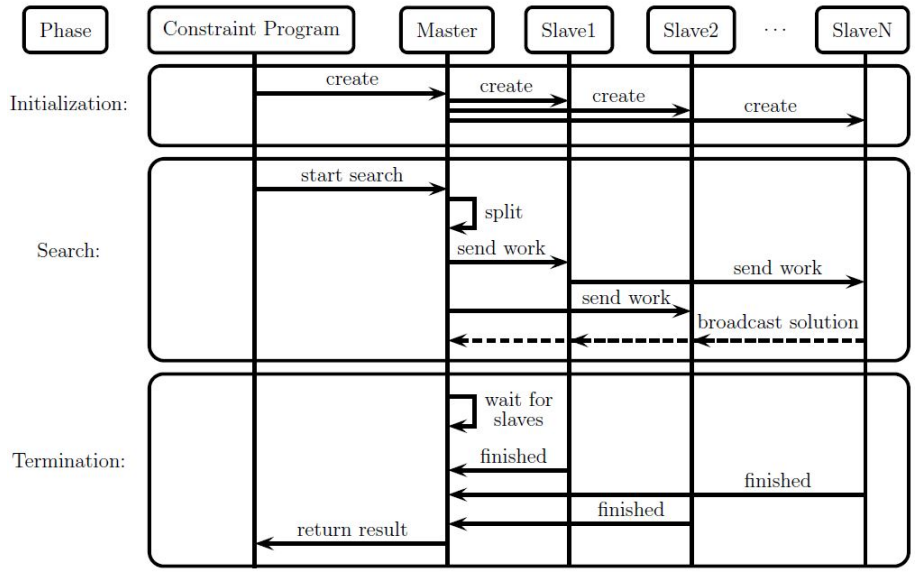


Fig. 2. Model of parallel constraint solving in JaCoP [11].

the selection of the load-balancing methods and communication models in order to adapt the system to a particular CSP.

From the results of the CSP solved with JaCoP, Rolf et al in [12] concluded that to make parallel consistence more independent of the problem to solve, the data should be shared through global constraints as a way to reduce the need for synchronization.

In [12] is stated that, due to time consumption, the solutions are normally not complete, which means that the domain of some variables may contain values that are only locally consistent, not being part of the solution.

Some authors claim that in addition to the parallel consistency, another dynamic knowledge can be used to improve performance during parallel search. The following section describes a system that adapts the parallel search while it is running.

3 Parallel adaptive search

One possible way to improve the performance of the search is to adapt it to the incremental knowledge about the search space while the search is running. Caniou et al in [2] developed a parallel Adaptive Search algorithm for solving CSP. According to Caniou et al in [2], the generic Adaptive Search algorithm is a domain-independent local search method for solving CSP, which the main iterative steps are:

1. To each constraint is defined a heuristic function (also called "error function") that indicates how much the respective constraint is violated;
2. To each variable is generated an error value that corresponds to the sum of the errors of the constraints (how much the constraint is violated) on which it appears;
3. The variable with the highest error (called the "culprit") is assigned with the value from its domain that resulted in the littlest error with the next configuration (all variables assigned with one value from the respective domain).

This algorithm uses three preventive methods to avoid entering in an infinite loop:

- Short-term memory - The configurations known as leading to a loop are identified in a list (known as the Tabu list) for a defined number of iterations. The configurations marked as Tabu are ignored on the above step 3;
- Reset - When the search stagnate around local minimums, some randomly selected variables are assigned with also randomly selected values from their domain;
- Restart - When the search surpasses a predefined number of iterations or an amount of variables marked as Tabu, it may be restarted from scratch.

Caniou et al in [2] used the freeware C-based framework library of Adaptive Search algorithm available for download at [4], combined with the OpenMPI (Open Message Passing Interface) for parallelization. OpenMPI allows a simple message passing in networked environments providing the next main features [9]:

- Support for network heterogeneity, allowing the use of multiple types of devices, operating systems and/or protocols;
- Thread safety and concurrency for safety and consistent resource sharing between threads;
- Network and process fault tolerance to avoid most of the system failures;
- Dynamic process spawning allowing the addition of processes to a running job.

For the parallelization, every available core receives a sequential fork of the Adaptive Search method, and runs it for a predefined number of iterations. After that number of iterations a test is made to check if there is any message indicating that a process has found a solution. If a solution has been found, the execution is terminated. If during that amount of iterations, more that one process has found a solution, the fastest one is selected by the process 0, which has received their execution time.

Caniou et al in [2] used three benchmarks, namely the All Interval Series problem, the Perfect Square placement problem and the Magic Square problem. The All Interval Series problem can be explained has defining an $\mathbb{Z}_n$ vector $s = (s_1, ..., s_m)$, such that s is a permutation of $\mathbb{Z}_n = \{0, 1, ..., m-1\}$ and the interval vector $v = (|s_2 - s_1|, |s_3 - s_2|, ..., |s_m - s_{m-1}|)$ is a permutation of

$\mathbb{Z}_n - \{0\} = \{1, 2, ..., m-1\}$ [1]. The Perfect Square placement problem consists on placing n squares of different sizes inside a master square [8]. The Magic Square problem consists on filling a square grid with distinct integers, such that the sum of each horizontal, vertical and diagonal line of integers is equal [19].

Caniou et al in [2] achieved speedups of about 30 to 50 when using 64 and 256 cores respectively, and also states that the speedups are more relevant for bigger benchmarks.

Although sending a sequential fork of the search space to each parallel solver is somewhat an intuitive method of dividing the search space, one of the most balanced work distribution method in parallel systems is work stealing. The following section introduces this method and some works that implemented it.

4 Work stealing

Work stealing is a method to distribute the search space among the multiple instances of search engines (also called agents or workers) on parallel systems.

According to Gent et al in [5], multi-agent search is a promising approach to parallelism usage in CSP solving. This is done by using multiple agents to solve the same problem. Any of the agents is capable of solving the problem independently, or a part of it. Also, the agents may be different and may communicate between them.

An example of this approach was developed in [10], which implemented a distributed parallel constraint solver, called the Parallel Complete Constraint Solver (PaCCS). PaCCS is a constraint solver based on splitting the search space through multiple agents (workers) that when complete its search space, steals another from its co-workers. Each worker was built as a search engine that interleaves rule-based propagation and search.

The process of work stealing is transparent to the co-workers from which the work is stolen. According to Vasco in [10], work stealing is a highly parallelizable load-balancing technique that enables the full use of the power of multiprocessor computers. Vasco in [10] accomplished this by allowing workers to steal work from his teammates, or from other teams, such that no team keeps without work if there is any that can be spared.

Vasco in [10] implemented PaCCS for Unix, in C programming language with the objective of providing a back end to a higher level language allowing constraint modelling constructs and the transparent usage of the multiprocessing hardware available. PaCCS use POSIX threads for an easier memory sharing and the MPI standard for distribute the search space through the workers and for transmitting other required messages.

According to Vasco in [10], each team corresponds to a MPI process, and each worker and controller is a POSIX thread of that process.

The internal representation of the CSP variables domains is named as the domain store. This was implemented as an array of domains in a contiguous region of memory. Each variable domain was implemented with a fixed-size bitmap, allowing the use of two more fields containing the maximum and minimum value of

the respective variable domain. Next to each store is included information about the variable which domain was divided yielding this store. The CSP variables and constraints are also stored in shared memory.

According to Vasco in [10], the domain store contains all the dynamic information needed to define a search space. As such, when a worker steals work from another worker (teammate or not), one store is the stolen unit. Each worker maintains a pool of stores arranged as an array of stores indexed accordingly to its age and may split his current search space in multiple stores that may be stolen by other workers from the same team or from a different one.

In figure 3 is represented the main constituents of PaCCS architecture.

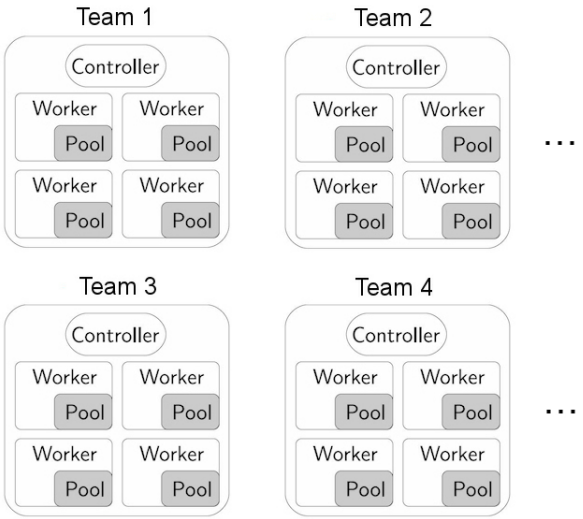


Fig. 3. Architecture of PaCCS [10].

PaCCS is able to run on multiprocessing systems constituted by multiprocessors, networked computers or both. To achieve this level of distribution, PaCCS is based on a two levels architecture:

- Lower level - It represents the teams. Each team is constituted by a group of tightly coupled workers that share resources;
- Higher level - It consists on the coordination of the teams for solving the CSP.

Vasco in [10] tested his implementation by solving n-Queens, Langford Numbers, Golomb Ruler and Quadratic Assignment problems. The Langford Numbers problem consists in defining m sets of integers from 1 to n, such that between two consecutive occurrences of integer i exist exactly i other numbers [10]. The Golomb Ruler problem consists in defining a set of n marks on a integers imaginary ruler, such that the distance between each pair of marks is unique between

all pairs. Also the first and last mark define the size of the ruler, and the ruler should have the minimum size possible [10]. The Quadratic Assignment Problem consists on assigning a set of n facilities to a set of n locations, while minimizing the sum of the distances between locations multiplied by the flow or weight of the facilities [15].

The results obtained by Vasco in [10] showed that PaCCS is a very scalable parallel constraint solver which achieved an almost linear performance for all the tested problems.

Chu et al in [3] developed an adaptive work stealing algorithm that automatically execute different work stealing techniques, according to the estimated solution density (estimated probability of containing a solution) of each sub-tree.

According to Chu et al in [3], the efficiency of the used branching heuristic is directly related with the efficiency of the achieved load balancing. For developing an optimal branching heuristic Chu et al in [3] used a sub-tree solution density estimation algorithm that defines the branching confidence of each node. The confidence of a node is the estimated ratio of the solution density in the sub-trees derived from that node.

For estimating the solution density, Chu et al in [3] attended to the following properties:

- The solution density between nearby sub-threes is highly correlated because many of the constraints are shared between them;
- As the number of nodes from a sub-tree that aren't the solution increases, the solution density of that sub-tree and the nearby sub-trees decreases.

Following that properties, Chu et al in [3] formulated the expressions 1, 2 and 3 to calculate each node solution density:

- For uncorrelated sub-trees:

$$S = \sum_{i=j+1}^{n} \frac{A_i}{n-j} \tag{1}$$

- For correlated sub-trees:

$$S = \frac{\sum_{i=1}^{n} A_i k_i}{\sum_{i=1}^{n} k_i} \tag{2}$$

- For updating solution densities as new nodes are explored:

$$S = \frac{1}{n} \sum_{i=1}^{n} A_i \; and \; A_i = 0 \; for \; 1 \leq i \leq j \tag{3}$$

Where:

- S represents the node solution density estimation;
- n represents the search tree number of nodes;
- A_i represents the solution density of the sub-tree root node i;
- j represents the child sub-tree;

- k_i represents the child sub-tree i number of nodes.

After formulating the above solution density equations, for the cases when a sub-tree is searched and failed to find a solution, Chu et al in [3] formulated the confidence update equation as:

$$\bar{r}'_i = \frac{\bar{r}_i - \prod_{k=1}^{i} \bar{r}_i}{1 - \prod_{k=1}^{i} \bar{r}_i} \tag{4}$$

Where:

- r_i represents the confidence value of the node i levels above the searched sub-tree before being updated with the search results;
- r'_i represents the confidence value of the node i levels above the searched sub-tree after being updated with the search results;
- i represents the number of nodes above the root sub-tree that has been searched;
- $\bar{r}_i = r_i$ and $\bar{r}'_i = r'_i$ if the searched sub-tree belongs to the left branch of the node i;
- $\bar{r}_i = 1 - r_i$ and $\bar{r}'_i = 1 - r'_i$ if the searched sub-tree belongs to the right branch of the node i.

Using the formulas 1, 2, 3 e 4, Chu et al in [3] manage to assign a confidence value to each node while searching for a solution. As at the beginning of a search there are no confidence level for any node, the initial confidence value, ideally, could be developed by the problem modeler, as an expert on the problem to solve. If that's not possible, those values could just be equal in every node, being updated as the search continues.

With a confidence value assigned to every node, Chu et al in [3] managed to develop a confidence-based search algorithm with the following features:

- After a sub-tree is fully explored, the confidence value of all the above nodes is updated through the formula 3;
- The number of threads exploring each branch is updated as search advances;
- When a worker finish his sub-tree and needs a new one, he "steals" it by following the next rules:
  - Always starts searching for an unexplored node on the root of the search tree;
  - Search for an unexplored node through the left branch of the current explored node if $|\frac{a+1}{a+b+1} - r| \leq |\frac{a}{a+b+1} - r|$, otherwise search through right (a, and b are the number of working threads on the left branch and right branch, respectively, and r is the confidence value of the current explored node);
  - If after going down the tree for a certain number of nodes (depth), no unexplored node is found, it will steal the first unexplored node above that depth (even if it has fewer confidence);
  - The above mentioned allowed depth value is dynamically changed to maintain a minimum sub-tree size, so that work stealing doesn't occur more often then a predefined threshold to limit communication costs;

- A master coordinates all the workers;
- A worker works on a sub-tree for a predefined restart time and after that time passes, it will return the results of his exploration to the master, and steals a new sub-tree.

Chu et al in [3] implemented his confidence-based work stealing algorithm using Gecode [13]. Gecode is a free software library developed to simplify the implementation of constraint-based systems and applications. It is implemented in C++ and has bindings for Python, Prolog, Ruby, Java and other programming languages.

Chu et al in [3] tested their system with the Traveling Salesman, the Golomb Ruler, the n-Queens, the Knights and the Squaring the Square problems. The Traveling Salesman Problem consists in defining the shortest path that visits a group of cities. Each city can only be visited once, except for the origin one, that must also be the one to return at the end [21]. The Knights problem consists on moving a Knight on a chessboard such that he visits every square only once [17]. The Squaring the Square problem consists on dividing an integral square in n smaller integer squares, all with different sizes [16].

Those tests where run using eight threads on a Dell PowerEdge 6850 with four 3.0 GHz Dual Core Pro 7120 CPUs and 32 Gb of memory. The used reset time was 5 s and the minimum allowed time between work stealing for each thread was 0.5 s.

According to Chu et al in [3], even using biased initial confidence values was sufficient to obtain an almost linear speedup. On the contrary, if the initial confidence values were specifically assigned to try achieving the best results, the outcome was worse. Chu et al in [3] states that the developed system was capable of achieving a speedup of about 7 using the 8 threads for all the tested problems, and also states that the communication costs where almost imperceptible.

Apart from solving academic problems like the Golomb Ruler and the Knights, other researchers branched their CSP solving to scheduling problems. The following section presents some systems developed for scheduling problems solving in parallel environments.

5 Scheduling problems solving

Scheduling problems are among the most complex ones that are solved by parallel systems implementations.

Kim in [7] developed a Linear Programming (LP) based heuristic specific for list scheduling with s-precedence[3] constraints on uniform parallel machines. Linear Programming or linear optimization. Is a method to obtain the best results in a mathematical model, which requirements are represented by linear relations [18].

[3] A s-precedence constraint between two jobs i and j, means that j can only start after i starts.

The heuristic intended to schedule s-precedence constrained jobs on m uniform parallel machines in order to reduce the processing time required. This heuristic was based on the next steps:

1. A cutting plane method was used to solve the LP problem - Kim in [7] used an iterative approach to find new valid inequalities[4] of constraints, which allow to refine the set of constraints. For generating the new inequalities, Kim in [7] developed a separation algorithm (also called as separation problem), that allow to solve the LP problem in polynomial time;
2. Assigning the unscheduled job with the smallest start time to the machine on which it will finish first. The unscheduled job with the smallest start time (s_j^{LP}) was found by the formula:

   $s_j^{LP} = \frac{C_j^{LP} - p_j}{max_{i \in M} S_i}$, where:
   - C_j^{LP} is the optimal solution of the LP problem, found by the implemented heuristic;
   - p_j is the processing time of job j;
   - $max_{i \in M} S_i$ represents the fastest machine, where i is the machine from group M, and S_i is the speed of that machine.
3. The step 2 was repeated until all jobs are scheduled.

According to Kim in [7], the overall complexity of the developed heuristic is $O(n^5 L)$, where L is the number of bits required for the inputs of the LP problem. In terms of number of machines, jobs and the density of the constraints, according to Kim in [7], the developed heuristic performs better as:

- The number of jobs per machines (n/m) increases;
- The density of s-precedence constraints increases;
- The range of distribution generating machine speeds (S_i) decreases;
- The range of distribution generating job weights (w_j) decreases.

The heuristic was implemented in C++ and the LP problem in the heuristic was solved by the commercial solver IBM ILOG CPLEX. CPLEX is a software developed by IBM, capable of modeling and solving business issues mathematically. It provides interfaces for C, C++, C#, Java, Visual Basic, python, FORTRAN and MATLAB [6].

Xie et al in [22] developed a scheduling constraint based solver for massive parallel systems, as an IBM BlueGene/L and BlueGene/P, with 65,536 and 1,048,576 processors, respectively. The parallel solver developed by Xie et al in [22] was based on the C++ constraint programming based Watson Scheduling Library, developed at IBM.

Xie et al in [22] implemented load balancing by dynamically partitioning the search space among the available processors. This was done by dividing processors into masters and workers. The master processors has a global view of the search tree and coordinates the workers by dividing the search tree between

[4] An inequality is a relation between two different elements.

them and keeping track of the branches that have been explored or are yet to be explored. Each worker has only one master, but each master is able to coordinate multiple workers.

The workers request a sub-tree from their master and explores it. Each ramification on the search tree corresponds to a constraint added to the tree. So, a sub-tree (or sub-problem) is passed to the workers as a set of constraints, using the MPI standard. Each worker can receive a sub-problem in the beginning of problem solving, or after exhausting the search on the previous sub-tree.

Each master keep a representation, named the Job Tree, of the parts of the search tree that has been explored, are being explored or are unexplored by the workers. The unexplored sub-trees are the ones that the masters send to the workers that claim a new sub-problem.

When the solving process starts the search tree is divided between the masters in a static way. For load balancing purposes, a number of possible branchings is evaluated before the search tree is divided. After that division each master creates a job tree by exploring some part of his sub tree. Then, if no solution was found, starts dispatching peaces of his sub-tree to the workers that request them. When a worker, while exploring his sub-tree, states that it had become too large (more nodes then the predefined threshold), he sends a peace of his sub-tree to his master, and keeps working on the part he didn't send. His master expands his Job Tree with the new explored nodes.

After testing their implementation, Xie et al in [22] stated that they almost achieved linear scaling with one master. But with multiple masters the results where far from linear scaling. Xie et al in [22] states that for achieving that kind of performance with multiple masters, a technique for dynamically allocating sub-problems between masters must be developed.

In the following section are presented the conclusions retrieved while developing this article.

6 Conclusion

According to the works reviewed in this article, there is still some work to be done as a way to achieve the same programming simplicity for multi-core systems as for single-core. Nevertheless, there are some works that already simplify the scalability problem, like the PaCCS developed in [10].

Nevertheless, most of the systems presented in this article achieved almost linear speedup when solving some CSP problems.

The biggest challenge is to develop a parallel system capable of achieving the optimal speedup for any type of problem. In this matter some works like [2] and [3] try to adapt the search for a solution while searching, allowing the dynamic adaptation to different problems.

As seen in [3] and [10], work stealing is a work distribution technique that allows a very good load-balancing, but the level of needed communications must be very well attended during the system implementation to ensure a minimum increase on communication costs, and consequent decrease in performance.

Although personal computers and laptops are constantly being updated with more parallel processing power, and above all, despite the increasing number of supercomputers with thousands or even millions of cores, parallel programming still has much room for improvement.

References

1. Alsinet, T., Béjar, R., Cabiscol, A., Fernàndez, C., Manyà, F.: Minimal and redundant sat encodings for the all-interval-series problem. In: Escrig, M., Toledo, F., Golobardes, E. (eds.) Topics in Artificial Intelligence, Lecture Notes in Computer Science, vol. 2504, pp. 139–144. Springer Berlin Heidelberg (2002), http://dx.doi.org/10.1007/3-540-36079-4_12
2. Caniou, Y., Codognet, P., Diaz, D., Abreu, S.: Experiments in parallel constraint-based local search. In: The 11th European Conference on Evolutionary Computation and Metaheuristics in Combinatorial Optimization (EvoCOP 2011). pp. 96–107. LNCS, Torino, Italy (April 2011), http://cri-dist.univ-paris1.fr/diaz/publications/ADAPTIVE/evocop11.pdf
3. Chu, G., Schulte, C., Stuckey, P.J.: Confidence-based work stealing in parallel constraint programming. In: Gent, I.P. (ed.) CP. Lecture Notes in Computer Science, vol. 5732, pp. 226–241. Springer (2009), http://ww2.cs.mu.oz.au/~pjs/papers/cp09-co.pdf
4. Diaz, D., Codognet, P., Abreu, S.: Adaptive search distribution (Jun 2012), http://cri-dist.univ-paris1.fr/diaz//adaptive/, [Online; accessed 14-January-2014]
5. Gent, I.P., Jefferson, C., Miguel, I., Moore, N.C., Nightingale, P., Prosser, P.e.a.: A preliminary review of literature on parallel constraint solving. In Workshop on parallel methods for constraint solving (2011), http://pn.host.cs.st-andrews.ac.uk/multicore-cp-review-wshop.pdf
6. IBM: CPLEX Optimizer, http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/, [Online; accessed 19-December-2013]
7. Kim, E.S.: Scheduling of uniform parallel machines with s-precedence constraints. Math. Comput. Model. 54(1-2), 576–583 (jul 2011), http://dx.doi.org/10.1016/j.mcm.2011.03.001
8. Michel, L., See, A., Van Hentenryck, P.: Transparent parallelization of constraint programming. INFORMS JOURNAL ON COMPUTING 21(3), 363–382 (2009), http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.186.6169&rep=rep1&type=pdf
9. Open MPI Team: Open MPI: Open Source High Performance Computing (November 2013), http://www.open-mpi.org/, [Online; accessed 15-January-2014]
10. Pedro, V.: Constraint Programming on Hierarchical Multiprocessor Systems. Ph.D. thesis, Universidade de Évora (May 2012), http://www.di.uevora.pt/~vp/pubs/vp-phd.pdf
11. Rolf, C.C., Kuchcinski, K.: Load-balancing methods for parallel and distributed constraint solving. In: CLUSTER. pp. 304–309. IEEE (2008), http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4663786&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D4663786
12. Rolf, C.C., Kuchcinski, K.: Parallel solving in constraint programming. In: MCC 2010: Third Swedish Workshop on Multi-Core Computing (November 2010), http://fileadmin.cs.lth.se/cs/Personal/Carl_Christian_Rolf/mcc2-paper-final.pdf

13. Schulte, C., Duchier, D., Konvicka, F., Szokoli, G., Tack, G.e.a.: Generic constraint development environment (November 2013), http://www.gecode.org/, [Online; accessed 18-January-2014]
14. Szymanek, R.: JaCoP (November 2008), http://jacop.osolpro.com/index.php?option=com_content&view=article&id=46&Itemid=28, [Online; accessed 15-December-2013]
15. Wikipedia: Quadratic assignment problem (January 2013), http://en.wikipedia.org/wiki/Quadratic_assignment_problem, [Online; accessed 30-December-2013]
16. Wikipedia: Squaring the square (September 2013), http://en.wikipedia.org/wiki/Squaring_the_square, [Online; accessed 20-December-2013]
17. Wikipedia: Knight's tour (January 2014), http://en.wikipedia.org/wiki/Knight's_tour, [Online; accessed 24-January-2014]
18. Wikipedia: Linear programming (January 2014), http://en.wikipedia.org/wiki/Linear_programming, [Online; accessed 13-January-2014]
19. Wikipedia: Magic square (January 2014), http://en.wikipedia.org/wiki/Magic_square, [Online; accessed 24-January-2014]
20. Wikipedia: Tianhe-2 (January 2014), http://en.wikipedia.org/wiki/Tianhe-2, [Online; accessed 24-January-2014]
21. Wikipedia: Travelling salesman problem (January 2014), http://en.wikipedia.org/wiki/Travelling_salesman_problem, [Online; accessed 16-January-2014]
22. Xie, F., Davenport, A.: Solving scheduling problems using parallel message-passing based constraint programming. Association for the Advancement of Artificial Intelligence (2009), http://optlab.mcmaster.ca/~feng/files/WSL_MPI.pdf

Extração de Informação e Classificação de Textos em Língua Natural

Nuno Miranda

Universidade de Évora

Resumo O presente trabalho é um levantamento de conceitos e do estado da arte das principais abordagens e metodologias envolvidas na extração de informação e na classificação de textos em Língua Natural. Este trabalho ainda fará uma breve análise nas áreas de Representação Ontológica e de Aprendizagem Automática. Este levantamento é um estudo prévio de extrema importância para a realização de futuros trabalhos, mais complexos, na área de classificação e de extração de informação.

1 Introdução

Actualmente a nossa sociedade auto designa-se por "A sociedade da informação". No entanto tal designação é frequentemente usada num sentido lato e de pura mediatização sem grande análise em profundidade sobre a qualidade/quantidade e a real utilidade da informação que dispomos.

É certo que nos últimos 50 anos acumulamos e coleccionamos mais informação do que na restante história da humanidade. No entanto surgem dúvidas, quanto ao real proveito de tais quantidades massivas de informação. Pois pior do que não ter informação é ser inundado por informação e não saber "navegar"nela.

- De que interessa ter uma enorme biblioteca com milhares de obras se estas não se encontrarem devidamente identificadas, organizadas, catalogadas?
- De que interessa ter uma enorme Biblioteca se não soubermos ler o seu conteúdo?
- De que interessa ter uma enorme Biblioteca onde os livros se encontram dispersos numa entropia tão elevada que é impossível estabelecer qualquer relação ou ligação entre conceitos e conteúdos das informações neles contidas?

Tais questões não se aplicam apenas a uma Biblioteca desorganizada, aplicam-se também a todo o conhecimento humano. Pois mais importante do que ter apenas dados em qualquer contexto, é sobretudo, ter acesso aos conceitos provenientes do cruzamento desses mesmos dados, obtendo-se assim informação e não apenas dados.

1.1 Objectivos

Os objectivos propostos e estipulados para este trabalho é de ganhar conhecimentos em:

- Formas de representação de conhecimento recorrendo a ontologias;
- Algoritmos de classificação existentes;
- Formas de extracção de informação a partir de bases textuais.
- Analise de outros trabalhos já desenvolvidos nestas áreas.

2 Conceitos

2.1 Ontologias e Web Ontology Language

Uma ontologia é um modelo de dados que representa um conjunto de conceitos dentro de um determinado domínio bem como as relações entre esses conceitos, as suas propriedades e também pode representar as restrições dos conceitos, hierarquia, relações e propriedades nesse domínio.

Desta forma uma ontologia permite que sejam feitos levantamentos estruturados sobre os mais diversos campos do conhecimento humano de uma maneira hierárquica e organizada.

As ontologias podem ser representadas em diversas linguagens, sendo a OWL uma das mais difundidas e utilizadas. A sua sigla têm origem no inglês *Web Ontology Language* e é um *standard* do *World Wide Web Consortium* (W3C) [1], e como o seu nome indica, foi desenvolvida para ser utilizada na Web Semântica. No entanto, pode ser utilizada facilmente noutros domínios para representar qualquer ontologia sobre um determinado domínio.

O OWL possui três dialectos; o OWL Full, o OWL DL e o OWL Lite. A diferença entres eles está no grau de expressividade que permitem associar aos conceitos e relações do domínio em estudo, sendo o OWL Lite o de expressividade mais simples, seguido pelo OWL DL, e pelo OWL Full que é o mais complexo e expressivo.

Os três dialectos estão contidos uns nos outros, podendo ser vistos como extensões de expressividade do dialecto anterior. Isto significa que uma ontologia definida em OLW Lite é valida em OWL DL, e por sua vez uma ontologia definida em OWL DL também é válida em OWL Full. O inverso destas relações já não se verifica.

As ontologias são geralmente constituídas por quatro elementos básicos:
- Classe - Grupos ou colecções abstractas que tanto podem conter ou agrupar outras classes ou instâncias de classes.
- Instância de classe - São elementos concretos de uma determinada classe em que os atributos tomam valores concretos. Não são entidades abstractas mas objectos concretos e objectivos.
- Atributo - São características que descrevem propriedades das classes, e que podem tomar diferentes valores nas várias instâncias de uma determinada classe.
- Relações - Como o nome indica, são relações entre classes, instâncias de classes e atributos. As relações podem ter ou não restrições.

2.2 Aprendizagem Automática Supervisionada.

Na aprendizagem supervisionada, os algoritmos de aprendizagem automática tem acesso a uma entidade externa que quase pode ser vista como um "Professor".

Essa entidade externa vai ser responsável por fornecer ao algoritmo bons exemplos para que ele sobre esses exemplos possa criar os seus modelos de aprendizagem e então criar regras indutivas para generalizar a partir do conjunto limitado fornecido pelo "professor".

O "professor"é que vai dispor do conhecimento da classificação dos objectos, e vai saber atribuir para determinado conjunto de atributos uma determinada classe classificadora do objecto. Assim o algoritmo de aprendizagem automática irá aprender baseado nos "conhecimentos"do "professor".

Mas esta aprendizagem "Aluno - Professor"tem sempre em vista o objectivo que o algoritmo não fique restrito a saber classificar apenas casos iguais aos apresentados pelo "professor", mas que tenha alguma inteligência indutiva para saber classificar eventuais novos casos que surjam diferentes dos apresentados pelo "professor".

Nos casos práticos, o papel de "professor"é efectuado por humanos que classificam previamente conjuntos de dados seguindo um conjunto de regras obtidas através da observação e do raciocínio lógico humano, ficando assim esses algoritmos "viciados"pelos "professores".

Esta técnica é utilizada por exemplo em vários domínios de classificação automática, tais como classificação de imagens e de textos, em que são apresentados exemplos previamente classificados e rotulados para depois o algoritmo generalizar essa classificação e automatiza-la a novos casos.

2.3 Aprendizagem Automática Não-Supervisionada.

A aprendizagem não-supervisionada é uma aprendizagem que não tem qualquer tipo de entidade externa que ensine e faculte exemplos de aprendizagem ao algoritmo de aprendizagem automática. Isto pode parecer estranho à primeira vista, pois se o algoritmo de aprendizagem automática não tem qualquer conhecimento prévio ou exemplos de como agrupar os objectos correctamente como pode ele agrupar o que quer que seja correctamente?

É precisamente na resposta à pergunta anterior que reside a motivação de existir a aprendizagem automática não-supervisionada, pois a aprendizagem não-supervisionada tem um objectivo bastante diferente da aprendizagem supervisionada. A aprendizagem supervisionada parte de um conjunto de objectos pré-classificados e induz para novos casos. A aprendizagem não-supervisionada, parte de início sem nenhuma "ideia"pré-adquirida e vai então agrupar os objectos em classes ditas abstractas, a partir das propriedades dos objectos.

O objectivo é permitir que quando se tem grandes quantidades de objectos complexos e aparentemente caóticos de serem classificados por humanos, tornando-se assim impossível de existir uma entidade com o papel de "professor", pois os humanos com o papel de "professor", não têm capacidade de análise e síntese das propriedades dos objectos devido à sua elevada complexidade.

Mas com a aprendizagem não-supervisionada é possível que os algoritmos de aprendizagem automática efectuem a criação de um conjunto de classes classificativas para uma família de objectos, que de outro modo seria impossível obter devido à complexidade dos objectos em estudo.

2.4 Algoritmos de Classificação Automática

Os algoritmos de aprendizagem automática supervisionada agrupam-se em várias famílias, conforme os seus mecanismos base de funcionamento. Dentro de cada família, o princípio geral de funcionamento é semelhante, variando apenas alguns pontos ou afinações.

Árvores de Decisão Os algoritmos de aprendizagem automática baseados em árvores de decisão, são uma das família mais fáceis de perceber conceptualmente o seu funcionamento. Baseiam-se em simples árvores de decisão onde cada nó é uma condição e cada folha é um resultado final. A Figura 1 apresenta um exemplo para determinar se um dia é indicado ou não para jogar ténis.

O funcionamento da árvore é muito simples. Parte-se da raiz, que é o primeiro nó e onde se encontra a primeira condição, depois segue-se caminho conforme o nosso atributo cumpre essa condição. Cada ramo da árvore corresponde a um dos valores possíveis do atributo do nó de onde partem esses ramos. Segue-se sucessivamente para o nó seguinte até chegar às folhas da árvore. Cada folha tem a classificação final, podendo haver várias folhas com o mesmo resultado.

Desta descrição é possível concluir que uma árvore de decisão não passa de uma disjunção de conjunções lógicas sendo os ramos as conjunções e os nós as disjunções.

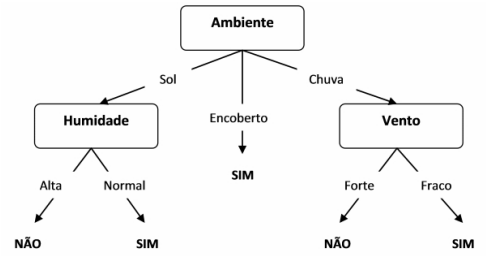


Figura 1. Árvore de decisão simples em que cada nó é uma condição e cada folha é um resultado final. Neste caso típico, para determinar se a classe "Ir Jogar Ténis"toma valores positivos ou negativos basta ir respondendo às condições e ir descendo a árvore até chegar a uma das folhas com o resultado final.

Algoritmo ID3. O algoritmo de aprendizagem automática ID3 [2] foi inventado por Ross Quinlan e é considerado um marco e um ponto de partida nos algoritmos de árvores de decisão, pois é um dos mais simples e fáceis de compreender.

Algoritmo C4.5. Este algoritmo é uma versão melhorada do ID3, que conta com nova abordagem e regras na construção da árvore, para que ela não seja sobre-ajustada aos casos de treino.

Este algoritmo também foi desenvolvido pelo mesmo autor do ID3, Ross Quinlan. Tanto o algoritmo ID3 como o C4.5 [3] são algoritmos livres, no entanto existe uma versão comercial do C4.5 com alguns melhoramentos chamada de C5.0 [4].

Todo o processo de construção da árvore de decisão do C4.5 é igual ao do algoritmo ID3. A principal diferença e melhoria é que o C4.5 após efectuar a construção da árvore de decisão, efectua a chamada poda da árvore, com o objectivo de cortar da árvore os ramos demasiado longos e demasiado específicos e que são responsáveis por sobre-ajustar a árvore ao conjunto de aprendizagem.

Esta técnica é chamada de pós-poda, pois ocorre após a árvore estar toda criada. Existem também outros algoritmos da família das árvores de decisão que usam outra técnica apelidada de pré-poda, que consiste em restringir o crescimento da árvore logo durante a sua criação [5].

A pós-poda do C4.5 tem como objectivo reduzir a complexidade da árvore, que implica eliminar algumas das suas sub-árvores, reduzindo assim a altura da árvore e aproximar as folhas à raiz.

Para ser efectuada uma determinada poda é efectuada uma avaliação estatística. Para cada nó são avaliados os erros de classificação que resultam desse nó e dos seus nós descendentes; só é efectuada a poda do nó se esta não implicar uma redução no desempenho da árvore. Neste aspecto o C4.5 é um pouco conservador, pois esta avaliação é pessimista, de modo a que não se corra o risco de reduzir a eficácia da árvore. Existem outros algoritmos que "ariscam"mais e efectuam uma poda mais drástica da árvore.

Aprendizagem Probabilística ou Bayesiana é outra grande família de algoritmos de aprendizagem automática. Como na família anteriormente visada, em que todos os algoritmos tinham em comum uma árvore de decisão na sua base, esta família dos algoritmos probabilísticos ou Bayesianos [6] também tem algo comum na sua base de funcionamento: cálculos probabilísticos que têm como base o teorema de Bayes.

Naïve Bayes. [6] é um dos algoritmos de aprendizagem automática mais conhecido e utilizado que tem como base para o seu funcionamento um classificador probabilístico baseado no teorema de Bayes. A designação "Naïve"provém do algoritmo pressupor que os vários atributos que descrevem

os objectos são independentes, o que na realidade raramente acontece. Assim, entre os vários atributos que discriminam a classe do objecto, cada atributo contribui independentemente para a probabilidade do objecto fazer parte de uma classe ou outra, não havendo qualquer correlação entre os diversos atributos na hora de decidir a classe do objecto.

No entanto o facto do algoritmo fazer essa simplificação não implica que ele obtenha maus resultados. Pelo contrário, o algoritmo Naïve Bayes é um algoritmo que na maior parte dos domínios apresenta bons resultados.

Máquinas de Vectores de Suporte (SVM's) As Máquinas de Vectores de Suporte, mais conhecidas por SVM's[1], são uma família de algoritmos de aprendizagem automática desenvolvida inicialmente pelos trabalhos de Vapnik e Chervonenkis [7].

De uma maneira muito simplista, temos os objectos da nossa colecção que pretendemos classificar. Esses objectos podem ser classificados de modo binário, tomando as classes C e C', e são caracterizados pelos atributos do conjunto $A = \{A_1, A_2, ..., A_n\}$.

Cada objecto pode ser representado na estrutura de SVM's como sendo um vector n-dimensional num espaço vectorial de dimensão n obtendo uma determinada disposição geográfica consoante os valores dos seus atributos.

O classificador dos SVM's surge como um algoritmo que vai obter e optimizar um hiperplano de dimensão $n-1$ dentro do nosso espaço n-dimensional, que separa as duas classes. Esse hiperplano pode ser visto como uma fronteira, mas ao invés de ser uma fronteira bidimensional como a dos mapas, é uma fronteira de dimensão $n-1$.

Quando qualquer novo objecto for adicionado à colecção e se pretender a sua classificação referente à classe C ou C', basta representar esse objecto no espaço vectorial n-dimensional, e ver se a sua representação ocorre de um lado ou de outro da "fronteira"que separa as duas classes.

No entanto, no espaço vectorial pode existir uma infinidade de hiperplanos capazes de dividir as duas classes de objectos, levantando a questão de qual hiperplano se adequa melhor. Sendo o algoritmo SVM's responsável por essa decisão.

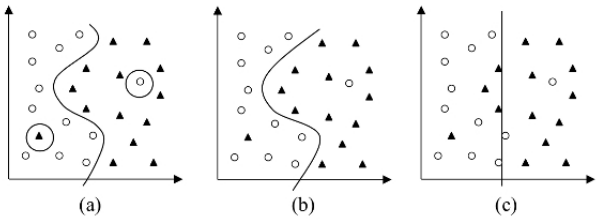


Figura 2. Exemplos de classificação Binária. Desde o modelo excessivamente complexo, permissível a *outliers* desviantes e a provável má classificação para novos casos (a), modelo equilibrado e já insensível a *outliers* desviantes (b), e modelo demasiado simplista e com elevado número de classificações erradas (c).

Essa decisão é baseada numa optimização matemática, que por norma tenta obter o hiperplano que consegue maximizar a separação entre as classes, de modo a que a distância média do hiperplano aos elementos mais próximos de C e C' seja a maior possível.

No entanto nem sempre os SVM's lineares ou suaves são suficientes para se adaptarem a todos os casos de estudo com sucesso, pois em determinados casos os dados assumem tal complexidade que são impossíveis de discriminar correctamente com um hiperplano linear ou linear suave.

[1] Do Inglês, *Support Vector Machines*.

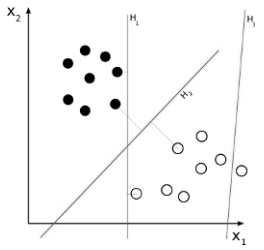


Figura 3. Exemplos de Hiperplanos numa simplificação em duas dimensões para fácil compreensão. Temos duas classes, sendo o H_3 um hiperplano falhado, pois divide mal as classes. Os hiperplanos H_1 e H_2 separam bem as duas classes, no entanto o H_2 é o melhor, pois maximiza a margem média aos elementos mais próximos das duas classes.

Para estes problemas existem os SVM's não lineares, que podem tomar diversas formas e complexidades. Esses diversos SVM's não lineares são criados por diferentes funções de núcleo[2]. Sendo esta função responsável pelo cálculo e optimização da hiper-superfície separadora. Como exemplo, os núcleos não lineares mais difundidos são os polinomiais, radiais e hiperbólicos.

2.5 Bases Representativas

A maior parte dos algoritmos de classificação de texto baseados em aprendizagem supervisionada trabalha sobre um saco de palavras que é uma listagem de palavras que ocorrem no domínio em estudo. Sobre essas palavras são efectuados estudos de contagem e frequências de modo a extrair daí regras de classificação. Assim uma frase inicialmente constituída por palavras passa a ser representada por um vector de atributos que discriminam essa frase. Como os algoritmos de classificação não lidam directamente com as frases de palavras, vão ter como entrada os vectores de atributos.

A forma de representar essas contagens e frequência pode seguir diversas abordagens distintas. A essas abordagens chama-se "Bases Representativas".

As bases representativas podem ser obtidas a partir dos seguintes parâmetros:

$\boldsymbol{f}_{ij}$ − número de ocorrências da palavra i no documento j

$\boldsymbol{fd}_j$ − o número total de palavras no documento j

$\boldsymbol{ft}_i$ − o número de documentos em que o termo i aparece pelo menos uma vez

$\boldsymbol{v}_{ij}$ − a "importância"de um determinado termo i no documento j.

Base Binária É a base representativa mais simples onde o valor do atributo toma o valore de 1 ou 0 consoante a palavra ocorra ou não, respectivamente. Uma palavra que ocorra apenas uma vez ou ocorra um número elevado de vezes, tem o mesmo peso, que neste caso é 1.

Em certos cenários tal limitação não é importante, mas noutras situações não é bom, pois não discrimina a cardinalidade da ocorrência do termo, apenas a assinala.

É um dos métodos mais simples pois não é normalizado nem tem em conta a ocorrência do termo no conjunto. Tem como vantagem não requerer qualquer calculo pós-contagem.

E toma a seguinte representação:

[2] Do Inglês, *Kernel function*.

$$v_{ij} = \begin{cases} 1, f_{ij} > 0 \\ 0, else \end{cases}$$

Base de Ocorrências de Termos Esta base representativa sendo mais complexa que a binária também é relativamente simples, pois também não é normalizada. No entanto permite obter mais informação que a representação em base binária, pois permite registar a cardinalidade de ocorrências dos termos na frase. Quando um termo ocorre n vezes, é o próprio valor n que o representa, ou seja:

$$v_{ij} = f_{ij}$$

Base de Frequência de Termos Esta base representativa é em tudo idêntica a Base de Ocorrências de Termos, com a excepção de sofrer normalização Euclidiana. Assim um termo em vez de ser representado pelo seu número absoluto de ocorrências é representado no intervalo $[0, 1]$. Este valor é obtido a partir de:

$$v_{ij} = \frac{f_{ij}}{fd_j}$$

TFIDF Por fim temos a base representativa TFIDF[3], que tem em conta a ocorrência do termo no documento e no conjunto de todos os documentos ou corpus.

Um termo por um lado tem mais peso quanto mais se repete no documento, mas por sua vez esse termo perde peso quantas mais vezes se repetir no corpus dos documentos. Esta base representativa aplicada a este trabalho em vez de utilizar como unidade de classificação o documento utilizou a frase.

Obtém-se pela fórmula:

$$v_{ij} = \frac{f_{ij}}{fd_j} \log\left(\frac{|D|}{ft_i}\right)$$

3 Trabalho Relacionado

Esta secção faz uma síntese do estado da arte relacionado com a área deste trabalho. Incidindo sobre duas áreas distintas, a área de classificação de textos em língua natural na Secção 3.1 e a área da extracção de informação a partir de textos em língua natural na Secção 3.2.

3.1 Classificação de Textos em Língua Natural

Existem diversos trabalhos na área de classificação de textos que se dividem em diferentes abordagens dependendo do nível de informação básica com que trabalham. Há trabalhos que utilizam um nível de sub-palavra em que a palavra é decomposta e analisada a sua morfologia. Outros trabalhos trabalham ao nível da palavra, em que a palavra é a entidade básica e é analisada a sua informação lexical. Surgem ainda abordagens que trabalham num nível acima da palavra, uns seguem a via semântica, em que o sentido dos textos é analisado, e a via pragmática em que o texto é analisado de acordo com o seu significado e contexto em que ocorre. Existem no entanto, técnicas mais complexas que para além da manipulação directa sobre o saco de palavras, recorrem ainda à análise morfológica, sintáctica e semântica dos textos.

[3] Sigla proveniente do Inglês, *Term Frequency-Inverse Document Frequency.*

Thorsten Joachims [8] efectua testes de comparação entre diversos algoritmos para a mesma situação experimental: Naive Bayes, Rocchio, C4.5, K-n vizinhos os SVM com *kernel* RBF e polinomial. Joachims chega a conclusão que os SVM apresentam os melhores resultados quer a nível de eficácia temporal quer a nível de desempenho da precisão e cobertura.

Joachims explora os motivos que conduzem os algoritmos SVM a obterem bons resultados de classificação de textos em relação a outros algoritmos quando num contexto de aprendizagem supervisionada, e conclui:

- Os SVM permitem trabalhar facilmente e sem quebra de performance com uma elevada dimensão de atributos de entrada em problemas de classificação de texto.
- Por permitir um grande número de atributos sem quebras de performance, não é necessário descartar parte dos atributos menos importantes para se incrementar a performance.
- Os SVM's suportam o modo *sparse* nos atributos de entrada. O modo *sparse* apenas representa atributos com valores diferentes de zero, o que é muito importante, pois em problemas de classificação de texto a maioria dos atributos tomam valores zero. Assim os SVM's ao suportarem este modo, é lhes possível eliminar grande parte da redundância inútil dos dados de entrada, contribuindo também para a menor dimensão dos ficheiros de entrada.
- Por fim Joachims refere que a maior parte dos problemas de classificação de textos com diversas classes são geralmente representados por modelos lineares, o que encaixa perfeitamente no perfil dos SVM com *kernels* linear, sendo desnecessária a utilização de *kernels* com uma maior complexidade.

Para terminar, apenas há a referir que os trabalhos de Joachims foram inteiramente desenvolvidos no idioma inglês.

Akiko Aizawa [9] apresenta um método que incorpora técnicas de processamento de língua natural nos tradicionais processo de classificação de textos. Para isso utiliza modelos de linguagem probabilísticos baseados nos pesos dos termos, estimação de ocorrência de termos e recurso a analisadores morfológicos das palavras *(POS, part-of-speech)*.

Os resultados apresentados mostram que a utilização das técnicas de processamento de língua natural na classificação de textos utilizando SVM melhora visivelmente o desempenho do classificador (cobertura e precisão) quando comparado com a abordagem tradicional do simples saco de palavras.

Este trabalho também foi inteiramente desenvolvido no idioma Inglês.

Gonçalves e Quaresma [10] comparam o desempenho entre os algoritmos de SVM, C4.5 e Naive Bayes, na classificação de textos jurídicos na língua Portuguesa.

Além de apresentarem a tradicional abordagem pelo saco de palavras, foram usadas técnicas de transformação das palavras no seu lema e de selecção de palavras utilizando a sua classificação morfológica e recorrendo a listas de palavras de paragem para descartar palavras irrelevantes.

A nível de resultados, os melhores resultados de F1 foram apresentados pelo C4.5, mas seguido de muito próximo dos SVM com uma diferença quase irrelevante. No entanto a nível de performance temporal o SVM ganha com grande vantagem, pois para a experiência em causa demorou apenas 10 minutos ao invés das 8 horas necessárias para o processamento pelo C4.5. Já as técnicas de transformação utilizadas mostraram um significativo melhoramento dos resultados finais em comparação ao simples saco de palavras.

Silva e Vieira [11] apresentam um trabalho semelhante ao anteriormente descrito, mas utilizando um conjunto de dados distintos, mas chegam a conclusões semelhantes. O corpus do estudo teve como base um conjunto de artigos do Jornal Folha de São Paulo de 1994.

Neste trabalho foi realizada uma comparação entre os algoritmos SVM e árvores de decisão em tarefas de classificação de texto. Para além da utilização do saco de palavras tradicional, também

recorrem a informação linguística na construção dos atributos que descrevem os documentos. Para a extracção de informações linguísticas foi utilizado o analisador sintáctico PALAVRAS [12], e a selecção dos atributos era feita na classe gramatical das palavras.

A nível de resultados, constataram que as classes gramaticais discriminantes que mais melhoravam os resultados dos SVM e das árvores de decisão foram os substantivos e nomes próprios, e os mais irrelevantes os verbos e sintagma nominais (que já funcionam ao nível da multi-palavra).

O desempenho dos algoritmos SVM e árvores de decisão foi semelhante, embora tenham constatado que para poucos atributos, as árvores de decisão levam uma ligeira vantagem sobre os SVMs, situação que se inverte quando o número de atributos aumenta.

Noutro trabalho de Silva [13], é feita a mesma experiência, mas utilizando redes neuronais artificiais para a etapa de classificação. Os resultados desta abordagem são bons, no entanto ficam ligeiramente atrás dos resultados obtidos pelos SVM em Silva [11].

Bloehdorn [14], constata melhoramentos em tarefas de classificação de textos com recurso ao uso de características extraídas de ontologias sobre o domínio dos textos a ser classificados. O estudo incide sobre o domínio da medicina e as ontologias auxiliares à classificação são geradas automaticamente através de aprendizagem não supervisionada.

A abordagem é baseada na distribuição de hipóteses, verificando durante o processo de classificação, se os termos analisados são semanticamente similares aos do contexto no qual estão inseridos na ontologia. Nesta fase pode ocorrer um processo de generalização, que tem como finalidade adicionar conceitos mais gerais aos conceitos existentes na ontologia. Desta forma, cria-se uma abrangência maior para com os conceitos na ontologia dos diversos documentos analisados e que possuem características em comum.

À semelhança de Bloehdorn [14], Wu *et al* [15] realiza a classificação de texto com recurso a informação de ontologias de domínio adquiridas dos textos, através de regras morfológicas e métodos estatísticos.

A ontologia de domínio pode ser utilizada para identificar a estrutura conceptual das frases de um documento, e assim classificá-la. Além disso, pode ser utilizada como base para outras aplicações além da classificação de textos, como por exemplo, sistemas de pergunta e resposta e sistemas de organização de conhecimento.

No mesmo trabalho foi efectuada a comparação de resultados com as diversas ontologias geradas em bruto e com versões das mesmas revistas por humanos, onde as ontologias revistas apresentavam resultados ligeiramente melhores.

Segundo os autores a utilização de ontologias de domínio apresenta vantagens relativamente a outros mecanismos de representação do conhecimento, já que podem ser lidas, interpretadas e editadas por seres humanos.

Em Cordeiro [16] é efectuada uma extracção de elementos relevantes em textos e páginas da Web. Este trabalho tem duas partes, sendo a primeira dedicada a classificação automática de textos no domínio relativo à venda de habitações e a segunda parte relativa à extracção de elementos relevantes referentes aos mesmos anúncios.

A primeira parte, resume-se a classificar os textos em duas classes, textos de venda de habitações e textos de não venda de habitações.

Para isso foram utilizadas e testadas três abordagens distintas de modo a poder ser efectuado um estudo da qual obteria melhores resultados. As abordagens foram k-n vizinhos mais próximos, Naive Bayes e Naive Bayes com escolha de atributos. A diferença entre o Naive Bayes e Naive Bayes com escolha de atributos (50 e 200), é que no primeiro todas as palavras pertencentes ao saco de palavras de anúncios funcionam como atributos, enquanto com escolha de atributos só alguns dos mais relevantes são escolhidos para servirem de atributos.

A nível de resultados, Cordeiro concluiu que os melhores eram obtidos com o Naive Bayes de 200 atributos, seguindo-se o Naive Baies com 50 atributos, depois o Naive Bayes com todos os atributos e por fim o k-n vizinhos mais próximos.

3.2 Extracção de Informação em Textos

Existe uma diversidade enorme de trabalhos realizados que seguem diversas vias e abordagens no processo de extracção da informação de textos em língua natural. No entanto, na área de extracção de informação é obrigatório referir o ciclo de conferencias MUC do inglês *"Message Understanding Conferences"* decorridas entre 1987 (MUC 1) até 1997 (MUC 7) que promoviam a competição de sistemas de extracção de informação e que foram promissoras para a criação de vários sistemas inovadores e que ainda hoje são referência.

Sistemas MUC Riloff [17] apresentou o primeiro sistema de extracção baseado na criação de um dicionário de extracção, denominado de AutoSlug. O dicionário consistia numa colecção de padrões de extracção designados por nós de extracção. Os nós de extracção eram apenas uma palavra de activação com algumas regras de restrição linguística a serem aplicadas ao texto de onde se pretendia extrair informação.

Este sistema era treinado com um conjunto de exemplos anotados manualmente e por norma as palavras de activação eram nomes, verbos ou substantivos. O sistema tinha um analisador sintáctico que classificava as palavras, após uma palavra de activação ser desencadeada eram evocadas as respectivas regras de restrições que iam determinar se essa palavra seria ou não considerada para a extracção de informação.

Dois anos mais tarde em Riloff [18] surgiu com uma nova versão que apresentava alguns melhoramentos significativos. As palavras de activação passaram a poder ter mais do que um conjunto de regras de extracção de modo a permitir ter diversos contextos e significados diferentes. Também deixou de ser necessário anotar todos os exemplos de treino, sendo apenas necessário anotar os mais relevantes para o domínio do problema em questão.

Nas ultimas conferencias realizadas, a MUC-6 e MUC-7 um dos sistemas que mais se destacou devido aos seus bons resultados foi o LOLITA [19] que tem origem no inglês *"Large-scale, Object-based, Linguistic Interactor, Translator, and Analyser"*. É um projecto em desenvolvimento desde 1986 no laboratório de engenharia de linguagem natural da Universidade de Durham.

Este sistema é completamente genérico para língua natural e pode ser facilmente adaptado a qualquer domínio. O sistema consiste numa complexa e vasta rede semântica, com mais de 100000 nós e 1500 regras gramaticais. Esta abordagem é inspirada na já célebre *WordNet* [20]. Este sistema é considerado um dos maiores sistemas de processamento de linguagem natural e serve de motor para um conjunto de aplicações implementadas que vão desde a tradução entre línguas até à extracção de informação como é o caso do trabalho descrito por Marco Constantino em [21].

A configuração e utilização do sistema num domínio específico é feita com relativa facilidade [22], através da criação de um conjunto de modelos específico para o domínio do problema em análise. Os modelos neste caso, não são mais que uma estrutura com um conjunto de campos para serem preenchidos segundo regras explicitamente definidas no próprio modelo.

Como ponto menos forte deste sistema, há a apontar o facto que as regras de extracção de elementos relevantes terem de ser definidas manualmente pelos utilizadores. Através dos modelos que dão alguma liberdade de ajuste e de afinação em domínios relativamente pequenos torna-se pouco eficiente de executar em domínios muito latos e abrangentes.

O grande poder do sistema LOLITA reside na sua base de conhecimento interna, que assenta numa rede semântica, que permite um processamento muito bom sobre a linguagem natural.

Krupa [23] participou na MUC-6 com o sistema Hasten, onde obteve bons resultados na competição desse ano.

O sistema também se baseia em nós de extracção com palavras activadoras e regras de semântica. As palavras activadoras têm de ser marcadas à mão para cada domínio, no entanto as regras de semânticas de cada palavra são criadas automaticamente com algoritmos de aprendizagem.

Após se criar a lista de nós activadores com as palavras e respectivas regras, o sistema está apto a extrair informação relevante de novos textos.

Sempre que se encontra uma palavra activadora no texto, são extraídas as suas regras de semântica e comparadas com às registadas no nó de activação dessa palavra. Se as regras estiverem até uma determinada distância, a palavra é considerada, e a informação extraída.

Soderland [24] apresentou na MUC o sistema Crystal. O sistema era baseado num dicionário de extracção com uma série de palavras importantes no domínio em causa, conjuntamente com as respectivas restrições de validação para a extracção. Essas restrições incidiam nas estruturas morfológicas, sintácticas e de conceito.

Para criar o conjunto de treino as palavras tinham de ser etiquetadas ao nível da sintaxe e da semântica e eram ainda identificadas as ocorrências de conceitos. Os conceitos podiam variar consoante o domínio em estudo, mas por norma eram utilizados os conceitos "Pessoa Genérica", "Nome Próprio", "Organização Genérica", "Evento"e "Empresa".

Com esse conjunto de treino, o sistema infere regras de restrições à extracção da informação.

Sistemas fora do âmbito MUC Em Cordeiro [16], já referenciado na Secção 3.1 de classificação de textos em língua natural, a segunda parte desse trabalho é relativa à extracção de elementos relevantes nos anúncios seleccionados na primeira parte do trabalho. Os elementos relevantes eram referentes ao tipo, preço e o local da habitação.

Para a fase de extracção foram usadas duas abordagens, sendo elas a extracção por via de regras pré-definidas manualmente e outra por via de regras definidas através de um sistema de aprendizagem com árvores de decisão com o algoritmo C4.5.

Tanto numa abordagem como na outra, as regras obtidas diziam respeito aos diversos elementos que se pretendiam extrair (tipo, local e preço), e as regras simplesmente definiam a vizinhança do elemento em análise para extracção.

No entanto Cordeiro chegou à conclusão que a abordagem com regras criadas pelo C4.5 era superior na extracção dos elementos relevantes para os três tipos de elementos (tipo, local e preço).

Cordeiro teve ainda de estudar a dimensão ideal do contexto a analisar, ou seja a quantidade de palavras antes e depois a serem analisadas de modo a decidir a extracção ou não da informação. Chegou à conclusão que o valor em que maximizava a cobertura e precisão era com janela de 4, ou seja analisando 4 palavras antes e 4 palavras depois.

4 Conclusão

Em conclusão deste breve trabalho, podemos observar que os principais objectivos foram concluídos. Nomeadamente o levantamento sobre ontologias, uma análise sobre as principais famílias de algoritmos de aprendizagem e de classificação automática. Assim como das formas de extração de informação em bases textuais.

Por fim, também se atingiu o objectivo de efectuar uma breve análise de outros trabalhos relevantes, na área de classificação de textos e de extração de informação. Trabalhos esses, de onde

se obtiveram ideias e conhecimentos importantes que podem ser utilizados no desenvolvimento de trabalhos futuros.

Referências

1. Consortium, W.W.W.: (Consultado em 2013) http://www.w3.org/.
2. Quinlan, J.R.: Induction of decision trees. Mach. Learn (1986) 81–106
3. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA (1993)
4. RESEARCH, R.: (Consultado em Março 2010) http://www.rulequest.com/see5-comparison.html.
5. Quinlan, J.R.: Bagging, boosting, and c4.5. In: In Proceedings of the Thirteenth National Conference on Artificial Intelligence, AAAI Press (1996) 725–730
6. John, G., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann (1995) 338–345
7. Vapnik, V., Chervonenkis, A.: A note on one class of perceptrons. Automation and Remote Control **25** (1964)
8. Joachims, T., Informatik, F., Informatik, F., Informatik, F., Informatik, F., Viii, L.: Text categorization with support vector machines: Learning with many relevant features (1997)
9. Pages, S.N., Aizawa, A.: Akiko aizawa linguistic techniques to improve the performance of automatic text categorization. In: In Proceedings 6th NLP Pac. Rim Symp. NLPRS-01. (2001) 307–314
10. Gonçalves, T., Quaresma, P.: A preliminary approach to the multilabel classification problem of Portuguese juridical documents. In Moura-Pires, F., Abreu, S., eds.: EPIA-03, 11th Portuguese Conference on Artificial Intelligence. LNAI 2902, Évora, PT, Springer-Verlag (December 2003) 435–444
11. Silva, C., Vieira, R.: Categorização de textos da língua Portuguesa com árvores de decisão, SVM e informações linguísticas. In: TIL-07, 5º workshop em Tecnologia da Informação e da Linguagem Humana, Rio de Janeiro, BR (July 2007) 1650–1658
12. Eckhard, S.A., Bick, E., Haber, R., Santos, D.: F!oresta sintfi(c)tica": A treebank for portuguese (2002)
13. Silva, C., Vieira, R.: Uso de informações lingüísticas em categorização de textos utilizando redes neurais artificiais. In: SBNR-04, 8º Simpósio Brasileiro de Redes Neurais, Rio de Janeiro, BR (2004) 1–16
14. Bloehdorn, S., Hotho, A.: 2006): Learning ontologies to improve text clustering and classification. In: Proceedings of the 29th Annual Conference of the German Classification Society (GfKl, Springer (2005)
15. et al, S.H.W.: Text categorization using automatically acquired domain ontology, Sapporo, JP (2003)
16. da Costa Cordeiro, J.P.: Extracção de elementos relevantes em texto/páginas da world wide web. Master's thesis, Departamento de Ciência de Computadores Faculdade de Ciências da Universidade do Porto (Julho 2003)
17. Riloff, E.: Automatically constructing a dictionary for information extraction tasks. National Conference on Artificial Intelligence (1993)
18. Riloff, E.: An empirical study of automated dictionary construction for information extraction in three domains. Artificial Intelligence **85** (1996) 101–134
19. Garigliano R., Urbanowicz A., N.D.J.: Description of the lolita system, as used in muc-7, University of Durham (1998)
20. G., M.: Wordnet: An online lexical database. In: International Journal of Lexicography. (1990)
21. M., C.: Financial information extraction using pre-defined and user-definable templates in the lolita system, University of Durham (1997)
22. M., C.: The lolita user-definable template interface, University of Durham (2001)
23. R., K.G.: Description of the sra system as used for muc-6, 221-235. San Mateo: Morgan Kaufmann, In Proceedings of the Sixth Message Understanding Conference (MUC-6) (1995)
24. G., S.S.: Learnning domain-specific text analysis rules, University of Massachusetts at Amherst (1996)

New botnets trends
Social and Mobile Botnets

Alexandra Margarida Moedas

[1] Universidade de Évora alexandramoedas@gmail.com
[2] Instituto Politécnico de Beja amoedas@ipbeja.pt

Abstract. Botnets have become a serious threat to enterprise networks and the Internet itself. Recently we have seen an increase in malware on mobile devices and online social networks, botnets are one of them. This paper presents a brief introduction to botnets basic concepts, how they work and perform malicious activities with special focus on mobile and social botnet.

Keywords: Botnet, Malware, Mobile Botnet, Online Social Networks

1 Introduction

Botnets have become a serious threat to enterprise networks and the Internet itself, they are used for various purposes, most of them related to illegitimate activity, such as launching Distributed Denial of Service (DDoS) attacks, sending spam, trojan and phishing emails, illegally distributing pirated media and software, stealing information and computing resources, e-business extortion and identity theft. Experts believe that approximately 16 to 25% of the computers connected to the Internet are members of botnets.[23]
Detect and shut down a botnet is a big challenge, botnets developers use numerous evasion techniques to prevent their detection and consequent deactivation.
The recent growth of botnet activity in cyberspace has attracted the attention of the research community. This paper presents some of this research with special focus on mobile and social botnet.
The paper is organized as follows. Section 2 describes some Basic Concepts. Section 3 presents some of the Attacks that can be preformed by botnets. In Section 4, Detection, Defense and Evasion techniques. In Section 5 is approached the concept Mobile Botnet and some examples. In Section 6, the concept Social Botnets and some examples. Finally, Section 7 presents some conclusions.

2 Basic Concepts

Botnets are networks formed by infected machines, called bots (derived from the word robot) that are controlled by one or more attackers, called botmasters or botherder with the intention of performing malicious activities.

The critical difference between botnets and other malware is that botmasters use a Command and Control (C&C) channel to coordinate large numbers of individual bots to launch potentially much more damaging attacks.[7]
Botnets usually use well defined communication protocols such as IRC(Internet Relay Chat), HTTP(Hypertext Transfer Protocol), P2P(Peer-to-Peer), or others. IRC is a chat system that provides one-to-one and one-to-many instant messaging over the Internet [17], and the first protocol used to control botnets.

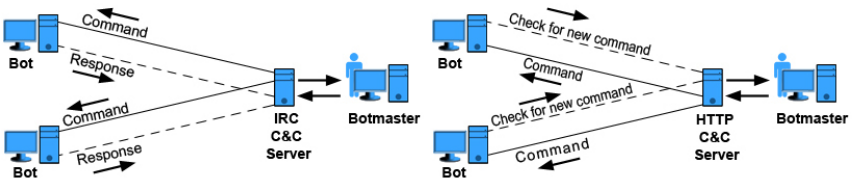
2.1 Botnet, Bot, Botmaster, C&C channel

The botmaster controls the bots through various C&C channels, C&C is the most important component of a botnet because it is used to establish communication between botmaster and bots.
A bot is a software program (malware) installed in a vulnerable host that allow botmasters to control de host computer remotely and make them perform various actions, normally malicious. What distinguishes a bot from other types of malware is the C&C channel.
Botmasters or botherders are malicious users who control botnets by issuing commands to bots to perform illegal activities. A botnet is a collection of bots connected to a C&C channel, i.e., a network of bots that is waiting for a command to perform malicious activities. [23]
C&C infrastructure is the most critical component of a botnet because is the only way to control bots within the botnet and is necessary for maintaining a stable connection within this infrastructure to operate efficiently. The choice of the C&C architecture is crucial for the success of the botnet, it determines her robustness, stability and reaction time.

2.2 Topologies and Protocols

C&C channels can operate on different communication protocols and use various topologies.
Topology can be divided into two main categories: centralized and distributed. The centralized is a model like the client-server, all bots establish their communication channel with only one or few servers responsible for sending commands to bots and provide malware updates. This topology is easy to implement, produces little overhead, has a quick reaction time, good coordination and it is easy to monitor, straightforward in their implementation and extremely stable. But it has a problem, the C&C server itself is a central point of failure, allowing a detected botnet to be shut down quite easily, this weakness motivated the development of decentralized architectures.
The decentralized architecture is implemented as a P2P system, in this model doesn't exists a centralized server, all member nodes are equally responsible for passing on traffic. This model is harder to neutralize because the discovery of several bots does not necessary mean the loss of the entire botnet, but their design and development can be very complex and the maintenance of the C&C channels may incur significant costs, that's why many botmasters prefer a C&C

structure using an IRC channel.
An ideal botnet would hence have the simplicity and stability of star-shaped networks, coupled with the scalability and resilience of P2P ones.[4]
In order to disperse their communications in the general flow of Internet traffic, botnets commonly use standard Internet protocols such as IRC, HTTP, SIP(Session Initiation Protocol), or P2P file sharing protocols, to support the C&C communications occurring over the network. These protocols determine the type of architecture of the C&C and according to them, botnets can be classified as IRC-based, HTTP-based, DNS-based or P2P, according to their C&C architecture.[7]
The IRC was the first protocol used to create botnets, is the most frequent, mainly due to its implementation simplicity. The attacker uses a compromised machine to setup an IRC server, and posts his commands to one ore more IRC channels, individual bots connect to these channels, listening for commands to execute. [4] This protocol is ideal for centralized and allows nearly instant communication among large botnets, but has a disadvantage, IRC traffic is not very common on the Internet, can be easily detected, filtered, or blocked.
In order to contour this problem, botmasters started using HTTP to manage centralized botnets. HTTP is a regular protocol for Web traffic, the advantage of using it for C&C is that it must be allowed to pass through virtually all firewalls, since HTTP comprises a majority of Internet traffic, even closed firewalls that only provide Web access will allow HTTP traffic to pass. However, this protocol has an a disadvantage, does not provide the instant communication and built-in, scale-up properties of IRC: bots must manually poll the central server at specific intervals. With large botnets, these intervals must be large enough and distributed well to avoid overloading the server with simultaneous requests. [21] This type of botnet tries to hide their traffic behind legitimate network ac-



(a) IRC Botnet (b) HTTP Botnet

Fig. 1. Botnets Topologies (Retrieved from http://securityaffairs.co/wordpress/13747/cyber-crime/http-botnets-the-dark-side-of-an-standard-protocol.html)

tivity. Zeus[1] and SpyEye[2], are examples of HTTP botnets, dedicated to steal personal information, while other HTTP botnets essentially aim to send spam messages, like Rustock, which uses a custom encryption scheme on top of HTTP to conceal its C&C traffic.
Zeus was discovered in 2007 and is one of the most financially lucrative bots in history, the FBI estimates that to date this botnet may have stolen hundreds of millions of dollars. In 2011, the source code was leaked online, which has lead to an explosion os Zeus variants, because of this, Zeus infections and botnets continue to account for a large number of global botnet installations.[9] SpyEye was developed in late 2009 and its purpose was identical to that of Zeus: financial theft. When the source code to Zeus was given to the author of SpyEye in 2010, it was assumed that newer versions of SpyEye would be built into a newer and stronger piece of malware. SpyEye remains a major source of malware infections online today. Rutstock was first identified in 2006 and it was a major spam generator, capable of sending up to 25000 spam emails per hour. [9]

2.3 Lifecycle

Botnet starts her lifecycle when a vulnerability in an operating system or software are exploited or users have been fooled to run unwanted software on their computer. Malware is often distributed as spam within a malicious attachments, spam linked to infected websites, open file shares, through instant messaging (IM) or by scanning after vulnerabilities. [1]
A typical lifecycle can be described in five steps: Creation, Infection, Rallying, Waiting and Executing. [21]
The first step is develop de bot software, after that, the botmaster infects the victim computer, once the victim machine becomes infected with the bot, it is known as a zombie. There are many ways to perform the initial infection, through software vulnerabilities, Drive-by download, email or pirated software [9]. In the first case the attacker exploits a vulnerability in a running service to automatically gain access and install the bot software without any user interaction. Drive-by download[3] refers to the unintentional download of a virus or malicious software (malware) onto your computer or mobile device, in this method the attacker hosts his file on a Web server and entices people to visit the site, when the user loads a certain page, the software is automatically installed without user interaction. Email attachment, is a method less popular lately, the attacker sends an attachment that will automatically install the bot software when the user opens it, usually without any interaction. Trojan Horse is another method used for initial infection, in this case the attacker bundles his malicious software with seemingly benign and useful software, the user is fully aware of the installation process, but he does not know about the hidden bot functionality.
After infection, the bot starts up for the first time and attempts to contact its

[1] http://www.fortiguard.com/legacy/analysis/zeusanalysis.html
[2] http://www.sciencedirect.com/science/article/pii/S1389128612002666
[3] https://blogs.mcafee.com/consumer/drive-by-download

C&C server(s) in a process known as Rallying. After joining the C&C network, the bot waits for commands from the botmaster, this step is called Waiting. During this time, very little (if any) traffic passes between the victim and the C&C servers. The last step, Executing, is when the bot receives commands from de botmaster, execute them and return the results via C&C network. In this step many activities can occur: the botmaster can send newer version of the malware or tell the bot to watch for certain patterns such as any attempts made to log into an online bank account, also execute other commands, such as recording online activities, sending spam emails, participating in denial of service attacks, and installing additional malware on the compromised system. After execute a command, the bot returns to the waiting state.
The botmaster must be able to counteract when its associated nodes leave the botnet. Such maintenance operations have a fundamental role in ensuring robustness, since only highly reliable and available botnet organizations can benefit from their highly collaborative features.

3 Attacks

After the infection the bots are used to carry out a variety of automated tasks such as sending, stealing, denial of service (DoS) and click fraud.
The sending includes send spam, viruses and spyware, stealing includes steal of personal an private information and communicate it back to the malicious user, such as credit card numbers, bank credentials and other sensitive personal information.
Send spam is one of the biggest activities performed by botnets, some of the largest botnets in history were responsible for sending out literally billions of messages a day.
Other type of attack is DDoS against a specified target, which is a malicious attempt to make a server or a network resource unavailable to users, usually by temporarily interrupting or suspending the services of a host connected to the Internet. Cybercriminals extort money from Web site owners, in exchange for regaining control of the compromised sites. The financial gain isn't the only motivation for this type of attack, sometimes is for the simple thrill of the botherder. In Click fraud cybercriminals use bots to boost Web advertising billings by automatically clicking on Internet ads. [26]
According to a technical report publish by the enterprise Fortinet [9], botnets are also used to Search Engine Optimization (SEO) poisoning, corporate and industrial espionage and bitcoin mining.
Botmasters perform the SEO poisoning boosting search engine rankings artificially to drive searchers to Websites that inject malware into a victims machine, or send the victim to sites that sell counterfeit goods or fake prescription drugs.
There is evidence that some botnets have been used in combination with targeted email attacks against both corporations and governments in the attempt to steal valuable intellectual property information and state secrets.

Bitcoin[4] is a virtual currency that can be traded anonymously online for products and services, botmasters are also capable of install bitcoin software on a victims PC and harness the processing power of that computer to mine coins and sell them on the grey or black market for real currency.

4 Detection, Defense and Evasion techniques

Botnet detection is perhaps one of the first actions that should be taken when combating network security threats. Given the potential power of botnets to conduct different malicious activities and cyber warfare, detection techniques play an important role in this process.[23]
Detection techniques can be classified into two main categories: those based on setting up honeynets and passive traffic monitoring. The last category has been further divided into other subcategories: signature based, anomaly-based and DNS based. Signature based regards the detection of known botnets, anomaly based detects botnets using known anomalies and DNS based analysis of DNS traffic generated by botnets. A honeynet[5] is a network of honeypots. The basic idea of a honeypot is to learn about attacker techniques by attracting attacks to a seemingly vulnerable host. A honeypot could be used to gain valuable information about attack methods used elsewhere or imminent attacks before they happen. Honeypots are used routinely in research and production environments[5].
Using network security tools such as IPS(Intrusion prevention system) or anti-malware doesn't guarantee protection against botnets, because botmasters use a range of techniques to evade existing methods of detecting and to prevent being disconnected, such as obfuscated binaries, encrypted C&C channels, fast-flux proxies protecting central C&C servers, customized comunication protocols. Centralized botnets have a single point a failure and became useless if the central entity is removed, to provide redundancy against this problem, botnets rely on dynamic DNS services, fast-flux DNS techniques or DGA (Domain Generation Algorithms). With fast-flux, bots query a certain domain that corresponds to a set of IP addresses that change frequently, while this makes it more difficult to take down or block a specific C&C server, the use of only one domain name constitutes a single point of failure. To fight this, botnets use domain flux instead, in which each bot independently uses a DGA to compute a list of domain names [25].
Botmasters can also use the capabilities offer by mobile phones to evade from detection. Most modern cell phones support text messaging services such as SMS and many smartphones also have full-featured IM(Instant Messaging) software, as a result, the botmaster can use a mobile device to control her botnet from any location with cell phone reception [5].
With these techniques the botmaster ensures they can stay a step ahead of any potential actions to shut down the C&C server.

[4] http://about-threats.trendmicro.com/us/definition/bitcoin/
[5] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.69.4398&rep=rep1&type=pdf

It is very difficult to find the presence of a bot, but there are some symptoms that can help us, such as: system running slower than usual, the hard drive LED is flashing wildly even though its in idle mode, files and folders have suddenly disappeared or have been changed in some fashion, a friend or colleague has informed the user that they have received a spam email from their email account, a firewall on the computer informs the user that a program on the PC is trying to connect to the Internet, a launch icon from a program downloaded from the Internet suddenly disappears, more error messages than usual are popping up and an online bank is suddenly asking for personal information its never required before [9].
Although it is difficult, there are some steps we can take to prevent infection: make sure all of the software used is obtained from a legitimate and trusted source and that all applications are fully patched with the most current updates, don't surfing on the Internet using an outdated Internet browser or using browser add-ons such as Adobe Flash or Oracles Java that aren't kept up-to-date, installing an antivirus software package and keeping it updated, get in the habit of setting up regular complete system scans, stick to one antivirus program, use a personal firewall program, and enable alerting whenever a program attempts to connect to the Internet.

5 Mobile Botnet

Mobile devices and apps are becoming ubiquitous to both personal and professional lives. According to Gartner, from the 1.875 billion mobile phones to be sold in 2013, 1 billion units will be smartphones, compared with 675 million units in 2012.[10]
With the growth in mobile devices, we also witnessed an increase of mobile malware, from March 2012 through March 2013, the total amount of malware the MtC (Mobile Threat Center) sampled across all mobile platforms grew 614 percent to 276,259 total malicious apps, compared with a 155 percent increase reported in 2011. [13]
Observing the trend of recent mobile malware, it is expected that mobile botnets will become a serious threat to smartphones soon. [12] The result could be the rise of toll fraud, SMS premium services and ransomware[6] as ways to generate cash.
Although botnets have not yet caused major outbreaks in the mobile world, with the rapidly-growing popularity of smartphones such as Apples iPhone and Android-based phones that store more personal data and gain more capabilities than earlier generation, but without adequate security and privacy protection, attacks targeting mobile devices are becoming more sophisticated and is expected that botnets become a severe threat to smartphones soon.[12]
This imminent threat attracted the attention of researchers and there are some Proof-of-Concept (POC) in this area who shows that the threat is real and measures should be taken.

[6] http://us.norton.com/ransomware

The term mobile botnet refers a collection of hijacked smartphones under the control of a botmaster, trough a C&C for malicious purposes.
Mobile botnets are different from traditional ones, the communication is mostly between mobile devices which have specific characteristics. The battery power is rather limited on smartphones compared with PCs, if battery power consumption speed exceeds user expectations, the battery exhaustion is likely to be noticed by the user, leaving the bot open to detection. The communication has costs, if data costs begin to exceed the amount that the user had expected or agreed to pay, the bot could also be detected, the same happens with an abnormal amount of network traffic. The lack of public IP address and a constant change in network connectivity makes the robust P2P-based C&C in PC-based botnets impractical, and potentially impossible, in smartphones. [28] The diversity of operating system on smartphones is also a factor that can make harder the creation of a mobile botnet.
The refered characteristics are an disadvantage, but others are an advantage such as the capability to send SMS. Mobile devices can send SMS to a predefined server device and delete them immediately. It is reported that SMS text messaging is the most widely used data application on the planet, with 2.4 billion active users, or 74% of all mobile phone subscribers sending and receiving text messages on their phones.[12] So, choosing SMS as C&C channel have advantages and is a viable solution for a mobile botnet. Not only is SMS ubiquitous to every mobile phone, but botmasters and bots are also able to disguise SMS messages, send bulk messages from the Internet at very low cost while hiding their identities. Thus, using SMS is both economical and efficient for the botnet.[12]
Although sending SMS messages through the cellular networks is always possible, the botmasters want to hide their identity and lower costs as much as possible. To achieve this goal, botmasters can use the Internet to disseminate C&C messages to the mobile botnet.
Until recently, mobile networks have been relatively isolated from the Internet, so there has been little need for protecting against botnets. However, this situation is rapidly changing. With the rapid development of the computing and Internet access (i.e., using WiFi, GPRS and 3G) capabilities of smartphones, constructing practical mobile botnets has become an underlying trend. [1]
In Section 3 was described the attacks performed by traditional botnets, but with the emergence of mobile botnets there are new attacks to consider. An infected mobile device can send MMS(Multimedia Messaging Service) or SMS to other mobile devices or to services number, so a mobile botnet will be able to send a short message to a voting service and not be detected as being sent by a botnet. Like traditional botnets, mobile ones can also retrieve sensitive information from the victims, which can be more meaningful considering the increase of personal data on mobile devices. There are some service which, the smartphones can give money to charity organizations. If the smartphone called or sent a text message to the specific service number, then the subscriber will pays a preset amount of money. The botmaster can create its own service number and programs all the bots to call or sent a text message to the specific service number. Of course, the

price should be low, so the subscribers would not notice and be suspicious about the extra charges. [11]

5.1 Examples

According to the 2013 Juniper Networks Third Annual Mobile Threats Report [13], one prominent example of a mobile botnet is the Tascudap, which was identified in December 2012. The Tascudap Trojan malware uses compromised devices as part of a botnet. It comes in an app package that mimics the icon used by the official Google Play store to trick users into clicking on the icon when they come across it on third-party application stores, other webpages or in phishing messages. If the user accidentally clicks on the fake Google Play icon, it will activate the malware. Once the malware has been activated, it will attempt to contact its C&C, registers the device's phone number and then waits for commands. Messages supported by the malicious application could allow the compromised device to begin to take part in a distributed denial-of-service attack, send SMS messages to premium rate numbers, and monitor incoming/outgoing SMS messages and Internet usage.
In December 2013 FireEye has uncovered MisoSMS, this mobile botnet has been used in at least 64 spyware campaigns, stealing text messages and emailing them to cybercriminals in China. MisoSMS infects Android systems by deploying a class of malicious Android apps. The mobile malware masquerades as an Android settings app used for administrative tasks. When executed, it secretly steals the users personal SMS messages and emails them to a C&C infrastructure hosted in China. [19]
Los Angels based cybercrime research company Intelcrawler made the discovery and described the threat as a mobile botnet named XXXX.apk. The botnet has been found on 23,856 compromised smartphones in all, including the HTC Sensation and Amaze 4G, the Google Nexus, Samsung GT I9300, Galaxy Note 2, LG Motion 4G, Huawei U8665 and the Alcatel One Touch.[2]

6 Social Botnet

Social networking sites are becoming more popular by the day, millions of people daily use social networking sites such as Facebook, LinkedIn, Myspace, Twitter and many more, Facebook is considered to be one of the most popular.
The structure of a social networking site is quite simple. Users register to the site, create their profile describing their interests and putting some personal information, and finally add friends/contacts to their profile. Adding a friend involves a confirmation step from the other party most of the times. The view of a users profile is usually limited to the friends of that user, unless the user wants the profile to be public. In that case, all users of the site can view it. Social networking sites also support the creation of groups and networks.
The increasing popularity of online social networks (ONS) has attracted botmasters attention, and recently begun to exploit social network websites to be

behave as their C&C infrastructures.[23]
ONS have some characteristics that make them attractive for botnet operations. First, the numerous computers that use them, second, the communication infrastructure of online social networks can be exploited as botnet C&C channels. Due to the huge number of messages that are delivered in online social networks, it is a daunting task to catch those that are used for botnet C&C. Many enterprise networks allow employees to visit online social networks during their work time but block P2P traffic at the enterprise gateways. Under such circumstance, botnets that use online social networks as their C&C channels can easily penetrate enterprise firewalls. [29]
There are studies that address this issue and show us that, like mobile botnets, it is possible to find some proof of concept that shows this threat is real and measures should be taken. In his master's theses A. Singh [24] built a botnet centered on Twitter, he used it to build the C&C botnet structure, in his project the botmaster sends commands to the bots using tweets, then as instructed in tweets, bots installed on the victims system would fetch those tweets and perform according to the code already installed on the victims system. One of the most important factor for the success of any botnet is the ability to hide their activity, the botnet proposed by A. Singh [24] the tweets posted by the botmaster, that are used to send instructions to the bots, don't look different in any perspective from other tweets and, hence, will not be treated as suspicious. This botnet is able to perform different types of attacks: browsing the web page, fetching the user information from Twitter profile in the form of a Twitter ID, and profile picture, stopping and restarting the services running at victims system, mailing confidential files to the botmaster, capturing information regarding users work and sending it to the botmaster and processing DOS commands by a bot within the victims system.
Through social networks, botmasters can make use of social engineering attacks (SEA), which continue to be an increasing attack vector for the propagation of malicious programs to spread bot programs and construct practical high-infection botnets more easily. [16]
Hackers use social media as covert channel to their advantage and are launching their C&C based attacks. The advantage they have at their disposal is that social media websites look innocent and are open to public; therefore, it is extremely difficult to filter out tweets that are harmful such as C&C compared to genuine and bogus spam that is not responsible for evil works.[24]
Given the popularity of both smartphones and ONS, it is only a matter of time before attackers exploit both to launch new types of attacks. In [8] M.R. Faghani proposes a new cellular botnet named SoCellBot that exploits OSNs to recruit bots and uses OSN messaging systems as communication channels between bots. The communication costs on mobile phone can be a problem when SMS is used for C&C channel and can make the botnet more vulnerable to detection, ONS can be a solution for this problem. Most cellular network providers offer OSN access to their clients free of charge, this makes OSN messaging systems a cost-effective solution for cellular bots to send and receive commands and control

messages[8]. There are other reasons for success of social botnets, messages exchanged in OSNs are usually encrypted, making it hard for cellular network providers to identify and block botnet messages and the topology of an OSN-based botnet is more resilient to bot failures or unavailability (compared with commonly seen botnets using on SMS) thanks to the highly clustered structure of the social network graph.
Besides the studies performed to prove that social botnets are a real threat, some attacks have been reported performed by this kind of botnet. In December 2013, it was reported that a botnet [6] stole two million logins and passwords for services such as Facebook, Google and Twitter.

6.1 Examples

The Facebook Socialbot Network, NazBot and Koobface were first examples of Socialbot Network (SbN). With on-line social networks growing and more users depending on these networks, more cases are expected to appear in the next years making OSN very attractive for botmasters.
The development of Facebook Socialbot Network [3] allowed to deepen the knowledge on social botnets. This SbN is a group of adaptive socialbots that are orchestrated in a command-and-control fashion. The results show that OSNs, such as Facebook, can be infiltrated with a success rate of up to 80%, depending on users's privacy settings, a successful infiltration can result in privacy breaches where even more users's data are exposed when compared to a purely public access, and in practice, OSN security defenses, such as the Facebook Immune System, are not effective enough in detecting or stopping a large-scale infiltration as it occurs.
NazBot[7] was accidentally found in 2009 by Jose Nazario from Arbor Networks, he found a bot that used Twitter as its C&C.
Koobface compromises user accounts on OSNs and uses these accounts to promote a provocative message with a hyperlink. The link points to a phishing website that asks the user to install a Flash plu-gin which is, in fact, the Koobface executable. Koobface evolved from a SbN that does not rely on hijacked profiles. Koobface thus requires infecting many initial zombie machines through OSN-independent distribution channels.[23]

7 Conclusions

Botnets become one of the most severe threats on the Internet, and are linked to most forms of Internet crime. They can perform a considerable number of severe attacks capable of compromising largely corporate networks, government websites among others, capable of causing severe damage. Is urgent to understand how botnets work, how they are evolving, the new trends so the battle is as efficient as possible. This knowledge is difficult to achieve because C&C

[7] http://profsandhu.com/confrnc/misconf/ACNS-2010.pdf

strategies and methods evolve rapidly.
The success of the botnets depends on their ability to hide and the robustness of communication between bot and botmaster, so the cybercriminals are always looking for new means to evade detection and always trying to improve their defense techniques, to accomplish this, botmasters are constantly looking for new communication platforms for delivering C&C information.
Lately we have witnessed an exponential growth of mobile devices and ONSs, these factors attracted the attention of cybercriminals and they started to exploit the vulnerabilities and realized that mobile devices and ONS can be used to build more robust and resilient botnets.

References

[1] Ahmed, R., Dharaskar, R., :Study of Mobile Botnets: An Analysis from the Perspective of Efficient Generalized Forensics Framework for Mobile Devices, International Journal of Computer Applications, (2012)

[2] Bell, L., (January 2014) http://www.theinquirer.net/inquirer/news/2322028/24-000-android-devices-are-hit-by-xxxxapk-mobile-botnet, [Online; accessed 20-January-2014]

[3] Boshmaf, Y., Muslukhov, I., Beznosov, K., Ripeanu, M., :The Socialbot Network: When Bots Socialize for Fame and Money, ACSAC '11 Proceedings of the 27th Annual Computer Security Applications Conference, pp. 93-102 , (2011)

[4] Castiglione, A., Prisco, R., Santis, A., Fiore, U., Palmieri, F., :A botnet-based command and control approach relying on swarm intelligence, Journal of Network and Computer Applications , (2013) http://www.sciencedirect.com/science/article/pii/S1084804513001161

[5] Chen, T., Walsh, P., Chapter 5 - Guarding Against Network Intrusions, Computer and Information Security Handbook (Second Edition), pp. 81-95, Second Edition, (2013)

[6] Computer World (December 2013) http://www.computerworld.com.pt/2013/12/04/passwords-roubadas-de-facebook-google-ou-twitter/ [Online; accessed 24-January-2014]

[7] Correia, P., Rocha, E., Nogueira, A., Salvador, P., :Statistical Characterization of the Botnets C&C Traffic, Elsevier - Procedia Technology, (2012)

[8] Faghani, M., :SocellBot: A new botnet desingn to infect smartphones via social online networking, 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), (2012)

[9] Fortinet, :Anatomy of a Botnet, (2013) http://www.fortinet.com/sites/default/files/whitepapers/Anatomy-of-a-Botnet-WP.pdf

[10] Gartner (April 2013), http://www.gartner.com/newsroom/id/2408515, [Online; accessed 24- January-2014]

[11] Geng G., Xu G., Zhang, M., Guo, Y., :The Design of SMS Based Heterogeneous Mobile Botnet, Journal of Computers, (2012)

[12] Hua, J., Sakurai, K., :Botnet command and control based on Short Message Service and human mobility, Elsevier, pp. 579–597, (2012) http://www.sciencedirect.com/science/article/pii/S1389128612002162

[13] Juniper Networks , :Third Annual Mobile Threats Report, (2013) http://www.juniper.net/us/en/local/pdf/additional-resources/jnpr-2012-mobile-threats-report.pdf

[14] Kartaltepe, E., Morales J., Xu, S., Sandhu, R., :Social network-based botnet command-and-control: emerging threats and countermeasures, ACNS'10 Proceedings of the 8th international conference on Applied cryptography and network security, (2010) http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.185.4032&rep=rep1&type=pdf

[15] Krasser, S., Grizzard, J., Krasser, H., Grizzard, J., Owen, H., The Use of Honeynets to Increase Computer Network Security and User Awareness, Journal of Security Education, pp. 22-37, (2005)

[16] Li, S., Yun, X., Hao, Z., Cui, X., :A Propagation Model for Social Engineering Botnets in Social Networks, 12th International Conference on Parallel and Distributed Computing, Applications and Technologies, (2011)

[17] Li, C., Jiang, W., Zou, X., :Botnet: Survey and Case Study, Fourth International Conference on Innovative Computing, Information and Control, pp. 1184-1187, (2009)

[18] Nagaraja, S., Houmansadr, A., Piyawongwisal, P., Singh, V., Agarwal, P., Borisov, N., :Stegobot: a covert social network botnet, IH'11 Proceedings of the 13th international conference on Information hiding, (2011) http://www.cs.utexas.edu/ amir/papers/IH11-Stegobot.pdf

[19] Pidathala, V., Dharmdasani, H., Zhai, J., Bu, Z., (December 2013) http://www.fireeye.com/blog/technical/botnet-activities-research/2013/12/misosms.html, [Online; accessed 20- January-2014]

[20] Plohmann, D., :Botnets: 10 Tough Questions., European Network and Information Security Agency (ENISA), (2011) http://www.enisa.europa.eu/activities/Resilience-and-CIIP/critical-applications/botnets/botnets-10-tough-questions

[21] Ramsbrock, D., Wang, X., Chapter 12 - The Botnet Problem, Computer and Information Security Handbook (Second Edition) , pp. 223-238, Second Edition, (2013)

[22] Salles, R., :Editorial for Computer Networks special issue on "Botnet Activity: Analysis, Detection and Shutdown", Elsevier, pp. 375-377, (2012) http://www.sciencedirect.com/science/article/pii/S1389128612003076

[23] Silva, S., Silva, R., Pinto, R., Salles, R., :Botnets: A survey., Elsevier, (2012), http://www.sciencedirect.com/science/article/pii/S1389128612003568

[24] Singh, A., :Social Networking for Botnet Command and Control, The Faculty of the Department of Computer Science San Jose State University (2012).

[25] Stone-GroSS, B., Cova, M., GilBert, B., KeMMerer, R., KrueGel, C., viGna, G., Analysis of a Botnet Takeover, IEEE Computer and Realiability Societies, pp. 64-72, (2011)

[26] Symantec, http://us.norton.com/botnet/, [Online; accessed 21- January-2014]

[27] Tanner, B., Warner, G., Stern H., Olechowski, S., :Koobface: the Evolution of the Social Botnet, eCrime Researchers Summit (eCrime), (2010) http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5706694

[28] Xiang, C., Andbot: Towards Advanced Mobile Botnets, LEET'11 Proceedings of the 4th USENIX conference on Large-scale exploits and emergent threats, 2011

[29] Yan, G., :Peri-Watchdog: Hunting for hidden botnets in the periphery of online social networks, Elsevier, (2013) http://www.sciencedirect.com/science/article/pii/S1389128612002903

[30] Zeng, Y., Shin, K., Hu, X., :Design of SMS Commanded-and-Controlled and P2P-Structured Mobile Botnets, WISEC '12 Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks, pp. 137-148, (2012), http://dl.acm.org/citation.cfm?id=2185467&preflayout=tabs

Anotação Gramatical (POS-Tagging): Ferramentas, recursos, abordagens e avaliação

João Sequeira
d11594@alunos.uevora.pt

Universidade de Évora

Resumo Devido à enorme quantidade de informação em formato textual presente no mundo digital atualmente, o Processamento de Língua Natural tornou-se, nas últimas décadas, uma área de crescente interesse. Os sistemas para anotar ou extrair informação de conteúdos textuais têm sido amplamente investigados, permitindo assim resolver tarefas que manualmente são impossíveis de concretizar. Este artigo dá uma visão geral de vários aspetos ligados ao Processamento de Língua Natural. É abordada a anotação linguística de recursos onde é feita uma introdução à teoria usada como base e são apresentados alguns recursos existentes tanto para o Inglês como para o Português. Na anotação automática de textos são apresentados os tipos de abordagem (supervisionada e não supervisionada), algoritmos e como é efetuada a avaliação de desempenho dos sistemas. Por fim é descrita a tarefa da anotação gramatical (POS-Tagging), apresentando uma base teórica, um conjunto de ferramentas (para o Português, para o Inglês e multi-idiomas) e uma exposição sobre alguns problemas que ainda podem ser encontrados na área.

1 Introdução

Atualmente existe uma grande quantidade de conteúdos digitais de cariz académico, pessoal e noticioso, entre outros, disponíveis para consulta na Internet. A tarefa de obter informação, manualmente, de conteúdos não tratados, de fontes tão distintas, tornou-se praticamente impossível [1].

Em Julho de 2012, o número de utilizadores da Internet ascendiam aos 2,4 mil milhões no mundo, mais 295 milhões do que em 2011 [2]. O aumento da utilização da Internet deveu-se em muito à expansão da componente social: (i) redes sociais sejam para uso lúdico, por exemplo o Facebook[1], ou profissional, como por exemplo o LinkedIn[2], (ii) partilha de conteúdos, entre os vários exemplos pode-se enumerar a partilha de vídeos através da página Youtube[3].

Com o número elevado de conteúdos textuais no mundo digital apareceu a necessidade de implementar sistemas que consigam extrair informação dos mesmos. Assim, nos últimos anos tem existido uma crescente evolução na pesquisa e implementação de aplicações de Processamento de Língua Natural [4] (PLN) [3]. Existem diferentes áreas de pesquisa dentro do PLN como a anotação de argumentos sintático [4], a identificação e anotação de entidades mencionadas[5] [1,5] ou as tarefas avaliadas em diferentes conferências sobre subtemas associados ao PLN, um exemplo é a PAN 2013[6]: identificação de autores, identificação de plagiarismo e identificação de perfis de autores [6]. Outro exemplo é a Conferência sobre Aprendizagem de Língua Natural[7] realizada anualmente pelo Grupo de Interesse Especial em Aprendizagem de Língua Natural[8]. Esta última

[1] https://www.facebook.com/
[2] https://www.linkedin.com/
[3] http://www.youtube.com/
[4] Do inglês, *Natural Language Processing (NLP).*
[5] Do Inglês, *Named Entity Recognition.*
[6] Evento: (http://pan.webis.de/), realizado no âmbito da conferência CLEF 2013, http://www.clef2013.org/
[7] Do Inglês, *Conference on Natural Language Learning*, com a sigla CoNLL. Página dos eventos http://ifarm.nl/signll/conll/
[8] Do Inglês, *Special Interest Group on Natural Language Learning*, com a sigla SIGNLL.

existe desde 1997 e entre os seus temas já abordou análise de dependências[9] [7,8], anotação de papeis semânticos[10] [3] e anotação de entidades mencionadas [9].

Este artigo apresenta algumas das bases do PLN e aos poucos vai-se focando num dos subtemas denominado anotação de classes gramaticais[11]. O artigo possui mais 4 secções: na Secção 2 é abordada a anotação linguística, onde são expostos a sua componente teórica e os recursos que podem ser utilizados em sistemas de anotação automática; na Secção 3 é abordada a anotação automática de textos, onde são expostos os tipos de abordagem, exemplos de algoritmos e como são avaliados os sistemas de PLN; na Secção 4 é abordada a anotação gramatical de textos e os problemas que ainda persistem na área; na secção 5 é exposta a conclusão.

2 Anotação Linguística

A anotação linguística de recursos já existe há vários anos mas foi nos últimos, com o crescimento das aplicações PLN, que começou a possuir uma influência crucial [10]. A anotação de *corpora*[12] foi uma área que recebeu uma atenção especial, visto ser uma das bases dos sistemas de anotação. Usando as melhores práticas de modo a desenvolver recursos com qualidade e com conjuntos de anotações específicas para cada tarefa [10]. A anotação linguística permitiu criar uma ligação entre o estudo de textos, a linguística, e as tarefas que os sistemas automáticos realizam sobre textos. É através da anotação linguística de textos que se obtêm recursos textuais anotados (denominados por *corpora*).

Na Secção 2.1 será abordada a teoria da linguística e como esta está interligada com as diferentes tarefas de anotação. Na Secção 2.2 serão abordados alguns dos recursos disponíveis para as tarefas de anotação gramatical tanto para o Inglês como para o Português.

2.1 Linguística

A linguística possui sete níveis na sua descrição teórica: seis referentes à linguagem escrita e um referente à linguagem falada [11,12,13,14]:

- **Estudo das unidades mais pequenas (sons e letras)**:
  - **Fonologia**: Abrange o estudo dos sons individuais, ou seja, as unidades mais pequenas da linguagem falada.
  - **Ortografia**: Abrange o estudo das letras individualmente.
- **Estudo das unidades de tamanho médio (palavras, sintagmas e frases)**:
  - **Morfologia**: Abrange o estudo da criação e inflexão das palavras a partir das unidades mais básicas.
  - **Sintaxe**: Abrange o estudo da ordenação das palavras e da estrutura da frase, ou seja, qual o papel e posição de cada palavra na frase.
  - **Semântica**: Abrange o estudo do significado das palavras e das frases, ou seja, como conjugar os vários significados das palavras de modo a criar uma frase com sentido.
- **Estudo das unidades maiores (parágrafos e diálogos)**:
  - **Pragmática**: Abrange o estudo da função das frases, ou seja, a função da frase consoante a situação em que é empregada e como afectam a percepção do discurso.
  - **Discurso**: Abrange o estudo da coesão das diferentes frases num texto ou discurso, ou seja, como as frases anteriores afectam a frase actual.

Em anotação linguística é também possível separar as diferentes tarefas de anotação em níveis, como na componente teórica da linguística [11]. Visto este trabalho apenas focar a componente escrita da linguagem não será considerada a componente relativa aos sons, ou seja, a fonologia. Graham et al. em [11] indica que não é possível realizar uma correspondência perfeita entre a componente teórica e as tarefas práticas de anotação de *corpora*. Mas é possível realizar uma correspondência aproximada:

[9] Do Inglês, *Dependency Parsing*.
[10] Do Inglês, *Semantic Role Labeling*.
[11] Do Inglês *Part-of-Speech Tagging (POS Tagging)*.
[12] Recursos anotados para uso de aplicações PLN.

- **Ortografia** (por norma as tarefas deste nível são as primeiras a serem executadas de modo a prepararem o texto para as tarefas de níveis posteriores):
  - ***Tokenization*:** É a tarefa de segmentar o texto em *tokens* distintos, por norma em palavras.
  - **Deteção dos limites das frases[13]:** É a tarefa de segmentar o texto em frases distintas.
- **Morfologia** (utiliza as palavras obtidas pela toquenização):
  - **Anotação de classes gramaticais (POS-Tagging):** Utiliza as palavras e a forma como estas são compostas para lhes atribuir categorias gramaticais. Utilizando os morfemas (unidade mínima de uma palavra) é possível verificar se uma palavra é um nome, um verbo, um adjetivo, um pronome, entre outros [14].
- **Sintaxe** (utiliza as frases obtidas pela deteção dos limites das frases):
  - **Análise sintática:** Classifica os sintagmas que constituem uma frase, ou seja, anota o sintagma verbal (o predicado), sintagma nominal (tem por núcleo o nome), entre outros [14].
- **Semântica, Pragmática e Discurso** (tarefas associadas a este nível são vistas como pertencentes ao nível superior da linguística):
  - **Anotação de entidades mencionadas:** É a tarefa de identificar e anotar entidades (classes normalmente usadas: Pessoa, Organização, Local) no texto.
  - **Resolução de correferência:** É a tarefa de identificar num texto, referências pertencentes à mesma entidade.

2.2 Recursos

Em anotação linguística, consideram-se recursos, os *corpora* anotados usados nas diferentes tarefas de anotação automática. De seguida vão ser abordados dois *corpora* para a língua inglesa, *Brown Corpus* e *Penn Treebank*, e para a língua portuguesa o Bosque 8.0 e o LABEL-LEX.

Brown Corpus: foi o primeiro grande corpus criado para a língua inglesa, foi implementado, nos anos 60, por Francis e Kucera na Universidade de Brown, nos Estados Unidos [15,16]. Possui sensivelmente 500 exemplos de textos em Inglês obtidos de trabalhos publicados, escolhidos aleatoriamente, totalizando um milhão de palavras. Cada exemplo é composto por uma frase completa de 2000 ou mais palavras. Inicialmente o corpus era composto apenas pelas palavras, só posteriormente é que Greene e Rubin criaram, nos anos 70, uma aplicação para realizar a anotação das classes gramaticais. A aplicação consistia numa lista de regras, com as hipóteses que podiam coexistir. Foi obtido um resultado de aproximadamente 70% de acertos, os restantes 30% foram revistos manualmente [17]. No Brown Corpus foi considerado um número bastante elevado de etiquetas gramaticais (superior a 80), sem contar com combinações de etiquetas. A totalidade das etiquetas (singulares e complexas) podem ser visualizadas em [18,17,19].

Penn Treebank: é o corpus e respectivo conjunto de etiquetas mais populares e usados em tarefas de anotação gramatical. O projeto *Penn Treebank* [20,21] foi criado na Universidade da Pensilvânia [11]. Este projeto foi iniciado em 1989 e oito anos depois possuía 7 milhões de palavras classificadas com etiquetas gramaticais, 2 milhões de palavras em estruturas predicado-argumento, entre outros formatos [22]. Os textos etiquetados foram retirados de fontes tão diversas como manuais de computadores da IBM, artigos do *Wall Street Journal* e conversas telefónicas transcritas [22]. O conjunto de etiquetas gramaticais utilizado neste projecto, pode ser visto em Wilcock et al. [11] e resultou do seguinte processamento sobre o conjunto de etiquetas do corpus *Brown* [22]:

1. **Eliminação da redundância lexical e sintáctica:** Devido à existência de várias etiquetas que são únicas para um item lexical particular (por exemplo, uma etiqueta para a palavra *have* e outra para *be*), foi efetuada a eliminação destas instâncias lexicais redundantes.

[13] Do Inglês, *Sentence Boundary Detection*.

2. **Preocupação com a importância do contexto sintáctico:** No corpus *Brown* as palavras foram classificadas independentemente da sua função sintática. No *Penn Treebank* a anotação de uma palavra teve em conta a sua função sintática dentro da frase.
3. **Evitar que os classificadores tomem decisões arbitrárias:** É possível uma palavra possuir mais do que uma etiqueta gramatical, tentando mostrar que a etiqueta não pode ser decidida ou o classificador possui incerteza em qual a etiqueta correcta.

Bosque 8.0: está incorporado no projeto Floresta Sintá(c)tica[14] e nasceu da necessidade de disponibilizar à comunidade de língua Portuguesa um recurso essencial para ferramentas de PLN [23]. A Floresta Sintá(c)tica consiste em texto corrido, dividido em frases analisadas sintáticamente em estruturas de árvore pelo analisador sintático PALAVRAS [24]. A sua revisão foi feita utilizando aplicações automáticas e revisão manual [23]. O Bosque 8.0 é composto por 9368 frases dos primeiros 1000 extratos, totalmente revistos por linguistas, do CETEMPúblico e do CETEMFolha [25]. O CETEMPúblico, utiliza Português Europeu e foi criado com notícias do jornal Público [26]; o CETEMFolha utiliza Português do Brasil e foi criado com notícias do jornal Folha de S. Paulo [27,28].

Na Tabela 1 pode-se ver as etiquetas das classes utilizadas no Bosque 8.0.

Tabela 1. Conjunto de etiquetas gramaticais do Bosque 8.0. (Fonte: [25])

Etiqueta	Descrição
s	Substantivo
n-adj	Substantivo/Adjectivo
adj	Adjectivo
prop	Nome próprio
adv	Advérbio
v-fin	Verbo finito
v-ger	Verbo gerúndio
v-pcp	Verbo particípio
v-inf	Verbo infinitivo
art	Artigo
pron-det	Pronome determinativo
pron-rel	Pronome independente
pron-pess	Pronome pessoal
adv	Advérbio
prp	Preposição
intj	Interjeição
conj-s	Conjunção subordinativa
conj-c	Conjunção coordenativa
ec	Prefixos
x	Palavra estrangeira

LABEL-LEX: outro recurso linguístico para o Português que podemos considerar é o LABEL-LEX[15]. O LabEL[16] faz parte do Centro de Estudos da Linguagem da Faculdade de Letras da Universidade de Lisboa e desenvolve recursos linguísticos como gramáticas, léxicos e dicionários aplicados ao Português. O LABEL-LEX possui três módulos: (i) um composto por lemas; (ii) um de unidades lexicais multi-palavra, nomes e advérbios; (iii) um de gramáticas de restrições sintáticas. Apesar dos módulos apresentados deste projeto serem recursos linguísticos, não estão na mesma forma dos *corpora* apresentados antes, mas continuam a conter informação útil para

[14] Disponível em http://www.linguateca.pt.
[15] Página do projecto em http://label.ist.utl.pt/pt/apresentacao.php.
[16] Sigla de Laboratório de Engenharia de Linguagem

tarefas de anotação automática. O LABEL-LEX não será explorado em maior pormenor, estando na página do projeto a informação necessária para uma pesquisa mais aprofundada.

3 Anotação Automática

A anotação automática de textos consiste em criar aplicações que anotam textos em língua natural, como um todo (por exemplo classificar se um e-mail é *spam*[17] ou não), ou partes dele, como as palavras todas (exemplo disso é a anotação gramatical, falada neste trabalho) ou apenas algumas como na anotação de entidades mencionadas. Estas aplicações realizam a tarefa de atribuir classes de um conjunto pré-definido a textos em língua natural. Os conjuntos de classes dependem das tarefas para as quais os anotadores foram criados: (i) na anotação gramatical as classes são as categorias gramaticais; (ii) na anotação de entidades mencionadas as classes são dividas em locais, pessoas, organizações, entre outras.

A anotação automática de textos apareceu nos anos 60, mas foi nos anos 90 que começou a ser explorada de forma mais intensa, em parte, devido à disponibilidade de hardware mais rápido e com maior capacidade de processamento, permitindo assim "atacar"tarefas complexas com maiores cargas de dados [14].

Na Secção 3.1 serão abrangidos os tipos de abordagem ao problema da anotação automática consoante o tipo de aprendizagem dos sistemas (supervisionada e não supervisionada). Na Secção 3.2 serão abordados quatro algoritmos usados em tarefas de anotação automática. Na Secção 3.3 será abordada a medição do desempenho de sistemas de anotação automática e os tipos de erros que a influenciam.

3.1 Tipos de abordagem

É possível dividir os tipos de abordagem pela forma como os sistemas realizam o seu treino dos dados. Este processo é denominado por aprendizagem automática. As vantagens da aplicação da aprendizagem automática são [14]:

1. Obtenção de resultados similares aos obtidos por peritos na área da tarefa.
2. Poupança de tempo e de potencial humano.
3. Não são necessárias intervenções humanas na construção ou alteração dos anotadores.

Os dois tipos de sistemas conhecidos divergem no tipo de aprendizagem utilizada (supervisionada e não supervisionada) [29,30]:

- **Sistemas de aprendizagem supervisionada:** são fornecidos dados classificados, *corpora*, para o sistema realizar o treino através de exemplos, de modo a criar um modelo. Esse modelo será aplicado na anotação automática de novos dados.
- **Sistemas de aprendizagem não supervisionada:** não são fornecidos dados anotados para o sistema inferir a partir das classes lá dispostas. Estes sistemas realizam a sua aprendizagem por meio da criação de padrões, presentes nos dados, de modo a serem usados quando são confrontados com dados novos. Uma das técnicas mais comuns é o agrupamento[18] [31].

No resto deste artigo, das duas abordagens enumeradas, será dado destaque apenas aos sistemas de aprendizagem supervisionada e às suas características.

3.2 Algoritmos

Nesta secção serão abordados quatro algoritmos usados para realizar o treino de anotadores de PLN, remetendo para a respectiva bibliografia explicações mais aprofundadas:

[17] Do Inglês, *Sending and Posting Advertisement in Mass* com a sigla spam.
[18] Do Inglês, *clustering*.

- **Modelos Ocultos de Markov (MOM)**[19] **[32,33,34]:** Os modelos ocultos de Markov começaram a ser abordados nos anos 60 por Leonard E. Baum e os seus colaboradores, e ao longo dos anos foram aplicados a áreas como o reconhecimento da fala, desambiguação de classes gramaticais e bioinformática. Cada modelo é um autómato finito com observações e transições de estado estocásticas. Um autómato começa num determinado estado onde utiliza um processo probabilístico para gerar uma sequência de observações e emite uma observação para esse estado, transita para um novo estado e emite outra observação, seguindo sempre o mesmo processo até atingir o estado final. Um modelo oculto de Markov é dado por uma distribuição de probabilidade do estado inicial, um conjunto finito de estados, um conjunto de possíveis observações e duas distribuições de probabilidade condicionada, uma aplicada às transições entre cada dois estados e outra aplicada às observações emitidas por cada estado. Para a área da anotação automática, cada exemplo nos dados de treino é composto por uma sequência de etiquetas ligada a uma sequência de observações. A cada nova observação o objetivo é atribuir a sequência de etiquetas mais provável.
- **Modelos Condicionais de Markov (MCM)**[20] **[35,36,30]:** Os modelos condicionais de Markov, podem também ser denominados de modelos de máxima entropia de Markov e são baseados nos MOM. Tal como nos MOM cada modelo é um autómato finito com uma distribuição de probabilidade para o estado inicial, um conjunto finito de estados, um conjunto de possíveis observações e uma probabilidade para a transição. A diferença entre os MCM e MOM reside implementação das distribuições de probabilidade. Nos MOM é realizada a transição para o estado atual e posteriormente é emitida uma observação. Já os MCM não possuem duas distribuições de probabilidade separadas (de transição entre estados e de observação), mas apenas uma, onde a transição para o estado atual está condicionada ao estado anterior e à observação atual.
- **Campos Condicionais Aleatórios (CRF)**[21] **[37,38,30]:** Campos condicionais aleatórios, têm como base os MCM, são modelos gráficos não direccionados. Cada modelo define uma distribuição de probabilidade conjunta de sequências de etiquetas dado um conjunto de sequências de observações. Um CRF no momento de aprendizagem e anotação de um exemplo toma sempre em atenção as anotações efetuadas na sua vizinhança. São principalmente usados em análise de informação sequencial em aplicações linguísticas, como o reconhecimento de entidades, e também são usados para efetuar o reconhecimento de padrões usado em visão computacional.
- **Máquinas de Vectores de Suporte (SVM)**[22] **[39,40,30]:** As máquinas de vectores de suporte são classificadores lineares que tentam maximizar sempre o hiperplano[23] que permite separar os dados sem apresentar erros de anotação. As SVM permitem criar anotadores que nem são muito simples com tendência a cometer vários erros nem são muito rígidos e complexos que tendem a ter dificuldades em generalizar para novos dados. As SVM possuem características, como a sua boa capacidade de generalização e a sua robustez face a grandes quantidades de informação, que as torna como um dos algoritmos amplamente usado em tarefas de anotação automática.

3.3 Avaliação dos sistemas

Em PLN cada sistema necessita de ser avaliado, para, ao se perceber os erros que comete serem efetuados melhoramentos. Nesta secção é abordado o método mais usual de avaliação de sistemas de anotação automática e que tipos de erros podem influenciar essa avaliação.

A avaliação dos sistemas de anotação automática pode ser realizada através de [29]:

[19] Do Inglês, *Hidden Markov Models* com a sigla HMM.
[20] Do Inglês, *Conditional Markov Models* com a sigla CMM.
[21] Do Inglês, *Conditional Random Fields*.
[22] Do Inglês, *Support Vector Machines*.
[23] Um hiperplano é definido por uma função matemática. Serve para separar os exemplos positivos dos negativos.

– **Eficiência:** mede o tempo que o sistema necessita para concluir tarefas, por exemplo na construção do modelo ou na anotação de novos exemplos. A implementação do sistema e o tamanho do corpus poderá afetar os resultados.
– **Medidas do seu desempenho:** as medidas mais usuais são a precisão (π), cobertura (ρ) e medida F_β para cada classe do conjunto. Sendo calculadas posteriormente as médias dessas medidas no conjunto das classes.

Por norma são utilizadas as medidas de desempenho para analisar o resultado da anotação automática. É criada uma matriz de confusão onde se considera positiva (+) a classe em estudo e negativa (−) as restantes, e onde é realizada a comparação entre as classes a que os exemplos pertencem (classe correcta) e aquelas que o sistema classificou (classe prevista) [30].

A Tabela 2 mostra como é disposta a informação numa matriz de confusão [30]:

– **VP (nº de verdadeiros positivos):** exemplos em que a classe + é atribuída corretamente, ou seja, a classe prevista é igual à correta e o valor é positivo.
– **FP (nº de falsos positivos):** exemplos em que a classe + é atribuída incorretamente, ou seja, a classe prevista é classificada como positiva mas a correta é negativa.
– **FN (nº de falsos negativos):** exemplos que não foram classificados como + mas que deveriam ter sido, ou seja, a classe foi prevista como negativa quando a correta é positiva.
– **VN (nº de verdadeiros negativos):** exemplos em que a classe − foi atribuída corretamente, ou seja, ambas as classes são iguais e negativas.

Tabela 2. Matriz de confusão para as classes positiva (+) e negativa (−). Fonte: [30]

		Correcta	
		+	−
Prevista	+	VP	FP
	−	FN	VN

A matriz de confusão representada na Tabela 2, é o resultado da anotação prevista por um sistema comparada com a anotação feita por linguistas, considerada como correta, denominado por *Gold Standard* [41].

As anotações utilizadas para medir o desempenho podem ser de dois tipos: (i) anotação *per-token* onde cada *token* é anotado individualmente. Um exemplo deste tipo, é a anotação gramatical onde normalmente cada palavra é vista como um *token*, apesar de poderem ocorrer algumas exceções de palavras compostas; (ii) anotação *multi-token* onde um conjunto de *tokens* é anotado com a mesma etiqueta. Um exemplo deste tipo, é a anotação de entidades mencionadas onde podem ocorrer Nomes, Locais e Organizações com nomes compostos por vários *tokens*.

Os tipos de anotações incorretas que podem ocorrer aquando da comparação do resultado de um sistema com o respetivo *Gold Standard* são [41]:

– O sistema faz uma anotação onde não devia existir.
– O sistema falha uma anotação.
– O sistema faz a marcação certa dos limites mas erra na classe atribuída.
– O sistema atribui a classe certa mas falha na marcação dos limites.
– O sistema falha tanto na atribuição da classe como na marcação dos limites.

Após a construção da matriz de confusão do resultado de uma tarefa, é possível verificar o desempenho calculando a precisão, cobertura e medida F_β. A precisão é a proporção de exemplos classificados por um sistema e que estão corretos [42]. O valor da precisão é calculado pela expressão [30]:

$$\pi = \frac{VP}{VP + FP} \tag{1}$$

A cobertura é a proporção de exemplos corretos que são classificados por um sistema [42]. O valor da cobertura é calculado pela expressão [30]:

$$\rho = \frac{VP}{VP + FN} \tag{2}$$

A medida F_β calcula a média harmónica entre a precisão e a cobertura tornando-se, normalmente, a medida final para comparar sistemas [42] e é dada pela expressão, onde normalmente é utilizado o valor $\beta = 1$:

$$F_\beta = (1 + \beta^2) * \frac{\rho * \pi}{\beta^2 * \rho + \pi} \tag{3}$$

4 Anotação Gramatical (POS-Tagging)

A anotação gramatical consiste em atribuir classes gramaticais a *tokens*, neste caso, às palavras de um texto [11]. A anotação gramatical pode ser encarada como produto final ou como um passo intermédio para outra tarefa em que sejam necessárias características gramaticais, como exemplo, temos a anotação de entidades mencionadas [1,5], a anotação de argumentos sintáticos [4] e a extracção de relações entre entidades mencionadas [30].

A anotação gramatical caracteriza-se por ter as seguintes abordagens [30]: (i) linguística, onde são usadas gramáticas de regras; (ii) orientada a dados, onde são utilizados *corpora* classificados; (iii) híbrida, com numa fase orientada a dados e outra linguística para alguns casos especiais.

Na Figura 1 é possível visualizar a anotação gramatical da frase "Vera apagou a luz.".

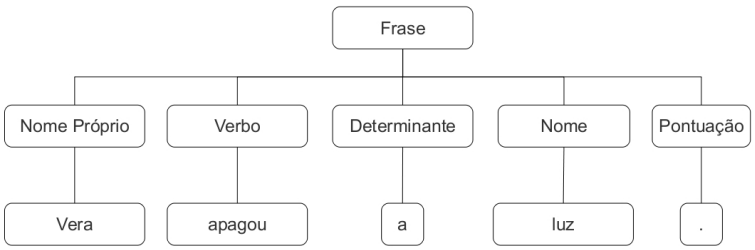


Figura 1. Anotação gramatical da frase `'Vera apagou a luz.'`. Fonte: [30]

Voutilainem em [43] define a anotação gramatical num processo com dois passos: (i) anotação das palavras com ambiguidade; (ii) eliminação da ambiguidade, removendo as hipóteses incorretas.

Generalizando a tarefa de anotação gramatical, para a abordagem orientada a dados, pode-se definir os seguintes passos:

1. Obter um corpus, como os exemplos da Secção 2.2.
2. Utilizar um algoritmo para treinar um modelo a partir do corpus.
3. Utilizar o anotador em conjunto com o modelo para realizar a anotação automática de novos textos.

Atualmente, sistemas com valores de precisão *per-token* a rondar os 97% são considerados estado da arte, visto nos últimos 18 anos o melhoramento nos valores de precisão ter sido inferior a 1%, passando de 96,63% para 97,24% [30].

Na Secção 4.1 são abordadas várias ferramentas de anotação gramatical, para o Inglês, para o Português e multi-idiomas dependendo dos dados usados no seu treino. Na Secção 4.2 são abordados aspetos que podiam ser alterados na avaliação do desempenho dos anotadores e erros que podem estar na origem dos anotadores não conseguirem passar a barreira dos 97% de precisão.

4.1 Trabalho Relacionado

Sendo o Inglês o idioma mais falado no mundo torna-se claro, o porquê, dos primeiros *corpora* (como o corpus Brown e posteriormente o PennTreebank) e ferramentas criadas tenham sido exclusivamente implementadas para processar textos escritos nesse idioma. Foi só nos últimos anos que se começou a abordar outros idiomas, principalmente em conferências, o que proporcionou também o desenvolvimento de ferramentas multi-língua ou específicas para esses idiomas. O Português não foi exceção nesse desenvolvimento.

Graham et al. [11] apresentam duas ferramentas, que entre as suas capacidades podem efetuar a anotação gramatical, numa delas pode-se trabalhar com o Português:

- **Wordfreak**[24] **[44]:** ferramenta de anotação de língua natural implementada em Java. Ao ser utilizada uma linguagem orientada a objetos, é possível que o seu código seja reutilizado para criar novos componentes. O Wordfreak não funciona apenas com a língua inglesa, também realiza anotações na língua chinesa e árabe. Permite realizar tarefas como deteção de limites de frases, *tokenization* ou anotação gramatical.
- **OpenNLP**[25]**:** é um conjunto de bibliotecas estatísticas baseadas em modelos de máxima entropia (um exemplo, são os MCM) para processamento de língua natural e foi implementado em Java. Entre as tarefas que realiza estão a anotação gramatical, *tokenization* e identificação de nomes. Através da página do projeto é possível aceder a um conjunto de modelos onde está incluído o Português, treinado com o Bosque (versão 7.3), utilizado na conferência CoNNL do ano 2006 [7].

Para o Português, tanto para a versão europeia como para a do Brasil, a Linguateca mantém um repositório de *corpora* classificados, tesauros (podem ser encarados como dicionários, visto serem listas de palavras com significados similares ou termos relativos a uma ideia ou domínio comum) e ferramentas de PLN para diversas tarefas.

Em [30] são citadas aplicações para o Português que permitem realizar anotação gramatical, apenas o SVMTool não está presente na página da Linguateca:

- **Português do Brasil:**
  - **Aelius**[26]**:** é um analisador morfossintáctico para português do Brasil implementado em Python.
  - **Curupira**[27]**:** é um analisador que utiliza uma gramática funcional livre de contexto com restrições [45,46].
- **Português Europeu:**
  - **Tree-Tagger [47,48] para Português e Galego**[28]**:** um analisador implementado utilizando o Tree-Tagger por Pablo Gamallo da Universidade de Santiago de Compostela.
  - **LX-SUITE**[29]**:** um conjunto de ferramentas implementado pelo grupo NLX[30] do Departamento de Informática da Universidade de Lisboa. Obteve uma precisão de 96,87% com o corpus LX-Corpus [49], gerado utilizando a ferramenta *TnT* (um anotador gramatical estatístico que pode ser treinado com diferentes idiomas) [50].
- **Vários idiomas:**
  - **FreeLing**[31]**:** uma biblioteca que suporta vários idiomas, entre eles o Português utilizando para isso o corpus Bosque. Esta biblioteca permite realizar análises morfológicas, análise de dependências, anotação gramatical entre outras [51]. Os analisadores usados são baseados em gramáticas livres de contexto [8]. O conjunto de etiquetas usadas é resultado de um projeto denominado por EAGLES[32].

[24] Disponível em http://wordfreak.sourceforge.net/index.html.
[25] Poderá ser consultado em http://opennlp.apache.org/index.html.
[26] Disponível em http://aelius.sourceforge.net/.
[27] Disponível em www.nilc.icmc.usp.br/nilc/tools/curupira.html.
[28] Disponível em gramatica.usc.es/ gamallo/tagger.htm.
[29] Disponível em http://lxcenter.di.fc.ul.pt/services/en/LXServicesSuite.html.
[30] Sigla para *Natural Language and Speech Group*.
[31] Disponível em http://nlp.lsi.upc.edu/freeling/.
[32] Disponível em http://www.ilc.cnr.it/EAGLES96/annotate/annotate.html.

- **SVMTool**[33]**:** é um gerador de analisadores sequenciais utilizando SVM. Os autores em [52] apoiaram o projeto nas seguintes características: (i) fácil configuração; (ii) facilidade de análise e extração de características dos dados de entrada; (iii) independente do idioma, desde que seja fornecido como entrada um corpus classificado no respetivo formato; (iv) testes com precisão acima dos 97%; (v) estratégias para tratar o ruído. Na página do SVMTool os autores disponibilizam modelos para o Catalão, Espanhol e três para o Inglês, com a informação do corpus usado e do respectivo processamento efetuado para criar cada um.

4.2 Problemas que ainda persistem

Como foi dito anteriormente, sistemas com um desempenho a rondar os 97% de precisão na anotação gramatical são considerados como estado da arte mas Christopher Manning em [53] prefere que a avaliação seja feita por outra prespetiva. Manning considera que o valor de 97% de precisão está sobrevalorizado visto o cálculo ser baseado numa anotação *per-token* onde sinais de pontuação e *tokens* não ambíguos influenciam o resultado. Seria mais realista calcular o valor da precisão de um sistema pelas anotações certas de frases inteiras, visto que apenas um erro na anotação de uma palavra influencia as restantes anotações da frase. Atualmente um desempenho estado da arte possui valores de precisão entre os 55% e os 57% para analisadores que acertam todas as anotações das frases. A precisão dos sistemas também é influenciada pela diferença de tópicos, época ou estilo de escrita entre o corpus utilizado para treino e os textos a serem anotados.

Manning, no seu trabalho, analisou as origens mais comuns dos erros que influenciam os anotadores gramaticais, e defende que as melhorias têm de ser feitas nos recursos linguísticos visto não ser possível melhorar significativamente as arquitecturas e algoritmos. O seu trabalho foi efetuado com o *Stanford Part-of-Speech Tagger* utilizando o *Penn Treebank* mas permitirá extrapolar conclusões para os recursos do Português.

Da sua análise, Manning dividiu os erros em sete tipos que podem ser vistos com as respetivas frequências na Tabela 3. Considerou 100 erros da secção 19 do *Penn Treebank*.

Tabela 3. Frequência e tipos de erros de anotação gramatical. (Fonte: [53])

Tipo de erro	Frequência
Lacuna lexical	4.5%
Palavra desconhecida	4.5%
Poderia plausivelmente acertar	16%
Linguística difícil	19.5%
Subespecificado/incerto	12.0%
Inconsistente/padrão errado	28.0%
Gold standard errado	15.5%

Os tipos de erros apresentados na Tabela 3 são:

– **Lacuna lexical:** a palavra ocorreu no corpus de treino, mas nunca com a etiqueta usada no contexto do teste;
– **Palavra desconhecida:** o anotador nunca viu a palavra e apenas pode usar as características de contexto que podem ser ambíguas.
– **Poderia plausivelmente acertar:** por vezes o anotador na prática falha o que na teoria devia de acertar com o contexto disponível.
– **Linguística difícil:** por vezes o anotador necessita de um conhecimento contextual maior do que o fornecido pelas características locais. Um exemplo é uma frase fornecer o contexto para a seguinte.

[33] Disponível em http://www.lsi.upc.edu/ nlp/SVMTool/.

– **Sub-especificado/incerto:** a etiqueta é ambígua ou incerta para o contexto em questão.
– **Inconsistente/padrão errado:** o *Gold standard* possui inconsistências, apresentando diferentes etiquetas para a mesma expressão usada em contextos similares.
– ***Gold standard* errado:** etiquetas do *Gold standard* estão erradas.

Manning em [53], sugere mudanças nos *corpora* para colmatar estes erros, mas tem existido por parte da comunidade ligada ao PLN uma reticência a efetuar tais mudanças. Porque ao tornar os *corpora* livres de erros, irá resultar na introdução *overfitting* nos modelos criados, gerando uma falta de adaptabilidade dos mesmos a novos casos. Visto os erros apresentados ocorrerem de forma sistemática, pelo menos no *Penn Treebank*, outra hipótese será acrescentar um conjunto de regras ao anotador gramatical, de modo a corrigir estes erros.

Se estes erros estão presentes em *corpora* para o Inglês será de supor que também possam ocorrer tipos de erros similares nos *corpora* do Português.

5 Conclusão

Durante este artigo tentou-se dar uma visão geral sobre vários aspetos da linguística, da anotação automática de textos e em particular do que é a anotação gramatical de textos.

A anotação gramatical de textos é uma das vertentes do PLN, que permite percorrer textos anotando gramaticalmente as palavras dos mesmos. Esta anotação pode ser usada como um produto final de um sistema ou como um passo intermédio para uma abordagem mais complexa. Tanto para o Inglês como para o Português já existem diversos sistemas e recursos linguísticos que permitem realizar a tarefa de anotação gramatical com um valor estado da arte.

O valor estado da arte para a anotação gramatical ronda os 97% de precisão quando visualizada numa vertente *per-token*. Quando se compara com a precisão da anotação certa de frases completas, entre 55% e 57%, existe uma discrepância superior a 40%. Esta discrepância deve-se ao facto de as frases não serem vistas como um todo aquando da anotação, mas apenas é considerado o contexto à volta de cada palavra, por norma numa janela de tamanho cinco. A inclusão dos sinais de pontuação também influência o valor da precisão *per-token*, visto serem *tokens* que os anotadores por norma acertam na sua anotação. Outros *tokens* que também influenciam esta discrepância são os não ambíguos, que são contados como certos no seu contexto mas que no conjunto da frase poderão não estar.

Verificou-se que em termos de arquitetura os sistemas de anotação gramatical neste momento estão no seu auge, logo para melhorar os seus resultados, as correções têm de vir dos erros efetuados devido aos recursos linguísticos. No *Penn Treebank* existe um conjunto de erros que influência os resultados dos anotadores, isto permite-nos supor que nos recursos linguísticos do Português poderão existir erros semelhantes. Mas ao seguir a abordagem de corrigir os erros presentes nos recursos linguísticos estaríamos a introduzir falta de adaptabilidade nos modelos gerados a partir desses recursos. Assim, a melhor maneira de adaptar os sistemas será introduzir meios que permitam corrigir erros dos anotadores que tendam a ocorrer de forma sistemática.

Como nota final, a anotação gramatical evoluiu muito nestes últimos anos, tal como toda a área do PLN, possuindo valores de precisão que rivalizam com anotadores humanos. Ao ser feita de forma automática, realiza a tarefa de uma forma mais rápida e não existe a possibilidade de erros devido ao cansaço. Apesar de a investigação continuar, estamos a chegar a um ponto em que a evolução, para colmatar os erros que ainda restam, se está a tornar mais difícil. Já não basta alterar as arquiteturas dos sistemas, para achar o que possui a precisão mais elevada, porque a maioria tem valores estado da arte. Temos de nos focar e analisar os erros dos recursos linguísticos, usados como treino, de modo a tentar mitigar o seu impacto na precisão final dos sistemas.

Referências

1. Miranda, N., Raminhos, R., Seabra, P., Sequeira, J., Gonçalves, T., Quaresma, P.: Reconhecimento de entidades nomeadas com SVM. In: Actas das Jornadas de Informática da Universidade de Évora 2010. (Novembro 2010)

2. Abreu, C., Paes, C.: Internet em números em 2012. http://expresso.sapo.pt/internet-em-numeros-em-2012=f775426 (Dezembro 2012)
3. Carreras, X., Màrquez, L.: Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In: Proceedings of the Ninth Conference on Computational Natural Language Learning, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 152–164
4. Sequeira, J., Gonçalves, T., Quaresma, P.: Semantic role labeling for portuguese – a preliminary approach. In: Computational Prodessing of the Portuguese Language: 10th International Conference, PROPOR 2012, Springer (2012) 193–203
5. Miranda, N., Raminhos, R., Seabra, P., Sequeira, J., Gonçalves, T., Quaresma, P.: Named entity recognition using machine learning techniques. In: EPIA'11 – 15th Portuguese Conference on Artificial Intelligence, Lisbon, PT (October 2011)
6. Rangel, F., Rosso, P., Koppel, M., Stamatatos, E., Inches, G.: Overview of the Author Profiling Task at PAN 2013
7. Buchholz, S., Marsi, E.: CoNLL-X shared task on Multilingual Dependency Parsing. In: CoNLLX'06 – 10th Conference on Computational Natural Language Learning. (2006) 149–164
8. Kubler, S., McDonald, R., Nivre, J.: Dependency Parsing. Synthesis Lectures on Human Language Technologies. Morgan & Claypool (2009)
9. T. K. Sang, E.F.: Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. In: Proceedings of CoNLL-2002, Taipei, Taiwan (2002) 155–158
10. Ide, N., Suderman, K.: Graf: A graph-based format for linguistic annotations. In: Proceedings of the Linguistic Annotation Workshop. LAW '07, Stroudsburg, PA, USA, Association for Computational Linguistics (2007) 1–8
11. Wilcock, G.: Introduction to Linguistic Annotation and text Analytics. Synthesis Lectures on Human Language Technologies. Morgan & Claypool (2009)
12. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Prentice Hall Series in Artificial Intelligence. Prentice Hall (2000)
13. Allen, J.: Natural Language Understanding (2nd Ed.). Benjamin-Cummings Publishing Co., Inc., Redwood City, CA, USA (1995)
14. Gonçalves, T.: Utilização de Informação Linguística na classificação de documentos em Língua Portuguesa. PhD thesis, Universidade de Évora (Novembro 2007)
15. Francis, W.N.: A Standard Sample of Present-day English for use with Digital Computers. Brown University (1964)
16. Francis, W.N., Kucera, H., Mackie, A.W.: Frequency Analysis of English Usage. Lexicon and Grammar. Houghton Mifflin (1982)
17. Greene, B., Rubin, G.: Automatic Grammatical Tagging of English. Department of Linguistics, Brown University (1971)
18. Atwell, E.: The brown corpus tag-set. http://www.scs.leeds.ac.uk/ccalas/tagsets/brown.html (Dezembro 2013)
19. Francis, W., Kucera, H.: Manual of information to accompany a standard corpus of present-day edited american english, for use with digital computers. http://icame.uib.no/brown/bcm.html (1997)
20. Marcus, M., Santorini, B., Marcinkiewicz, M.: Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics **19**(2) (1993) 313–330
21. Marcus, M., Taylor, A., MacIntyre, R., Bies, A., Cooper, C., Ferguson, M., Littman, A.: The Penn Treebank Project. http://www.cis.upenn.edu/ treebank/ (1999)
22. Taylor, A., Marcus, M., Santorini, B.: The penn treebank: An overview (2003)
23. Afonso, S., Bick, E., Haber, R., Santos, D.: Florestra Sintá(c)tica: A Treebank for Portuguese. In Rodrigues, M.G., Araujo, C.P.S., eds.: LREC'02 – 3rd International Conference on Language Resources and Evaluation, Las Palmas, Spain, ELRA (May 2002) 1698–1703
24. Bick, E.: The Parsing System "PALAVRAS": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Aarhus University Press (2000)
25. Linguateca: Florestra Sintá(c)tica. http://www.linguateca.pt/floresta/corpus.html (2009)
26. PÚBLICO Comunicação Social S.A.: Público. http://www.publico.pt (1990)
27. Empresa Folha da Manhã S.A.: Folha.com. http://www.folha.uol.com.br (1921)
28. Freitas, C., Afonso, S.: Bíblia florestal: Um manual lingüístico da Floresta Sintá(c)tica. http://www.linguateca.pt/Floresta/BibliaFlorestal (Setembro 2008)
29. Dietterich, T.: Machine learning. Nature Encyclopedia of Cognitive Science (2003)
30. Sequeira, J.: Extracção de relações entre entidades mencionadas. Master's thesis, Universidade de Évora (2011)

31. Ghahramani, Z.: Unsupervised Learning. Volume 3176/2004. Springer (2004)
32. Baum, L.E., Petrie, T.: Statistical inference for probabilistic functions of finite state markov chains. In: The Annals of Mathematical Statistics. Volume Volume 37. (1966) 1554–1563
33. Rabiner, L., Juang, B.: An Introduction to Hidden Markov Models. IEEE ASSP Magazine (Janeiro 1986)
34. Espindola, L.S.: Um Estudo sobre Modelos Ocultos de Markov, HMM - Hidden Markov Model. PhD thesis, Pontifícia Universidade Católica do Rio Grande do Sul, Faculdade de Informática, Porto Alegre (Junho 2009)
35. McCallum, A., Freitag, D., Pereira, F.: Maximum entropy markov models for information extraction and segmentation. In: Proceeding 17th International Conference on Machine Learning, Morgan Kaufmann (2000) 591–598
36. Pedras, J., Quaresma, P.: Extracção de informação de documentos em língua portuguesa: aplicação a domínios de anúncios. In: Actas das Jornadas de Informática da Universidade de Évora 2010. (2010)
37. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML'01 – 18th International Conference on Machine Learning. (2001) 282–289
38. Wallach, H.: Conditional random fields: An introduction (2004)
39. Kudoh, T., Matsumoto, Y.: Use of support vector learning for chunk identification. In: CoNLL'00 – 4th Conference on Computational Natural Language Learning. (2000) 142–144
40. Lorena, A., Carvalho, A.: Uma Introdução às Support Vector Machines. Revista de Informática Teórica e Aplicada - RITA **XIV**(2) (2007) 43–67
41. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Linguisticae Investigationes **30**(1) (2007) 3–26
42. Carreras, X., Màrquez, L.: Introduction to the CoNLL-2004 Shared Task: Semantic Role Labeling. In: CoNLL'04 – 8th Conference on Computational Natural Language Learning. (2004)
43. Voutilainen, A.: A syntax-based part-of-speech analyser. In: EACL'95 – 7th Conference of the European Chapter of the Association for Computational Linguistics. (1995) 157–164
44. Morton, T., LaCivita, J.: Wordfreak: An open tool for linguistic annotation. In: Proceedings of HLT-NAACL 2003, Edmonton (2003) 17–18
45. Martins, R.T., Hasegawa, R., Nunes, M.: Curupira: um parser functional para o português (Dezembro 2002)
46. Martins, R.T., Hasegawa, R., Nunes, M.: Curupira: a functional parser for brazilian portuguese. In Nuno J. Mamede, Jorge Baptista, I.T.M.d.G.V.N.E., ed.: Computational Processing of the Portuguese Language, 6th International Workshop, PROPOR 2003. Lecture Notes in Computer Science 2721, Springer (Junho 2003)
47. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: ICNMLP'94 – International Conference on New Methods in Language Processing. (1994)
48. Schmid, H.: Improvements in part-of-speech tagging with an application to german. In: EACL'95 – SIGDAT Workshop: From Text to Tags. (1995)
49. Branco, A., Silva, J.: Evaluating solutions for the rapid development of state-of-the-art pos taggers for portuguese. In: LREC'04 – 4th International Conference on Language Resources and Evaluation. (2004) 507–510
50. Brants, T.: TnT - A Statistical Part-of-Speech Tagger. In: ANLP'00 – 6th Applied Natural Language Processing. (2000)
51. Padró, L., Collado, M., Reese, S., Lloberes, M., Castellón, I.: Freeling 2.1: Five years of open-source language processing tools. In: LREC'10 – 7th Language Resources and Evaluation Conference. (2010)
52. Giménez, J., Márquez, L.: SVMTool: A general pos tagger generator based on support vector machines. In: LREC'04 – 4th International Conference on Language Resources and Evaluation. (2004)
53. Manning, C.D.: Part-of-speech tagging from 97linguistics? In: Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing - Volume Part I. CICLing'11, Berlin, Heidelberg, Springer-Verlag (2011) 171–189

Reconhecimento de entidades mencionadas em textos clínicos: Classificação e identificação de registos na área da pneumologia

Matheus Coppetti Silveira **, Vitor Beires Nogueira, and Irene Pimenta Rodrigues

Universidade de Évora
matheuscoppetti@gmail.com
{vbn,ipr}@uevora.pt
http://www.uevora.pt

Resumo O reconhecimento de entidades mencionadas (NER) em textos clínicos é uma tarefa que desempenha um papel de grande importância para a extração de informação e conhecimentos de pacientes e procedimentos médicos. Classificar as frases de acordo com os conceitos do SNOMED CT ajudará a determinar a especialização médica de cada um destes registos. Essa classificação é importante não apenas sob o ponto de vista organizacional, mas também, permite que textos organizados por especialidade sejam consultados e processados de forma mais eficiente de acordo com o domínio que pertencem. Este trabalho tem por objetivo a construção de um sistema que utilizará NER, através de uma abordagem baseada em regras, para classificar os textos médicos de acordo com os conceitos do SNOMED CT. Após esta classificação, utilizará técnicas para a identificação destes textos e determinar quais se inserem no domínio da pneumologia.

Keywords: NER, entidades mencionadas, processamento de língua natural, pneumologia, SNOMED

1 Introdução

O reconhecimento de entidades mencionadas (NER, do inglês *named entity recognition*) é um importante ramo no processamento de língua natural (PLN), tem como tarefa a classificação e pesquisa de expressões em textos de língua natural. Estas expressões vão desde nomes de pessoas e organizações a datas, possuindo dessa forma informações de valor sobre o texto.

O NER pode ser utilizado em diversas actividades de PLN, como por exemplo, uma ferramenta para pesquisa e filtro em um texto. Também, pode ser utilizado para realizar o pré-processamento para outras tarefas de PLN. As tarefas que utilizam NER são a tradução automática, *question answering*, *summarization*, modelação de língua natural e análise de sentimento em textos, e é retirando as vantagem do uso de entidades mencionadas (NE) e trabalhando com elas individualmente, que essas tarefas obtém melhores desempenhos.

O NER foi primeiramente introduzido no MUC-6 em 1995, e desde então os sistemas para NER estão migrado de aplicações baseadas em regras para modelos estatísticos. O estado da arte para a língua inglesa encontra-se nos 90% de precisão na identificação de entidades mencionadas, sendo assim, necessário preencher esta lacuna de desempenho em outros idiomas.

Embora tenha o seu estado da arte próximo aos 90%, em aplicações de NER para textos de domínio clínicos nem sempre se verifica estes valores. Devido as particularidades das informações nesses textos, são necessários *corpora* específicos para esta área, bem como o uso de regras que satisfaçam as necessidades para este ramo.

Os textos clínicos de pacientes armazenam informações de suma importância para a avaliação e processamento de dados desses. Esses textos são encontrados sob a forma de texto livre sendo assim de difícil processamento por ferramentas de *data mining* e de suporte à decisão. Dados de importância semântica como doenças, sintomas e medicamentos devem ser analisados para assim, extrair a maior quantidade possível do histórico clínico de um paciente.

** O presente trabalho foi realizado com apoio do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil

O SNOMED CT é um dicionário de terminologias clínicas e pode ser utilizado como uma ferramenta de suporte para as tarefas de classificação de NEs neste domínio. O SNOMED possui componente para a classificação de relações, conceitos e descrições[12].

A proposta deste trabalho é utilizar o SNOMED CT para classificação de textos clínicos e assim, determinar quais os pacientes apresentam problemas relacionados à área da pneumologia.

A seção 2 fará uma abordagem sobre as medidas de desempenho e métricas utilizadas para a comparação de resultados nas tarefas de processamento de língua natural.

Na seção 3 são apresentados os factores determinantes para o desempenho de sistemas que possuem tarefas de NER, entre eles, o idioma, o domínio e as entidades analisadas.

A seção 4 mostra as diferentes abordagens para as tarefas de reconhecimento de entidades mencionadas.

Uma visão geral sobre o estado da arte no uso de NER para a classificação de textos clínicos pode ser vista na seção 5.

A proposta deste trabalho é apresentada em maiores detalhes na seção 6, mostrando a metodologia utilizada, os métodos de preparação dos textos, a terminologia utilizada para a classificação das entidades mencionadas e o processo de classificação e qualificação dos termos etiquetados.

2 Medidas de Desempenho

É necessário em qualquer área de pesquisa, obter dados e comparar os resultados dos novos métodos. Para medir o desempenho de sistemas de NER é utilizado apenas um método e três métricas para essa avaliação: precisão, *recall* e medida-F. Utilizando essas medidas os objetos são classificados em duas classes: positivos (P) e negativos (N) e assim expandindo esta classificação em outras duas classes, os falsos positivos (FP) e falsos negativos (FN).

O cálculo do desempenho, de acordo com a medida desejada, pode ser feito da seguinte forma:

$$Precisao = \frac{P}{P+FP} \tag{1}$$

$$Recall = \frac{P}{P+FN} \tag{2}$$

$$Medida - F = \frac{2P}{2P+FP+FN} \tag{3}$$

A precisão pode ser definida como a garantia de que os objetos positivos foram marcados correctamente como positivos. O *recall* é a confiança de que todos os objetos positivos foram assinalados. A medida-F é uma média harmónica entre a precisão e o *recall* sob uma perspectiva mais geral.

As avaliações de desempenho são vistas sob diferentes formas por certas conferências em NER, como o MUC-6[11], o CoNLL[2] e o ACE[3].

3 Fatores de desempenho

Existem diversos fatores que alteram o desempenho das tarefas associadas ao NER, sendo os mais comuns o idioma, o domínio e a informação analisada.

3.1 Idioma

Pelo facto de que os sistemas baseados em regras serem desenvolvidos para um idioma específico, este é um dos principais factores para que determinam o desempenho das aplicações de NER pois a adaptação destes sistemas para outras línguas é de extrema dificuldade. A diferença de desempenho entre idiomas pode chegar a 20%, mesmo com a uso de sistemas de aprendizagem automático que permitem a escolha de funcionalidades independentemente do idioma utilizado.

Embora muitos dos sistemas tenham sido desenvolvidos para o inglês, outros idiomas também tem seus sistemas de NER com desempenho próximos ao estado da arte, como pode ser visto no quadro abaixo.

Idioma	Medida-F
Búlgaro	89%
Chinês	90%
Inglês	90%
Húngaro	92%

3.2 Domínio

O domínio das *corpora* utilizadas pode ter grande influência no desempenho dos sistemas de NER, não apenas relativo ao tamanho do *corpus* mas também pelo facto de alguns desses se mostrarem mais simples de serem utilizados pelos sistemas de NER, como é o caso de artigos de jornais e textos de redes sociais.

Os sistemas normalmente são treinados em um domínio e apesar disto, é desejado que eles sejam utilizados para domínio diferentes. Grande parte dos problemas de desempenho relacionados ao domínio é devido à necessidade de adaptação de sistemas treinados em diferentes domínios.

3.3 Entidades

A entidade que se deseja encontrar é um factor importante no que diz respeito à *performance*. Algumas entidades são mais fáceis de serem localizadas do que outras como é o caso dos países que são de mais fácil localização do que as organizações. Deve ser lavado em consideração que isto depende da definição das classes. Um exemplo de como esta definição tem impacto na pesquisa das entidades pode ser mostrado nas entidades do tipo *datetime*, estas entidades podem contem tanto datas em sua forma absoluta (i.e: 15 de janeiro de 2013) ou datas em uma forma relativa (i.e: próximo sábado).

Existem ainda, diferentes níveis de detalhe. O MUC-6 trabalha com três categorias, sendo que cada uma possui duas ou três subcategorias. Para o CoNLL há apenas quatro categorias. Sekine (2002) propõe o uso de 200 categorias. Um número mais elevado de categorias, apesar de permitir uma distinção maior na classificação, torna a pesquisa pelas entidades e a sua classificação uma tarefa mais difícil.

4 Abordagens

Independentemente do tipo de abordagem para a solução dos problemas em NER, existem dois passos para tal tarefa: criação e uso. É possível falar sobre o primeiro tipo que esse é feito durante a fase de criação da ferramenta, podendo ser feito de forma manual, ou seja, quando é feito pela construção de regras e intervenção humana. Quando a sua concepção se dá através de processos de análise dos parâmetros por um computador, se diz que esse é feito pelo aprendizagem automático (*machine learning* ou ML).

O segundo passo, o uso, pode ser dividido em modelos determinísticos e estocásticos. Esses modelos se diferenciam pelo facto dos modelos estocásticos se apoiam em distribuições de probabilidades. Essa diferença fica mais evidente quando são classificadas as NEs, enquanto nos modelos determinísticos classificam cada palavra com uma categoria, os modelos estocásticos atribuem à cada palavra um conjunto de possíveis classificações e suas respectivas probabilidades.

Sob uma perspectiva geral, quando se fala em sistemas baseados em regras, geralmente se tem sistemas determinísticos construídos de forma manual. Já, quando os sistemas são baseados em algum método de classificação, é mais comum de se observar sistemas estocásticos de aprendizagem automático.

5 Pesquisas Relacionadas

Diversas pesquisas foram realizadas na extração de informações de textos clínicos e biomédicos. Estas pesquisas buscam mapear não apenas textos em sua forma livre, mas também, interpretar conceitos descritos em códigos em uma dada terminologia, como SNOMED CT, ICD-9 entre outras.

Como um exemplo de pesquisas nessa área, o MetaMap (Aronson, 2001) realizou a descoberta de conceitos UMLS em textos clínicos através da combinação de frases com termos no *metathesaurus* do UMLS.

O Metamap além de utilizar de *parsing* para filtrar frases julgadas relevantes, usou ainda ferramentas para geração de variações nas frases. Outros exemplos de trabalhos que buscam a combinação de frases de textos clínicos com conceitos do UMLS podem ser vistos nas pesquisas de Zou et al. (2003) e Friedman et al. (2004).

Long (2005) e Patrick et al. (2007) desenvolveram seus trabalhos pesquisando por termos do SNOMED CT em frases de textos clínicos. Utilizaram técnicas que procuravam por abreviações, erros gramaticais, e diferentes ordenações de palavras para então realizar a comparação dessas com termos no SNOMED CT.

O presente trabalho faz a pesquisa de NEs, através de termos do SNOMED CT, em textos clínicos através de uma abordagem baseada em regras. Savova et al. (2010) mostraram em sua ferramenta uma solução semelhante, que busca pela identificação de NEs relativas a condições patológicas mapeadas através conceitos do SNOMED. Esse trabalho utilizou uma abordagem léxica baseada em regras e se apoiou em ferramentas para correções ortográficas e permutação de palavras. O uso destas técnicas permitiu alcançar uma precisão de 80% e *recall* de 60% segundo indicam seus relatórios [10].

Wang (2009) realizou uma comparação entre sistemas baseados em regras com sistemas de aprendizagem automático. Os sistemas deveriam classificar textos clínicos de diversas áreas relacionadas aos serviços de cuidados intensivos, e etiquetar os termos em 10 possíveis classes. Os métodos baseados em regras alcançaram uma precisão de 75% e *recall* de 52% utilizando os termos do UMLS, SNOMED e MOBY[1]. O processo de aprendizagem automático apresentou resultados acima dos vistos na abordagem baseada em regras, com precisão de 76.86% e *recall* de 66.26%, em processos que não utilizaram funcionalidades de análise de contexto. Utilizando de técnicas de interpretação de contexto a precisão foi de 84% e *recall* de 78.9%.

6 Metodologia

Textos clínicos na língua inglesa foram classificados em três categorias, sendo essas, doença ou disfunção, sintomas e productos ou medicamentos. A classificação das NEs foi feita através de uma abordagem baseada em regras e utilizando uma terminologia para o domínio clínico. As NEs classificadas foram novamente processadas para determinar se o texto tem ou não relações com a área da pneumologia.

A figura 1 mostra *workflow* do sistema. É possível analisar que um conjunto de textos médicos (A) tem suas frases classificadas por uma ferramenta de NER (B) que obtém as entidades mencionadas de acordo com os conceitos do SNOMED CT (C). Com a devida classificação dessas NEs, é gerado como saída um novo conjunto de documentos já etiquetados (D), esses por sua vez serão classificados pelo sistema, que com uso de consultas em SPARQL obtém informações semânticas de uma base de dados DBPedia (F) com o intuito de verificar a ligação dos textos com a área da pneumologia. Por fim, o sistema gera um novo conjunto de documentos, desta vez identificando aqueles que foram considerados como pertencentes ao domínio de interesse

6.1 Preparação dos textos

Os textos a serem analisados pelos sistema passam por uma série de etapas antes de serem classificados de acordo com os termos médicos. As etapas pelas quais os textos passam são:

- Conversão dos caracteres para letras minúsculas (*lower case*);
- Remoção de pontuação;
- *Tokenizing*;
- *POS-Tagging*;
- NER;
- Combinação e re-ordenação de palavras

As etapas de *POS-Tagging* e NER são feitas com o uso da plataforma NLTK para *Python*[2]. Para a execução do último passo descrito anteriormente, o sistema cria três novas listas de palavras, uma lista com

[1] dicionário da língua inglesa
[2] http://www.nltk.org/

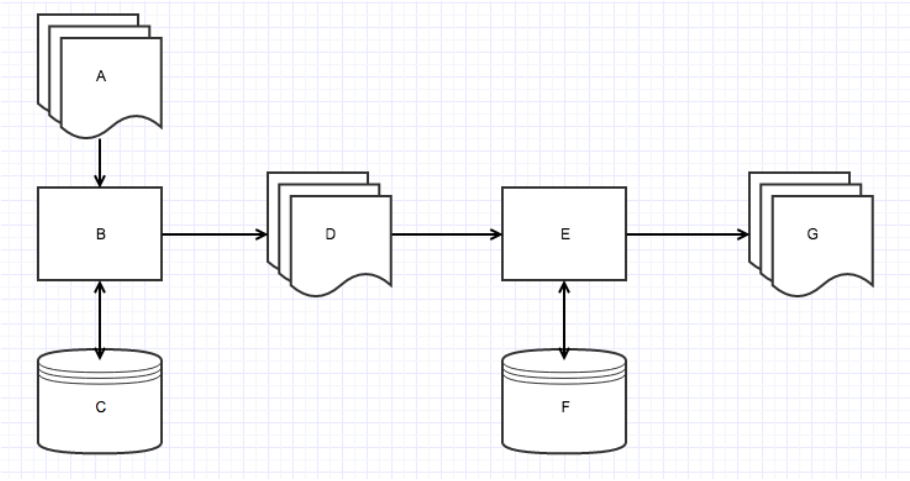


Figura 1. Workflow do funcionamento do sistema

todas as palavras etiquetadas na etapa de NER, uma segunda lista contendo apenas substantivos e adjetivos (na ordem em quem aparecem no texto), e por fim, uma última lista contendo apenas os substantivos da frase original. Após isto, cada palavra é combinada com até 4 das palavras a seguir a ela, e esta combinação é reordenada de forma que qualquer que seja a ordem destas palavras, ela seja prevista (i.e: *shows acute chest pain, shows chest pain acute, ...*). Este agrupamento e reordenação é feito para as três listas criadas com os termos classificados. Posteriormente é feita uma união destas combinações, remoção de termos duplicados e por fim uma ordenação alfabética dos termos.

A escolha por utilizar três listas geradas a partir dos termos da frase original, dá-se pelo facto de que é possível encontrar os sintomas apresentados de maneiras diferente como por exemplo *chest pain* pode ser descrito também como *pain in her/his chest*, assim, tratando apenas de substantivos ou substantivos e adjetivos, é possível buscar apenas pelos termos de maior relevância nas frases.

6.2 Terminologia utilizada

Foi utilizada a terminologia do Snomed CT. Essa terminologia clínica padronizada consiste de termos médicos organizados de maneira hierárquica. Esses termos além de apresentarem um nome específico e completo, apresentam ainda componentes semânticas que relacionam o termo com o conceito ao qual esse pertence. Os conceitos compreendidos pelo SNOMED CT são doenças ou condições atípicas (*disorder*), sintomas reportados pelo paciente (*finding*), partes do corpo (*body structure*), relações (*relation*), pessoas (*person*), valores qualificativos (*qualifier value*) e medicamentos ou productos (*product*)[13].

6.3 Processo de classificação dos textos clínicos

Para a identificação das entidades mencionadas correspondentes aos termos do SNOMED CT, são feitas pesquisas à uma base de dados contendo os conceitos e relações pertencentes ao domínio do SNOMED CT.

O sistema processa os textos e classifica os termos em três categorias diferentes:

- **Doenças ou condições anormais:** buscou a determinar se um termo presente nos textos representava uma doença ou alguma condição atípica. Essas condições representam patologias e não são momentâneas.
- **Sintomas:** os sintomas apresentados pelo paciente também foram classificados. Um sintoma não diz respeito à uma condição patológica e pode ser momentâneo.
- **Medicamentos:** foram etiquetados todos os productos ou medicamentos administrados em um paciente.

Esta classificação fica mais clara no exemplo:

```
Patient came to the causality complaining of chest pain for the past x days.
```

Após a classificação, esta frase retorna os seguintes termos:

```
[u'chest pain', u'finding', 29857009]
[u'pain', u'finding', 22253000]
[u'pain', u'finding', 275896009]
```

O sintoma *chest pain* é etiquetado como *finding*, que segundo possui o identificador 29857009 no SNOMED CT. Ao realizar a pesquisa pelo termo *chest pain*, é obtido como retorno um conjunto de termos que satisfazem a pesquisa, sejam eles, sintomas, procedimentos ou medicamentos, quaisquer termo que tenha uma relação com a palavra pesquisada será retornado, assim, o sistema deve ser capaz de analisar os dados retornados e aceitar apenas aqueles que são iguais ao termo pesquisado. É visto ainda que o termo *pain* é retornado duas vezes, sendo que cada uma delas, correspondendo a um conceito diferente no SNOMED CT.

6.4 Qualificação das etiquetas

O sistema tem como objetivo, ser capaz de avaliar quais dos textos processados tem ligação com doenças ou problemas respiratórios e pulmonares. Para alcançar tal objectivo é necessário então avaliar as NEs classificadas no passo anterior.

Tal avaliação é feita através de consultas em SPARQL à DBPedia. Essa consulta retorna um conjunto de instâncias com valores semânticos, sendo possível desta forma determinar a relação das NEs com problemas pulmonares e respiratórios.

Aos termos etiquetados ainda são adicionados novos termos que vão buscar a garantia de que a pesquisa retorne os resultados desejados, resultando desta forma as três variantes:

- chest pain + pulmonary
- chest pain + lung
- chest pain + respiratory

No caso de NEs compostas por mais de uma palavra, o sistema cria novos termos que serão consultados no caso de não obter respostas na consulta utilizando o termo em sua forma completa. Considerando o *chest pain*, se não houvesse nenhum retorno ou os resultados obtidos não fossem capaz de fornecer alguma resposta, o sistema ainda realizaria pesquisas com os termos *chest* e *pain*, totalizando assim, seis novos termos a serem consultados.

A pesquisa pelos termos *chest pain* + *respiratory* retornou 14 resultados, *chest pain* + *pulmonary* encontrou 12 e *chest pain* + *lung* 13. O sistema então considera que o termo *chest pain* obteve um acerto de 3/3, como se verifica na equação 4, que tem γ como número de retornos válidos. As possíveis taxas de acerto vão de 0-3/3, sendo que 3/3 é a óptima e 0/3 indica que nenhum termo foi encontrado.

$$acerto = \frac{\gamma}{3} \tag{4}$$

O sistema considera como positivo (tem relação com problemas pulmonares ou respiratórios) todos aqueles que obtiverem um acerto superior a 0.33. Assim, é necessário que seja satisfeita pelo menos uma das pesquisas (1/3 = 0.333). No caso do acerto ser menor que 0.33 o sistema faz as pesquisas com os termos derivados do original (i.e: *chest* + *respiratory*). O cálculo do acerto para este processe é feito de maneira diferente como mostra a equação 5.

$$acerto' = \frac{\sum_{i=1}^{n} acerto_i}{3+n} \tag{5}$$

Pode ser visto na equação 5 que é necessário somar os acertos individuais de cada termo ($acerto_i$) e ainda dividir este valor por $3+n$, onde n é o número de novos termos gerados. Assim, no caso dos termos *chest* e *pain*, apenas uma das consultas poderia retornar sem que nenhum resultado fosse encontrado, caso contrário o valor mínimo de 0.33 não seria atingido.

6.5 Avaliação dos resultados

Para obter uma maior fidelidade das avaliações, essas são comparadas com a opinião de um especialista para assim garantir que as avaliações foram correctas. A medida de desempenho adoptada no sistema é o *recall* (expressão 2) pois o que se deseja nesse é garantir que todos os registos com relação a problemas pulmonares ou respiratórios tenham sido classificados correctamente.

7 Conclusão e trabalhos futuros

Apesar do grande esforço aplicado em tarefas de NER, este ainda é um problema desafiador para os investigadores pois, mesmo com o grande avanço na área, ainda existem problemas a serem solucionados. Como exemplo desses problemas, há a questão da língua utilizada como entrada nesses, sendo este um factor determinante em questões de desempenho nos sistemas. Além do idioma o domínio utilizado para o treino também tem impactos em *performance*, tendo que serem pesquisadas alternativas para se conseguir aplicar sistemas treinados em diferentes domínios. Alcançar e superar estes desafios será útil em questões de reaproveitamento de sistemas em diversas aplicações.

Este trabalho tem como contribuição o desenvolvimento de uma ferramenta para NER baseado em regras que vai auxiliar na identificação de textos clínicos para a área da pneumologia. Esta tarefa é importante quando se tem uma grande colecção de dados de pacientes em forma de texto livre, e é necessário classificá-los de acordo com a especialização médica a qual pertencem.

Serão ainda realizadas novas implementações neste estudo em trabalhos futuros, como o uso de ferramentas para correções ortográficas, abreviação de palavras e reordenação de termos. Utilizar uma nova fonte de consulta, além da DBPedia, é um objetivo futuro, que vai garantir uma melhoria na classificação dos textos, uma vez que nem todos os termos do SNOMED CT possuem uma correspondência nesta fonte utilizada.

Além do uso de novas fontes externas, é pretendido também utilizar outros dicionários para classificação das NEs nas frases clínicas, além disso, se busca no futuro utilizar ontologias médicas para inferir perguntas nos textos clínicos classificados como pertencentes à área da pneumologia.

Referências

1. WANG, Yefeng. Annotating and Recognising Named Entities in Clinical Notes. ACL-IJCNLP 2009, p. 18, 2009.
2. TJONG KIM SANG, Erik F.; DE MEULDER, Fien. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4. Association for Computational Linguistics, 2003. p. 142-147.
3. DODDINGTON, George R. et al. The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation. In: LREC. 2004.
4. SEKINE, Satoshi; SUDO, Kiyoshi; NOBATA, Chikashi. Extended Named Entity Hierarchy. In: LREC. 2002.
5. ARONSON, Alan R. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In: Proceedings of the AMIA Symposium. American Medical Informatics Association, 2001. p. 17.
6. ZOU, Qinghua et al. IndexFinder: a method of extracting key concepts from clinical texts for indexing. In: AMIA Annual Symposium Proceedings. American Medical Informatics Association, 2003. p. 763.
7. FRIEDMAN, Carol et al. Automated encoding of clinical documents based on natural language processing. Journal of the American Medical Informatics Association, v. 11, n. 5, p. 392-402, 2004.
8. LONG, William. Extracting diagnoses from discharge summaries. In: AMIA annual symposium proceedings. American Medical Informatics Association, 2005. p. 470.
9. PATRICK, Jon; WANG, Yefeng; BUDD, Peter. An automated system for conversion of clinical notes into SNOMED clinical terminology. In: Proceedings of the fifth Australasian symposium on ACSW frontiers-Volume 68. Australian Computer Society, Inc., 2007. p. 219-226.
10. SAVOVA, Guergana K. et al. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. Journal of the American Medical Informatics Association, v. 17, n. 5, p. 507-513, 2010.
11. GRISHMAN, Ralph; SUNDHEIM, Beth. Design of the MUC-6 evaluation. In: Proceedings of a workshop on held at Vienna, Virginia: May 6-8, 1996. Association for Computational Linguistics, 1996. p. 413-422.
12. STEARNS, Michael Q. et al. SNOMED clinical terms: overview of the development process and project status. In: Proceedings of the AMIA Symposium. American Medical Informatics Association, 2001. p. 662.
13. IHTSDO, SNOMED CT. Style Guide: Clinical Findings. 2008.

A Discourse Controller to Improve Question Answering Systems for Semantic Web

Dora Melo[1,3], Irene Pimenta Rodrigues[2,3], and Vitor Beires Nogueira[2,3]

[1] Iscac - Coimbra Business School, Polytechnic Institute of Coimbra,
Quinta Agrícola-Bencanta, 3040-316 Coimbra, Portugal
[2] Department of Informatics, University of Évora,
Rua Romão Ramalho, No. 59, 7000-671 Évora, Portugal
[3] Centre for Artificial Intelligence (CENTRIA), Department of Informatics, FCT/UNL,
Quinta da Torre, 2829-516 Caparica, Portugal
dmelo@iscac.pt,{ipr,vbn}@di.uevora.pt

Abstract. The Question Answering systems for Natural Language on the Semantic Web require mechanisms to interpret the documents returned, beyond the matching words in documents (currently done by search engines through the internet), as well as to find a precise answer without requiring the help of user. In this paper, we introduce a Discourse Controller that, through the analysis of the semantic of the questions, the structure of the discourse that includes the intentions of the user and the context of the questions, the type of expected answer, deals with multiple answers and providing accurate and justified answers to the questions posed by the user in Natural Language. When the Discourse Controller can not decide the correct path to obtain the answer, it starts a controlled clarifying dialogue with the user. To evaluate the performance of the proposed framework, we present a small experimental test suite, the results obtained allow to verify the effectiveness of the system.

Keywords: Question Answering Systems, Ontologies, Natural Language Processing, Semantic Web

1 Introduction

The Question Answering systems for Natural Language on the Semantic Web requires mechanisms to interpret the documents returned, in addition to the correspondence between words in documents (currently done by search engines through the internet), as well as to find a precise answer, without requiring the help of user, e.g., requires the use of knowledge and reasoning to interpret and to obtain the questions' answers [17]. Consistent with the role of ontologies in structuring and organizing semantic information on the web, the Question Answering systems based on ontologies allows to explore the expressive power of ontologies and enrich the queries' interpretation. Ontologies and the Semantic Web [6] became essential formalisms to represent the conceptual domains of knowledge and promote the capabilities of Question Answering systems based on semantics.

In this paper, we introduce a Discourse Controller that, through the analysis of the semantic of the questions, the structure of the discourse that includes the intentions of the user and the context of the questions, the type of expected answer, deals with multiple answers and providing accurate and justified answers to the questions posed by the user in Natural Language (currently, we use only the English language). When the Discourse Controller cannot decide the correct path to obtain the answer, it starts a controlled clarifying dialogue with the user. The Discourse Controller makes use of ontologies, OWL2 descriptions and other web resources such as DBpedia [1] and WordNet [4]. When the Discourse Controller cannot decide the correct path to obtain the answer, it starts a controlled clarifying dialogue with the user. The proposed Discourse Controller is part of a Cooperative Question Answering system for Ontologies OWL2 presented in [12, 11].

The remaining sections of the paper are organized as follows. First, in Section 2, we present some related work, highlighting the similarities and differences with our proposal. In Section 3, we

introduce the proposed Discourse Controller, highlighting its capabilities. Hereafter, in Section 4, we present a preliminary evaluation which boils down to a first experimental set of tests done to the system. Finally, in Section 5, we present the conclusions and the future work.

2 Related Work

The Cooperative Question Answering systems are systems that automatically collaborate with the users, in order to obtain the information and clarification needed to provide the correct answer. These systems provide the user with additional information, intermediate answers, qualified answers and/or alternative questions. In this research field, we can find several proposals to these systems, among them: WEBCOOP, a Logic-Based Question Answering system, that integrates knowledge representation and advanced strategies of reasoning to generate cooperative answers to web queries; START [8] is a Natural Language Question Answering system that provides users with multimedia information access through the use of Natural Language annotations; PANTO [19] is a portable Natural Language interface to ontologies, which accepts input as Natural Language form and the output is in SPARQL query format. It is based on a triple model, in which parse tree is constructed for the data model using the Stanford parser; PowerAqua [10] is a multi-ontology based Question Answering system that takes as input queries expressed in Natural Language and is able to return answers drawn from relevant distributed resources on the Semantic Web; Querix [9] is an ontology-based Question Answering system which relies on clarification dialogues in case of ambiguities.

Like the systems presented, we also use: semantic interpretation techniques and syntactic parser to interpret and represent in some way the questions posed by the users; reasoning and inference techniques to extract and filter the information needed from the knowledge bases. Our proposal is a friendly, simple and cooperative Question Answering system. At this stage, we not claim that our proposal to be a complete intelligent system by interpreting and understanding the input questions. It exploit a reduce set of Natural Language Processing techniques (like Stanford parser, Discourse Representation Structures, WordNet), inference rules and opens a controlled dialogue with the user when can not continue the process of achieving the answer. The main difference is the cooperative way that it reaches the answers to the Natural Language questions posed by the user. The system interacts with the user in order to disambiguate and/or to guide the path to obtain the correct answer to the query posted, whenever this is possible to do by the reasoner. It also uses cooperation to provide more informed answers.

3 Discourse Controller

The Discourse Controller is the main component of the Cooperative Question Answering system for Ontologies, presented in [12, 11]. This component is invoked after the Natural Language question has been transformed into its semantic representation. Essentially, the Discourse Controller deals with the set of discourse entities and is able to compute the question posed by the user: verifies the question presupposition and chooses the sources of knowledge (Ontologies, WordNet [20], etc.) to be used; decides when the answer has been achieved or iterates using new sources of knowledge. The decision of when to relax a question in order to justify the answer, when to clarify a question and how to clarify it, is also taken in this module. Thus, the Discourse Controller represents the intentions and beliefs of the system and the user, the structure of discourse and the context of the question.

The architecture of the Discourse Controller is presented in Figure 1 and to help understand how it works, a brief discussion of the main components follows.

Question DRS is the Discourse Representation Structure of the Natural Language question posed by the user and is supported by Discourse Representation Theory [7]. The transformation of the Natural Language question into its corresponding DRS is supported by two system modules: the Syntactic Analysis and the Semantic Interpretation. The Syntactic Analysis module receives the Natural Language question posed by the user and use grammatical interpretation to generate a derivation tree of the question, using the Stanford parser, which is transformed

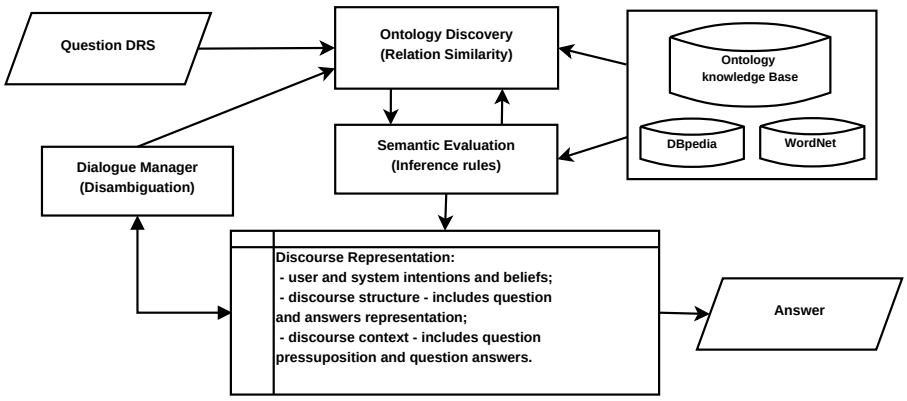


Fig. 1. Discourse Controller Architecture: Question DRS is the Discourse Representation Structure of the Natural Language question posed by the user; Ontology Discovery is the component that looks for knowledge base entities that represent the question's concepts; Semantic Evaluation is the component that reinterprets the semantic representation of the sentence based on the ontology considered; Dialogue Manager is the component that manages the controlled dialogue with the user; Discourse Representation is all the question's information obtained by the system.

into its syntactic structure. The Semantic Interpretation module is responsible for rewrites the syntactic structure of the question into its corresponding DRS. This process is based on first-order logic [5] extended with generalized quantifiers [2] and follows an approach similar to the one for constructing a Question Answering system over document databases proposed in [13].

Ontology Discovery is guided by the Discourse Controller to obtain the extension of sentence representation along with the reasoning process. The Ontology Discovery is invoked when the Discourse Controller has to look for knowledge base entities that represent the question's concepts. At this stage, the system has to transform the question's DRS predicates into their corresponding ontology representation.

Semantic Evaluation is intended to be the pragmatic evaluation step of the system, where the question semantic representation is transformed into a constraint satisfaction problem. This module reinterprets the semantic representation of the sentence, based on the ontology considered, in order to obtain the set of facts that represents the information provided by the question, e.g, the system has to find the entities of the knowledge base that are solutions to the ontology representation DRS. The solutions will be added to the solutions set and the ontology representation DRS will be added to the discourse representation of the question. The Semantic Evaluation module makes use of ontology, logic based semantic techniques, logic inferences and the SPARQL query.

3.1 Answer's Extraction

The answer's extraction consists in finding all solutions to the question posed by the user. That is, when the Natural Language question has been transformed into its semantic representation, the Discourse Controller tries to make sense of the input query by looking at the structure of the ontology and the information available on the Semantic Web, as well as using string similarity matching and generic lexical resources, in order to obtain the set of entities that are solutions to the question. The Discourse Controller must supervise the search (made at Ontology Discovery step) and validation (made by Semantic Interpretation module) of the entities among the knowledge base and when a solution is found, it will be added to the discourse representation of the initial question.

Consider the question *"Who is Alexander Mackenzie?"* and its semantic representation: `drs('PERSON', [who-X, exist-Y], [name(Y,'Alexander Mackenzie'), person(X)], [is(X,Y)]).` where: the discourse's referents are `where-X` and `exist-Y`, with `X` an entity of the discourse uni-

versally quantified and `Y` an existentially quantified discourse entity; the predicate main question is `is(X, Y)`; the presuppositions' predicates are `name(Y, 'Alexander Mackenzie')` and `person(X)`; and the type of the question is "PERSON".

For this query, we have to find entities in DBpedia, which are related to the name "Alexander Mackenzie", like `http://dbpedia.org/resource/Alexander_Mackenzie`, and have to get facts about those entities through non-taxonomic relation that verifies the question. For instance, the DBpedia contains the statement (triple RDF) `<http://dbpedia.org/resourceAlexander_Mackenzie> <http://dbpedia.org/ontology/type> <http://www.w3.org/1999/02/22-rdf-syntax-ns#Person>`, which validate the mapping of the questions' term in the ontology, e.g., `person` is mapped into `http://dbpedia.org/ontology/Person`, and determine a solution to the semantic representation of the question `X = Y = http://dbpedia.org/resource/Alexander_Mackenzie`. The solution, the RDF triples that generate the solution and the mappings of the question's terms in the ontology, which validate the semantic representation of the question, are added to the knowledge base (the discourse's representation) of the question.

3.2 Answer's Processing

The answer's processing consists in determining the answer returned to the user, which is interpreted with the information at the knowledge baseD. If the set of solutions is empty, the answer has to inform that fact and the user can re-write the initial question, or make a new one, or simply stop the process. If the set of solutions has only one solution, the answer presented to the user, besides direct and objective, also informs about the entities that served as support, allowing a better communication between the system and the user. Thus, the Discourse Controller presents the entities of the solution and respective descriptive summaries, provided by DBpedia.

If the set of solutions has multiple solutions, various interpretations can be made. If the Discourse Controller does not have enough information to decide which one is the correct, starts a controlled dialogue with the user. So, it presents a set of alternatives and the user's answer to those alternatives will clarify or restrict the subject is referring to. The set of alternatives consists of attributes which distinguish solutions. The Algorithm 1 shows how the system process to clarify the ambiguity of multiple solutions. The reformulation of this algorithm follows the idea presented by the authors of [16]. When the user's choice clarify the ambiguity, the system can provide the answer to the initial question.

Algorithm 1 Multiple solution's clarification.

Require: $S = \{s | s$ is a solution of the question$\}$
Ensure: Set an answer to the question
1: **while** $\#S > 1$ **do**
2: For each referent collect their properties
3: Evaluate the best property to differentiate the referents
4: Choose the best property
5: $A =$ values of the best property for each solution (where the property is defined)
6: Show the clarification's alternatives based on the set A
7: Receive the user's choice
8: Restrict the set of solutions S to the user's choice
9: **end while**
10: Show solution S to the user

To help understand how Algorithm 1 works, a brief discussion of the main steps follows.

Evaluation of the Properties The alternatives of clarification to present to the user must fulfil two important aspects: report as possible the user's intentions; and having information most likely to be known by the user. The second aspect is based on the parameter that the user can easily know more textual information than numerical. For instance, if the clarification concerns the characteristics of a person, most likely, it is easier for user know its country than its date of birth. Another possible parameter can be consider that the user would know better numerical information.

Regarding the first aspect, the choice of alternatives to be presented to the user consists in select the best property that contains more amount of information. Choosing the best property is based on the model of Decision Trees Learning [14], more precisely the ID3 (*Inductive Decision Tree*) algorithm used as a classification method in the construction of decision trees. The main reason for this choice is that decision trees can be applied to large data sets and make possible a real view of the nature of the decision process. In addition, decision trees are among the most practical and commonly method used in inductive inference. This method provides functions such as decision trees and these trees are trained according to a training set (samples classified in advance), and subsequently, other examples are classified according to the same tree. Since we are in an experimental phase, the implementation of ID3 algorithm (to the detriment of other more efficient and flexible algorithms, like C4.5 [15]) resulted in a simple task, allowing us to achieve the test results quickly.

The ID3 algorithm uses Entropy and Information Gain to build the decision tree. However, the classification of properties by maximizing the information gain gives preference to properties with many values. For that reason we introduced the Information Gain Ratio also as evaluation criteria, promoting properties which Entropy value is small and therefore promoting the properties that contain fewer values.

The Entropy of a set can be defined as the purity (certainty, accuracy) of that set. This concept borrowed from Information Theory defines the measure of "lack of information", namely the number of bits needed, on average, to represent the missing information, using optimal coding. If the set is completely uniform, the Entropy value is zero, and if the set is divided equitably, the Entropy value is equal to 1. Formally, the Entropy of a set is defined as follows:

Definition 1 (Entropy). *Given a set T, with instances of the class i, with probability $p_i \neq 0$. The Entropy of the set T is obtained by the following expression*

$$Entropy(T) = -\sum p_i \times log_2(p_i). \tag{1}$$

The Entropy of a set T verifies the property $0 < Entropy(T) < log_2(n)$, where n is the total of classes i.

The construction of a derivation tree is guided by the objective of reducing the Entropy, the difficulty of predicting the variable that defines the classes. The Information Gain defines the decrease in Entropy. Thus,

Definition 2 (Information Gain). *The Information Gain is the expected reduction in Entropy caused by partitioning the data according to the property testing P. The Information Gain value for the property P is obtained by the expression:*

$$Gain(T, P) = Entropy(T) - \sum_{v \in values(P)} (\frac{|T_v|}{|T|} \times Entropy(T_v)) \tag{2}$$

The concept Information Gain Ratio is the weight of Information Gain of the property relative to the Entropy of the property, e.g.,

Definition 3 (Information Gain Ratio). *Consider the property P of the set T, the Information Gain Ratio value of the property P in the set T is obtain using the following expression:*

$$GainRatio(T, P) = \frac{Gain(T, P)}{Entropy(T, P)} \tag{3}$$

The Information Gain Ratio is not defined when $Entropy(T, P) = 0$.

Choose the Best Property The Information Gain criterion of a property selects as property test one that maximizes the Information Gain. However, this criterion gives preference to attributes with many possible values (which corresponds to the number of edges of the decision tree). In these cases, it could be presented to the user an attribute totally irrelevant, where there is only one alternative for each possible value. Therefore, the number of alternatives would be equal to the

number of identifiers and the Entropy value would be minimal because, in each property, all samples (if only one) belong to the same class, that would generate a maximum gain, although totally useless. When this problem occurs, e.g., when the property P to the set T has $Entropy(T, P) = 0$, which corresponds to the Information Gain maximum value, we use the Information Gain Ratio as evaluation criteria to choose the best property.

However, even with these criteria we may have as best property, a property for which information is less known to the user. For example, if the best property represents numeric values (such as birth dates, number of citizen card, etc..), we assume that the user may not know such information. By the presentation of such alternatives to the user will result in unnecessary step. Thus, in these cases we define as priority the properties containing information (value) that is non-numeric.

Back to our example, the question posed by the user refers to the person "Alexander Mackenzie". When the Discourse Controller analyses the set of solutions detects the presence of multiple solutions, since there is 15 entities that represent the term "Alexander Mackenzie". The Discourse Controller does not have enough information to decide which solution is the correct and starts a controlled dialogue with the user to clarify the ambiguity, by performing the Algorithm 1.

In DBpedia databases, the properties are expressed by triples RDF, forming the properties that are associated to each question's referents. In the example, we exclude the referent `X` associated to the question adverb, because it is a referent that is semantically related to what the user wants to know. Thus, left us the referent `Y` which is associated with the name "Alexander Mackenzie". Consequently, the set T consists only of properties associated to the referent `Y`. The set T has 1168 properties, with $Entropy(T) = 3.6861$ and verifies $0 < Entropy(T) = 3.6861 < log_2(n) = log_2(15) = 3.9068905956$. To choose the best property, the system has to calculate the Information Gain and Information Gain Ratios values to each distinct property of the set T. The set T has 102 distinct properties and by applying the Definitions 2 and 3 for each property, we obtain 87 properties that have Entropy value equal to zero and that The Information Gain Ratio is not defined. The property with the highest value of the Information Gain Ratio is `http://dbpedia.org/property/placeOfBirth`.

Controlled Dialogue with the User Defined how the evaluation of the properties is made - through the use of Entropy, Information Gain and information Gain Ratio; defined how to choose the best property to submit their values to the user as alternative in the controlled dialogue - using the higher value of the Information Gain Ratio and the property information is non-numeric; according to Algorithm 1, remains to construct the set A, whose elements are the values of the best property, and then present them to the user, so that he can clarify the system on his intentions. In addition to specifying one of the possible alternatives presented by the system, the user also has three possibilities to interact with the system, namely: the symbol `?`, means that "I do not know"; the symbol `!`, meaning "show all answers"; and the term `quit`, which ends the process. In the first case, the system displays a new set of alternatives according to the current evaluation of properties. In the second case, the system displays all solutions and the process is finished. In the third case, the system simply ends the process.

Continuing our example and according to the evaluation made, the best property restricts the set of solutions into 9 solutions, because only these are characterized with the best property. Consider T_1 the set of events of the set T where the best property occurs. The set of alternatives A is formed by the different values of the best property in the set T_1. Then the system starts a controlled dialogue with the user by presenting the alternatives obtained. The choice made by the user will allow to clarify the system about what subject his referring to. The question posed to the user, by the system, is based on the description of the best property, e.g., the place of birth. Thus, the Discourse Controller present the following dialogue to the user:

```
USER: "Who is Alexander Mackenzie?"
SYSTEM: Do you know the place Of Birth?
1 - Scotland
2 - Logierait
3 - Perthshire
4 - Stornoway
5 - Lewis
```

```
6 - Outer Hebrides
7 - Warwick, Ontario
8 - Aberdeen
9 - Inverness
10 - Ontario
11 - Potosi, Wisconsin
12 - Liverpool
13 - Dumfries, Scotland
USER: 3
```

The alternative chosen by the user lead the system to one solution. So the ambiguity is clarified and the Discourse Controller is able to process the answer and return it to the user.

```
PERSON:
   Alexander Mackenzie, PC (January 28, 1822 { April 17, 1892), a building
   contractor and newspaper editor, was the second Prime Minister of Canada
   from November 7, 1873 to October 8, 1878.
RESOURCE:
   http://dbpedia.org/resource/Alexander_Mackenzie
```

Any of the alternatives presented, except 1, lead the system to one solution. The choice of alternative 1 lead the system to restrict more the set of solutions S and, since ambiguity remains, again the system has to execute the Algorithm 1 to clarify the ambiguity. More precisely, for the new set of solution repeat all the steps until the system can clarify the ambiguity and provide an answer to the user. Suppose that the user not know nothing about the place of birth. Then the system has to choose the next best property and present another set of alternatives to the user. The process will continue until the system could provide an answer to the user. The interaction is used to help the system in finding the right path to the answer. Thus, the cooperation between the user and the system, makes one able to get closer to the answer desired by the user. In many cases, the user is the only one who can help the system in the deduction and interpretation of information.

4 Experimental Results

The evaluation of the Discourse Controller requires the use of a knowledge base constituted only by the DBpedia ontology OWL2 [3], it covers about 359 classes forming a subsumption hierarchy, the classes are described by 1,775 different properties. The DBpedia Ontology currently contains about 2,350,000 instances. In order to complete the knowledge base, we use SPARQL *endpoints*, to query DBpedia database, and the DBpedia *Lookup Service*[4] to look up DBpedia URIs[5] by related keywords. The evaluation test was performed using a set of 84 direct "wh" questions, presented in TREC 9 (*The Ninth Text REtrieval Conference* [18]). The set under analysis contains only direct "wh" questions, which comprises the following questions:

```
222. Who is Anubis?
232. Who invented television?
390. Where was John Adams born?
459. When was John D. Rockefeller born?
534. Where is Windsor Castle?
759. What is the collective noun for geese?
```

In an initial analysis of the results, we started by observing that the system has not obtained any answer to 10 questions (12% of the questions). That is, the system did not found, in the knowledge base, the resources which identifies the questions terms or the entities that are solutions to the questions. For instance for the question 759, the system did not find resources able to relate the terms "collective noun" and "geese". Opening a dialogue with the user will allow us to rewrite the question posed, clarify the terms or place a new question. This way, and if the knowledge base

[4] http://wiki.dbpedia.org/lookup/

[5] http://tools.ietf.org/html/rfc3986/

has the answer, clearly we will be able to increase the results. Analysing the remaining corpus, reduced to 74 questions, we obtained 68 correct answers (81% of the questions), and we have verified it manually. Within these, 48 questions were multiple answers (57% of the questions) that, with the clarification of the user, the system returned the expected answer. We found that, for each multiple solution question, the system achieved an average of 3-4 solutions. Clearly a reduced set of alternatives, highlighting the potential of the system in searching the correct answer. The remaining 6 questions (7% of the questions), the system did not get the correct answer. These failures are identified with some factors that lead to incorrect interpretations, namely: semantic representation of the question; incomplete or badly formulated questions; dimension of the knowledge base, or incomplete information and non-uniform ontology resources.

This is still a preliminary evaluation, summarizing just a first set of tests, whose results have produced satisfactorily, allowing us to verify the effectiveness of the proposed question answering system and to identify the weaknesses that will allow us to improve its performance. In the future we intend to present a more complete evaluation, extend the set of questions to others questions' types and include the evaluation of the execution time. However, the results encouraged us to proceed.

5 Conclusions and Future Work

We presented a Discourse Controller that represents the semantics of the questions, the structure of the discourse that includes the intentions of the user and the context of the questions, which gives us the ability to deal with multiple answers and to provide justified answers. The experiments on a set of simple and direct "wh" questions have shown that our system produces promising results. We believe that adding a tool, like Discourse Controller, to the Question Answering system allows to improve substantially the system performance. Resorting to a controlled dialogue in order to clarify ambiguities, helps the system to better interpret the user's pretensions. These dialogues allows to increase the results obtained by the system and helps it to generate an answer more objective, clear and with the information desired by the user. Since one of our goals is to generate answers expressed in Natural Language and do not provide a list of results, we think that with the tests made shows the success of our proposal. Therefore, we believe that our proposal approaches rapidly to one that helps bridge the gap between the logic based Semantic Web and real world users. As future work, we plan to improve the search ontology techniques and the inference rules aiming to enhance the results of correct answers. In addition, we intend to implement and test other algorithms that will make the controlled dialogue with the user more efficient and flexible. We also plan to increase the number of tests, which cover the remaining types of questions (including more complex questions) and expected answers and to define a more complete quantitative, qualitative and comparative evaluation of the performance both of the system as of the Discourse Controller.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J.: Dbpedia: A nucleus for a web of open data. The Semantic Web 4825(Springer), 722–735 (2007)
2. Barwise, J., Cooper, R.: Generalized quantifiers and natural language. Linguistics and philosophy 4(2), 159–219 (1981)
3. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia-a crystallization point for the web of data. Web Semantics: Science, Services and Agents on the World Wide Web 7(3), 154–165 (2009)
4. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database (Language, Speech, and Communication). The MIT Press, illustrated edition edn. (May 1998)
5. Hodges, W.: Classical logic i: first-order logic. The Blackwell guide to philosophical logic pp. 9–32 (2001)
6. Horrocks, I.: Ontologies and the semantic web. Communications of the ACM 51(12), 58–67 (2008)
7. Kamp, H., Reyle, U.: From Discourse to Logic, Studies in Linguistics and Philosophy, vol. 42. Kluwer (1993)

8. Katz, B., Lin, J.J., Felshin, S.: The start multimedia information system: Current technology and future directions. In: Multimedia Information Systems. pp. 117–123. Arizona State University (2002)
9. Kaufmann, E., Bernstein, A., Zumstein, R.: Querix: A natural language interface to query ontologies based on clarification dialogs. In: 5th International Semantic Web Conference (ISWC 2006). pp. 980–981. Springer (November 2006)
10. Lopez, V., Fernández, M., Motta, E., Stieler, N.: Poweraqua: Supporting users in querying and exploring the semantic web. Semantic Web 3(3), 249–265 (2012)
11. Melo, D., Rodrigues, I.P., Nogueira, V.B.: Puzzle out the semantic web search. International Journal of Computational Linguistics and Applications 3(1), 91–106 (June 2012)
12. Melo, D., Rodrigues, I.P., Nogueira, V.B.: Work out the semantic web search: The cooperative way. Advances in Artificial Intelligence 2012, 3:3–3:3 (January 2012)
13. Quaresma, P., Rodrigues, I., Prolo, C., Vieira, R.: Um sistema de pergunta-resposta para uma base de documentos. Letras de Hoje 41(2), 43–63 (2006)
14. Quinlan, J.R.: Induction of decision trees. Mach. Learn. 1(1), 81–106 (Mar 1986)
15. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
16. Quintano, L., Rodrigues, I.: Using a logic programming framework to control database query dialogues in natural language. In: Proceedings of the 22nd international conference on Logic Programming. pp. 406–420. ICLP'06, Springer-Verlag, Berlin, Heidelberg (2006)
17. Saint-Dizier, P., Moens, M.F.: Knowledge and reasoning for question answering: Research perspectives. Information Processing & Management 47(6), 899 – 906 (2011)
18. Voorhees, E.M.: Overview of the trec-9 question answering track. In: Proceedings of the Ninth Text REtrieval Conference (TREC-9). pp. 71–80 (2001)
19. Wang, C., Xiong, M., Zhou, Q., Yu, Y.: Panto: A portable natural language interface to ontologies. In: Proceedings of the 4th European conference on The Semantic Web: Research and Applications. pp. 473–487. ESWC '07, Springer-Verlag, Berlin, Heidelberg (2007)
20. Witzig, S., Center, A.: Accessing wordnet from prolog. Artificial Intelligence Centre, University of Georgia pp. 1–18 (2003)

MR Brain Image Classification: A Comparative Study on Machine Learning Methods

Shib Sankar Bhowmick[1,2,⋆], Indrajit Saha[1,⋆], Luis Rato[2], and Debotosh Bhattacharjee[1]

[1] Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, West Bengal, India.
[2] Department of Informatics, University of Evora, Evora 7004-516, Portugal.
shibsankar.ece@gmail.com, indra@icm.edu.pl, lmr@di.uevora.pt, debotosh@ieee.org

Abstract. The brain tissue classification from magnetic resonance images provides valuable insight in neurological research study. A significant number of computational methods have been developed for pixel classification of magnetic resonance brain images. Here, we have shown a comparative study of various machine learning methods for this. The results of the classifiers are evaluated through prediction error analysis and several other performance measures. It is noticed from the results that the Support Vector Machine outperformed other classifiers. The superiority of the results is also established through statistical tests called Friedman test.

Keywords: Machine Learning, Multi-spectral Magnetic Resonance Images, Supervised Classifiers, Statistical Test.

1 Introduction

Machine learning is a kind of data processing technique that deals with developing program to learn from past data. Machine learning techniques helps us to solve highly complicated problems in a efficient way by formulating programs to imitate some of the facets of human mind [1]. Thus, the application of machine learning in intelligent computer programs improves the efficiency and accuracy in decisions making situations. Classification is the most widely used machine learning technique which is capable of separating non-overlapping data in to different segments. Therefore, classification is a process of finding a set of models which distinguishes class labels of different data objects [2].

However, the design of classifier is a crucial task in machine learning research. For a given classification task, the classifier considers both the complexity in it, as well as the size of the training dataset. Theoretically the optimal classifiers are not necessarily the best practical choice if they are offering higher complexity. It has been noticed that, the performance of the classifiers depends on the application and the information available for the given problem. Here, an overview of the application of four classifiers which includes Support Vector Machine (SVM) [3], k-Nearest Neighbor (k-NN) [4], Decision Tree (DT) [5] and Naive Bayesian (NB) [6, 7] are provided in Table 1. It shows the importance of classification methods and its wide range of application areas.

To further describe the application of classifiers and compare the performance of the aforementioned classifiers among themselves, a comparative study is required. Since, the classification of Magnetic Resonance (MR) brain images into different tissue classes is very important in clinical study and neurological pathology. Also the MR images are inherently noisy and imprecise in nature, hence, classification is a challenging task for these types of images. Here, a comparative study of MR brain image classification is performed with the use of machine learning methods like Support Vector Machine, k-Nearest Neighbor, Decision Tree and Naive Bayesian classifiers. The

⋆ Correspondence should be addressed to: Shib Sankar Bhowmick (shibsankar.ece@gmail.com) and Indrajit Saha (indra@icm.edu.pl).

Table 1. Summary of Applications of Classifiers in Engineering Problems

Area	Types of Classifier Applied with References
Feature extraction	SVM: [8–10], k-NN: [11], DT: [12]
Micro-array data analysis	SVM: [13–15], k-NN: [16], NB: [17].
Multi-sensor data fusion	SVM: [18], k-NN: [19], NB: [20], DT: [21].
Optimal power flow	SVM: [22], k-NN: [23], NB: [24].
Parameter estimation of chemical process	SVM: [25–27], k-NN: [28], DT: [29], NB: [30].
Remote sensing	SVM: [31], k-NN: [32], DT: [33], NB: [34].
Sentiment analysis	SVM: [35], k-NN: [36], DT: [37], NB: [38].
Signal processing	SVM: [39], k-NN: [40], DT: [41], NB: [42].

performance of these classifiers is demonstrated on several normal and multiple sclerosis lesion MR brain images. Effectiveness of these classification results is established quantitatively, visually and statistically.

The paper is organized as follows. Section 2 briefly describes the background of various classification techniques along with the overview of datasets. In Section 3, the performance of the classifiers are shown on several normal and multiple sclerosis lesion magnetic resonance brain images. Finally, Section 4 concludes the paper.

2 Methods and Materials

2.1 Machine Learning Methods

Support Vector Machine: The Support Vector Machine (SVM) is a state-of-the-art classification method introduced in 1992 by Boser *et al.* [3]. For a binary classification training data problem, suppose a data set consists of N feature vectors (x_i, y_i), where $y_i \in \{+1, -1\}$, denotes the class label for the data point x_i. The problem of finding the weight vector ν can be formulated as minimizing the following function:

$$L(\nu) = \frac{1}{2} \|\nu\|^2 \tag{1}$$

subject to

$$y_i[\nu \cdot \phi(x_i) + b] \geq 1, i = 1, \ldots, N \tag{2}$$

Here, b is the bias and the function $\phi(x)$ maps the input vector to the feature vector. The SVM classifier for the case on linearly inseparable data is given by

$$f(x) = \sum_{i=1}^{N} y_i \beta_i \mathcal{K}(x_i, x) + b \tag{3}$$

where $\mathcal{K}$ is the kernel matrix, and N is the number of input patterns having nonzero values of the Langrangian multipliers β_i. These N input patterns are called support vectors, and hence the name SVM. The Langrangian multipliers β_i can be obtained by maximizing the following:

$$Q(\beta) = \sum_{i=1}^{N} \beta_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j \beta_i \beta_j \mathcal{K}(x_i, x_j) \tag{4}$$

subject to

$$\sum_{i=1}^{N} y_i \beta_i = 0 \quad 0 \leq \beta_i \geq \mathcal{C}, \quad i = 1, ..., N \tag{5}$$

where $\mathcal{C}$ is the cost parameter, which controls the number of non separable points. Increasing $\mathcal{C}$ will increase the number of support vectors thus allowing fewer errors, but making the boundary separating the two classes more complex. On the other hand, a low value of $\mathcal{C}$ allows more non separable points, and therefore, has a simpler boundary. Only a small fraction of the β_i coefficients are nonzero. The corresponding pairs of x_i entries are known as support vectors and they fully define the decision function. Geometrically, the support vectors are the points lying near the separating hyperplane. $\mathcal{K}(x_i, x_j) = \phi(x_i).\phi(x_j)$ is called the kernel function. The kernel function may be linear or nonlinear, like Polynomial, Sigmoidal, Radial Basis Functions (RBF), etc. RBF kernels are of the following form:

$$\mathcal{K}(x_i, x_j) = e^{-\gamma|x_i - x_j|^2} \tag{6}$$

where x_i denotes the ith data point and γ is the weight. In this paper, the above mentioned RBF kernel is used. In addition, the extended version of the two-class SVM that deals with multiclass classification problem by designing a number of one against all two-class SVMs, is used here.

Naive Bayesian Classifier: The Naive Bayes (NB) classifier [6, 7] is developed based on the Bayes' theorem. It assumes that the attributes or features are conditionally independent for the given class label y to compute the class-conditional probability. Therefore, the assumption of conditional independence is defined as follows:

$$P(X|Y = y) = \prod_{i=1}^{n} P(X_i|Y = y), \tag{7}$$

where attribute set $\{X_1, X_2, ..., X_n\}^T$ consists of n attributes. Thereafter, it uses to compute the conditional probability of each X_i for given Y. In order to classify a test data, the classifier computes the posterior probability for each class Y and it is defined as follows:

$$P(Y|X) = \frac{P(Y)\prod_{i=1}^{n} P(X_i|Y)}{P(X)}. \tag{8}$$

Here, the posterior probabilities are computed by multiplying the priori probabilities with the class-conditional probabilities. The priori probability of each class is calculated by the fraction of training points that belong to each class.

***k*-Nearest Neighbor:** The traditional k-NN algorithm is well-known and widely used for its simplicity and easy implementation [4]. In k-NN classifiers, each unlabeled data point is classified by the majority voting of its k-nearest neighbors in the training set. Its performance thus depends crucially on the distance metric used to identify nearest neighbors. In the absence of prior knowledge, most k-NN classifiers use simple Euclidean metric to measure the distance between data points represented as vector inputs [43]. The class label assigned to a test data point is determined by the majority voting of its k nearest neighbors. For example, for the test data points, if we consider 5-NN algorithm and found 3 nearest neighbor data points are belonging in class c_1 and other 2 data points in class c_2, then the test data point should belong to class c_1.

Decision Tree: C4.5 [5] is a widely used Decision Tree generating algorithm and the extended version of ID3 algorithm. Both the algorithms have been developed by Ross Quinlan. Moreover, the Decision Trees generated by C4.5 are often used for classification, hence, it is also known as statistical classifier. To classify the data points, C4.5 uses the concept of entropy to build the Decision Trees from a set of training data. For this purpose, at every step, the highest information gained attribute is considered. Based on that attribute, decision is taken to split the training set into one or two subsets. The process will continue recursively until all nodes are exhausted. Thereafter, depending on user given parameters, C4.5 prunes the generated tree in order to classify the test data points.

2.2 Datasets

The MR Brain Images of normal brain and multiple sclerosis lesions brain are obtained from the Brainweb database [44]. The images are available in three bands: T1-weighted, T2-weighted and proton density (pd)-weighted. In our experiment, all bands are considered together for classification. The images correspond to the 1 mm slice thickness, 3% noise (relative to the brightest tissue) and with 20% intensity nonuniformity. The images of size 217×181 are available in 181 different Z planes. For the normal brain image data, the images of the Z planes Z10, Z60 and Z130 are considered. Similarly for the multiple sclerosis lesions brain image data the images of Z planes Z40, Z90 and Z140 are used for our experiments. The ground truth information of these images is also available at the Brainweb website [44]. From the ground truth information, it is observed that each of the Z planes Z10, Z60 and Z130 for normal brain images contains nine classes and for brain images of multiple sclerosis lesions, Z planes Z40, Z90 and Z140 are having classes of nine, eleven and nine, respectively.

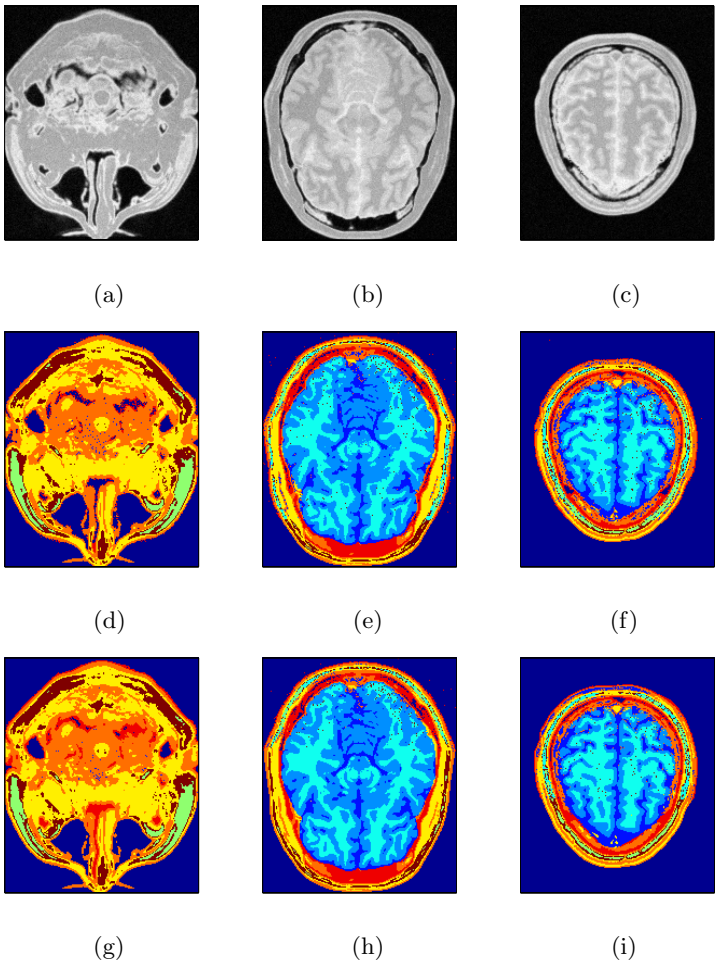


Fig. 1. (a), (b) and (c) are original T1-weighted MR images of the normal brains in Z10, Z60 and Z130 planes respectively, (d), (e) and (f) are MR images of the normal brains classified by SVM classifier in Z10, Z60 and Z130 planes respectively, and (g), (h) and (i) are MR images of the normal brain classified by k-NN classifier in Z10, Z60 and Z130 planes respectively.

3 Empirical Results

In this section, the experimental results of the compared machine learning methods are analyzed. For this purpose, different measures of the classifiers, i.e., prediction error analysis, evaluation of other validity measures like *Kappa*-Index (KI) [45], Minkowski Score (MS) [46] and Adjusted Rand Index (ARI) [47] as well as statistical tests of the prediction errors are discussed in following subsections.

In this experiment, the parameters of SVM such as γ for kernel function and the soft margin $\mathcal{C}$ (cost parameter), are set to be 0.5 and 2.0, respectively. Note that, RBF (Radial Basis Function) kernel is used here for SVM. The k value for the k-NN classifier is chosen as 13 for the satisfactory operation of the classifier and for the case of DT, C4.5 classifier is used.

3.1 Results and Discussions

We compare the performance of Machine Learning Methods like SVM, k-NN, C4.5 or DT and NB, in this section. As there are no separate training and testing data for the aforementioned images, hence these image pixels are randomly divided into 70% training dataset and 30% testing dataset to compute the error rate of each classifier.

Table 2. Average values of Prediction error (In %) of different Classifiers for MR brain images

MR Image		**Machine Learning Method**			
		SVM	***k*-NN**	**DT**	**NB**
Normal	**Z10**	10.39	10.47	11.29	10.74
Brain	**Z60**	11.18	11.19	11.26	11.46
	Z130	10.11	10.47	11.16	11.26
Multiple	**Z40**	18.89	19.53	19.77	19.25
Sclerosis	**Z90**	10.71	10.99	11.21	11.55
Lesion Brain	**Z140**	09.09	09.79	10.92	10.63

Table 2 shows the average results of prediction error produced by different classifiers for MR images of the above mentioned Z planes of normal and multiple sclerosis lesions brains. It is evident from the table that for all the images, the SVM classifier produces better average prediction error values compared to that produced by the other classifiers. It also appears that k-NN classifier perform reasonably good in terms of predicting average error values. Figures 1(a) to (c) show the original MR normal brain images in T1 band projected on Z10, Z60 and Z130 planes, respectively. Figures 1(d) to (f) and (g) to (i) show the segmented images of MR normal brain for SVM and k-NN on Z10, Z60 and Z130 planes, respectively. It appears from these figures that the SVM classifier has identified the different tissue classes of the normal brain images reasonably well.

On the other hand, Table 3 reports the average values of KI, MS and ARI of different classifiers for MR brain images. The KI, MS and ARI values are also found better for SVM. Moreover, it is observed that the results of SVM and k-NN are superior in their corresponding groups while the SVM performs better than the k-NN. Figures 2(a) to (c) show the T1-weighted original images corresponding to the Z planes Z40, Z90 and Z140 for the multiple sclerosis lesions brain image data. The corresponding segmented images obtained by SVM and k-NN classifier are also shown in Figures 2(d) to (f) and (g) to (i), respectively. It is clear from the figures that the SVM classifier has identified the different homogeneous regions of the images very well. Hence, form the above quantitative and visual results for both the brain image datasets it is evident that the SVM classifier outperforms all its competitors.

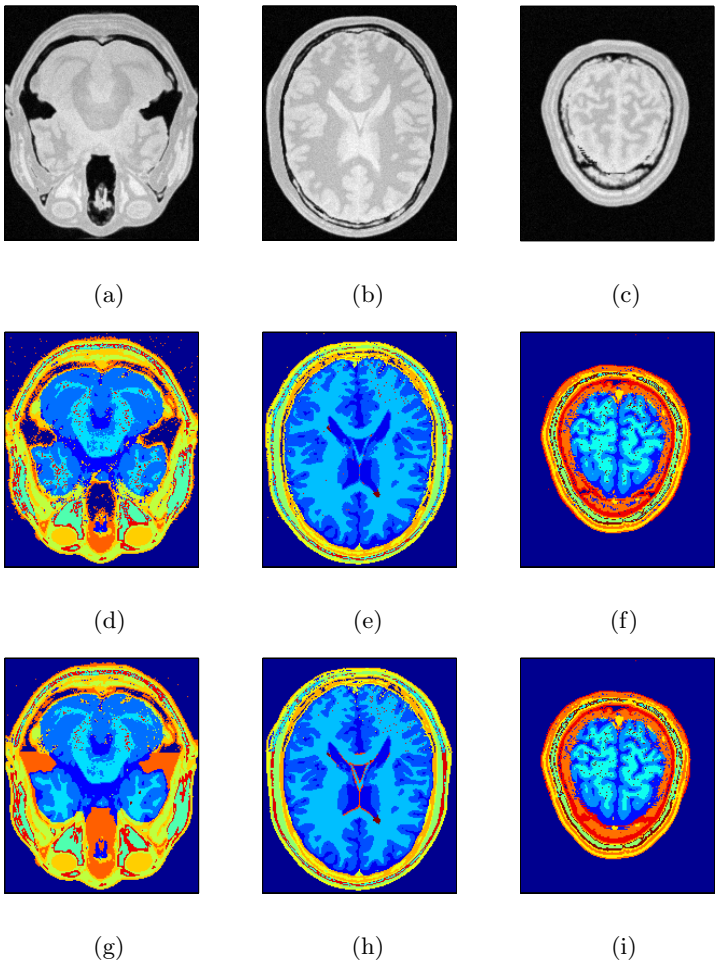


Fig. 2. (a), (b) and (c) are original T1-weighted MR images of the multiple sclerosis lesions brains in Z40, Z90 and Z140 planes respectively, (d), (e) and (f) are MR images of the multiple sclerosis lesions brains classified by SVM classifier in Z40, Z90 and Z140 planes respectively, and (g), (h) and (i) are MR images of the multiple sclerosis lesions brains classified by k-NN classifier in Z40, Z90 and Z140 planes respectively.

Table 3. Average values of KI, MS and ARI over 20 runs of different classifiers for MR brain images

Classifier	**Normal Brain**									**Multiple Sclerosis Lesion Brain**								
	Z10			**Z60**			**Z130**			**Z40**			**Z90**			**Z140**		
	KI	**MS**	**ARI**	**KI**	**MS**	**ARI**	**KI**	**MS**	**ARI**	**KI**	**MS**	**ARI**	**KI**	**MS**	**ARI**	**KI**	**MS**	**ARI**
SVM	0.83	0.45	0.67	0.86	0.39	0.80	0.85	0.45	0.67	0.77	0.48	0.61	0.85	0.44	0.71	0.87	0.39	0.80
***k*-NN**	0.87	0.39	0.80	0.87	0.37	0.80	0.87	0.37	0.80	0.81	0.44	0.71	0.89	0.36	0.81	0.08	0.39	0.80
DT	0.82	0.44	0.71	0.83	0.45	0.68	0.78	0.45	0.68	0.77	0.47	0.66	0.81	0.45	0.68	0.76	0.48	0.61
NB	0.89	0.34	0.85	0.86	0.39	0.80	0.84	0.45	0.67	0.82	0.44	0.71	0.87	0.39	0.80	0.86	0.39	0.80

3.2 Statistical Significance Test

Statistical significance of the results produced by different classifiers, are analyzed at here. For this purpose, Friedman test [48, 49] is conducted. Generally, Friedman test ranks the classifiers for each dataset separately. To compute the average rank $\mathcal{R}_j$, let r_i^j be the rank of the jth algorithm for ith dataset where the number of datasets and algorithms are N and Q respectively. Therefore the average rank $\mathcal{R}_j = \frac{1}{N}\sum_i r_i^j$.

Under the null-hypothesis, which states that all the algorithms are equivalent and so their ranks $\mathcal{R}_j$ should be equal. The Friedman statistic (chi square value) is computed as follows:

$$\chi_F^2 = \frac{12N}{Q(Q+1)} \left[\sum_j \mathcal{R}_j^2 - \frac{Q(Q+1)^2}{4} \right] \tag{9}$$

The Friedman statistic is distributed according to χ_F^2 with $Q-1$ degrees of freedom, when $N > 10$ and $Q > 5$. For a smaller number of algorithms and data sets, exact critical values are computed [50, 51].

Table 4. The Friedman ranks of all classifiers for MR brain images

MR Image		**Machine Learning Method**			
		SVM	***k*-NN**	**DT**	**NB**
Normal Brain	**Z10**	3	2.5	4	3.5
	Z60	3.5	2.5	3	4
	Z130	2	2.5	3.5	4
Multiple Sclerosis	**Z40**	2.5	3.5	4	3
	Z90	2.5	3	3.5	4
Lesion Brain	**Z140**	2	3	4	3.5
Average Rank		**2.583**	**2.833**	**3.666**	**3.666**

Table 4 reports the ranks of different classifiers for different images as well as average ranks for each classifier. From Friedman test the average rank for the classifiers SVM, k-NN, DT and NB are computed as 2.583, 2.833, 3.666 and 3.666, respectively. Moreover, from this average ranks using Equation 9, χ_F^2 is computed as 31.693. Therefore, its corresponding p value is 0.11×10^{-4} at $\alpha = 0.05$ significance level, which emphasize the acceptance of alternative hypothesis strongly. So, the results produced by the SVM are statistically significant.

4 Conclusion

In this paper, a comparative study of various machine learning methods for multispectral Magnetic Resonance brain images is conducted. For the machine learning methods, Support Vector Machine, k-Nearest Neighbor, Naive Bayesian and C4.5 or Decision Tree are used. The classification results reveals that the average values of prediction errors produced by the Support Vector Machine are better than the other classifiers. The investigation of *Kappa*-Index, Minkowski Score and Adjusted Rand Index indicates the same for Support Vector Machine. Furthermore, statistical test also shows that the average error values produced by Support Vector Machine are statistically significant. Finally, considering all conducted tests and statistics, it is established that the results of Support Vector Machine are quantitatively, visually and statistically superior than other three classifiers.

Acknowledgment

This work is partially supported by Erasmus Mundus Mobility with Asia (EMMA) grant 2012 from the European Union at the Department of Informatics, University of Evora in Portugal.

References

1. Lin, Y., Wu, M., Bloom, J.A., Cox, I.J., Miller, M.: Rotation, scale, and translation resilient public watermarking for images. IEEE Transactions on Image Processing **10**(5) (2001) 767–782
2. Chaudhary, A., Kolhe, S., Rajkamal: Machine learning techniques for mobile intelligent systems: A study. In: Proceedings of the IEEE International conference on Wireless and Optical Communications Networks (2012)
3. Boser, B.E., Guyon, I.M., N.Vapnik, V.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory (1992) 144–152
4. Sun, S.L.: Ensembles of feature subspaces for object detection. Lecture Notes in Computer Science **5552** (2009) 996–1004
5. Quinlan, J.R.: C4.5: Programs for machine learning. Morgan Kaufmann Publishers, San Francisco, USA (1993)
6. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. John Wiley & Sons, New York (1973)
7. Langley, P., Iba, W., Thompson, K.: An analysis of bayesian classifiers. In: Proceedings of the International Conference on Artificial Intelligence (1992)
8. Saha, I., Maulik, U., Bandyopadhyay, S., Plewczynski, D.: SVMeFC: SVM ensemble fuzzy clustering for satellite image segmentation. IEEE Geoscience and Remote Sensing Letters **9**(1) (2011) 52–55
9. Saha, I., Mukhopadhyay, A.: Improved crisp and fuzzy clustering techniques for categorical data. IAENG International Journal of Computer Science **35**(1) (2008) 438–450
10. Saha, I., Maulik, U., Plewczynski, D.: A new multi-objective technique for differential fuzzy clustering. Applied Soft Computing **11**(2) (2011) 2765–2776
11. Balamurugan, A.A., Rajaram, R., Pramala, S., Rajalakshmi, S., Jeyendran, C., Prakash, J.D.S.: Nb+: An improved nave bayesian algorithm. Knowledge-Based Systems **24**(5) (2011) 563–569
12. Suresh, S., Sundararajan, N., Saratchandran, P.: Risk-sensitive loss functions for sparse multi-category classification problems. Information Sciences **178**(12) (2008) 2621–2638
13. Saha, I., Maulik, U., Bandyopadhyay, S., Plewczynski, D.: Improvement of new automatic differential fuzzy clustering using svm classifier for microarray analysis. Expert Systems with Applications **38**(12) (2011) 15122–15133
14. Saha, I., Plewczynski, D., Maulik, U., Bandyopadhyay, S.: Improved differential evolution for microarray analysis. International Journal of Data Mining and Bioinformatics **6**(1) (2012) 86–103
15. Saha, I., Maulik, U., Bandyopadhyay, S., Plewczynski, D.: Unsupervised and supervised learning approaches together for microarray analysis. Fundamenta Informaticae **106**(1) (2011) 45–73
16. Liaw, Y.C., Leou, M.L., Wu, C.M.: Fast exact k nearest neighbors search using an orthogonal search tree. Pattern Recognition **43**(6) (2010) 2351–2358
17. Luo, H., Puthusserypady, S.: A sparse bayesian method for determination of flexible design matrix for fmri data analysis. IEEE Transactions on Circuits and Systems I: Regular Papers **52**(12) (2005) 2699–2706
18. Shaopeng, L., Gao, R.X., John, D., Staudenmayer, J., Freedson, P.S.: SVM-based multi-sensor fusion for free-living physical activity assessment. In: Proceedings of Annual International Conference of the IEEE for Engineering in Medicine and Biology Society (2011) 3188–3191
19. Liu, L., Yang, S., Wang, D.: Force-imitated particle swarm optimization using the near-neighbor effect for locating multiple optima. Information Sciences **182**(1) (2012) 139–155
20. Chair, Z., Varshney, P.K.: Distributed bayesian hypothesis testing with distributed data fusion. IEEE Transactions on Systems, Man and Cybernetics **18**(5) (1988) 695–699
21. Pal, M., Mather, P.M.: An assessment of the effectiveness of decision tree methods for land cover classification. Remote Sensing of Environment **86**(4) (2003) 554–565
22. Omitaomu, O.A., Jeong, M.K., Badiru, A.B., Hines, J.W.: Online support vector regression approach for the monitoring of motor shaft misalignment and feedwater flow rate. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews **37**(5) (2007) 962–970
23. Guohui, L., Yanhong, L., Jianjun, L., Shu, L., Fumin, Y.: Continuous reverse k nearest neighbor monitoring on moving objects in road networks. Information Systems **35**(8) (2010) 860– 83
24. Jianchao, H., fu, T.Z., Bin, W.M., Haiyang, J.: The optimal bidding models for power energy producer based on static bayesian game theory. In: Proceedings of Conference of Control and Decision (2008) 3123–3128
25. Saha, I., Mazzocco, G., Plewczynski, D.: Consensus classification of human leukocyte antigens class ii proteins. Immunogenetics **65**(2) (2013) 97–105

26. Saha, I., Maulik, U., Bandyopadhyay, S., Plewczynski, D.: Fuzzy clustering of physicochemical and biochemical properties of amino acids. Amino Acids **43**(2) (2012) 583–594
27. Plewczynski, D., Basu, S., Saha, I.: AMS 4.0: Consensus prediction of post-translational modifications in protein sequences. Amino Acids **43**(2) (2012) 573–582
28. Tran, T.N., Wehrens, R., Buydens, L.M.G.: Knn density-based clustering for high dimensional multispectral images. Second GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas (2003) 147–151
29. Mugambi, E.M., Hunter, A., Oatley, G., Kennedy, L.: Polynomial-fuzzy decision tree structures for classifying medical data. Knowledge-Based Systems **17**(2-4) (2004) 81–87
30. Xu, W., Baggeroer, A.B., Richmond, C.D.: Bayesian bou nds for matched-field parameter estimation. IEEE Transactions on Signal Processing **52**(12) (2004) 3293–3305
31. Bruzzone, L., Mingmin, C., Marconcini, M.: A novel transductive svm for semisupervised classification of remote-sensing images. IEEE Transactions on Geoscience and Remote Sensing **44**(11) (2006) 3363–3373
32. Zhu, H., Basir, O.: An adaptive fuzzy evidential nearest neighbor formulation for classifying remote sensing images. IEEE Transactions on Geoscience and Remote Sensing **43**(8) (2005) 1874–1889
33. Moustakidis, S., Mallinis, G., Koutsias, N., B.Theocharis, J., Petridis, V.: SVM-based fuzzy decision trees for classification of high spatial resolution remote sensing images. IEEE Transactions on Geoscience and Remote Sensing **50**(1) (2012) 149–169
34. Davis, D.T., Chen, Z., Hwang, J.N., Tsang, L., Njoku, E.: Solving inverse problems by bayesian iterative inversion of a forward model with applications to parameter mapping using smmr remote sensing data. IEEE Transactions on Geoscience and Remote Sensing **33**(5) (1995) 1182–1193
35. Huang, Z., Chen, H., Hsu, C.J., Chen, W.H., Wu, S.: Credit rating analysis with support vector machines and neural networks: a market comparative study. Decision Support Systems **37**(4) (2004) 543–558
36. Fan, W., Sun, S.: Sentiment classification for online comments on chinese news. In: Proceedings of International Conference of Computer Application and System Modeling **4** (2010) 740–745
37. Sethi, I.K., Yoo, J.H.: Design of multicategory multifeature split decision trees using perceptron learning. Pattern Recognition **27**(7) (1994) 939–947
38. Ghorpade, T., Ragha, L.: Featured based sentiment classification for hotel reviews using nlp and bayesian classification. In: Proceedings of International Conference of Communication, Information Computing Technology (2012) 1–5
39. Tran, Q.A., Li, X., Duan, H.: Efficient performance estimate for one-class support vector machine. Pattern Recognition Letters **26**(8) (2005) 1174–1182
40. Wu, Y., Ianakiev, K., Govindaraju, V.: Improved k-nearest neighbor classification. Pattern Recognition **35**(10) (2002) 2311–2318
41. Rivera, F., Sanchez-Elez, M., Fernandez, M., Hermida, R., Bagherzadeh, N.: Efficient mapping of hierarchical trees on coarse-grain reconfigurable architectures. In: Proceedings of International Conference of Hardware/Software Codesign and System Synthesis (2004) 30–35
42. hua, T.J.: Research of vehicle video image recognition technology based on naive bayesian classification model. In: Proceedings of Third International Conference of Information and Computing **2** (2010) 17–20
43. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. The Journal of Machine Learning Research **10** (2009) 207–244
44. Brainweb: Simulated brain database. `http://www.bic.mni.mcgill.ca/brainweb`
45. Cohen, J.A.: Coefficient of agreement for nominal scales. Educational and Psychological Measurement **20**(1) (1960) 37–46
46. Jardine, N., Sibson, R.: Mathematical Taxonomy. John Wiley and Sons, New Jersey, USA (1971)
47. Yeung, K.Y., Ruzzo, W.L.: An empirical study on principal component analysis for clustering gene expression data. Bioinformatics **17** (2001) 763–774
48. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association **32** (1937) 675–701
49. Friedman, M.: A comparison of alternative tests of significance for the problem of m rankings. Annals of Mathematical Statistics **11** (1940) 86–92
50. Zar, J.H.: Biostatistical Analysis (4th Edition). Prentice Hall, New Jersey, USA (1998)
51. Sheskin, D.J.: Handbook of parametric and nonparametric statistical procedures. Chapman & Hall/CRC, London, UK (2000)

Calibração e modelo de um canal automático experimental

José Duarte and Luís Rato

Universidade de Évora
d10401@alunos.uevora.pt, lmr@uevora.pt

1 Resumo

O canal experimental de larga escala do Núcleo de Hidráulica e Controlo de Canais (NuHCC) da Universidade de Évora é uma infraestrutura automatizada com cerca de 140 metros de extensão, composta por 6 autómatos do tipo PLCs (Programmable Logic Controller), diversos sensores e atuadores[1]. Está equipada com um sistema SCADA (Supervisory Control And Data Acquisition) que torna possível a monitorização e controlo, em tempo real, de todo o canal. Trata-se de um *software* proprietário da WIZCOM - de 2002 - e com algumas limitações em termos de comunicação com outras aplicações e/ou sistemas operativos. Para colmatar estas lacunas foi criado, em 2011, no âmbito de um projecto FCT[1], o Interface Multiplataforma para sistemas SCADA (imSCADA)[1]. Desenvolvido com o intuito de permitir uma comunicação direta entre qualquer ferramenta numérica (pe. Matlab, Simulink, Octave, etc) ou linguagem de programação, a única condição necessária é que esta permita comunicação através de *sockets*[2]. Diversos projetos de investigação das áreas de controlo e de decisão inteligente [1,2] têm sido realizados nesta infraestrutura.

A nossa investigação foca-se no estudo de Reconhecimento de Padrões para Deteção de Falhas em sistemas em tempo real. Diversos testes têm sido realizados, através do Matlab com a ajuda do imSCADA, com o intuito de calibrar uma das quatro comportas do referido canal. Trata-se de uma comporta vertical plana e retangular, é atuada por um motor elétrico e garante o fluxo de água através da abertura inferior, ou seja, a comporta sobe verticalmente para deixar a água passar. Quando o valor da abertura da comporta é igual a 0 milímetros significa que esta está totalmente fechada e que bloqueia totalmente a progressão da água. Quando se encontra a uma altura de 800 milímetros é considerada totalmente aberta. Os testes foram divididos em duas categorias, os de abertura total e os de abertura parcial. Os testes de abertura total implicam abrir ou fechar totalmente a comporta (ie. variações entre 0 e 800 mm), os parciais representam apenas mudanças parciais no estado da comporta (pe. subir ou descer a comporta 400 mm). Ao longo do percurso, definido pelo teste em causa, os valores (intermédios e totais) da altura a que comporta se encontra e o tempo que esta demora a deslocar-se, são registados. As cerca de 5 horas de testes realizados com amostras de tempo de na ordem de 1 segundo já revelam algumas características do comportamento da comporta, mas mais testes são necessários de forma a garantir a criação de um modelo bem ajustado.

Uma das aplicações deste trabalho experimental é a de, através da deteção de pequenos desvios no padrão de comportamento das comportas, serem emitidos alertas sobre falhas e avisos de manutenção preventiva do canal. A instalação de um SCADA *opensource*, nomeadamente o ScadaBR, irá permitir não só atualizar e melhorar o desempenho geral do sistema mas também possibilitará a integração tanto do imSCADA como do Sistema de Deteção de Falhas.

2 Agradecimentos

Gostaríamos de agradecer aos revisores e ao Prof. Doutor Manuel Rijo por todas as criticas, sugestões e comentários que nos permitiram melhorar o trabalho desenvolvido.

[1] AQUANET - Decentralised and Reconfigurable Control for Water delivery Multipurpose Canal Systems (PTDC/EEA-CRO/102102/2008)

[2] ORCHESTRA - Distributed Optimization and Control of Large Scale Water Delivery Systems, (PTDC/EMS-CRO/2042/2012)

Referências

[1] J. Duarte, L. Rato, and M. Rijo. Calibração e teste de modelos de controlo automático num canal de adução experimental. In *Proceedings do CLME'2011 / IIICEM - 6º Congresso Luso-Moçambicano de Engenharia, 3º Congresso Moçambicano de Engenharia*. Maputo, Moçambique, 2011.

[2] J. Duarte, L. Rato, P. Shirley, and M. Rijo. Multi-platform controller interface for scada application. In *Proceedings of 18th IFAC World Congress (IFAC2011)*, volume 18, pages 7885–7890. IFAC, Milan, Italy, 2011.

3pi: Primeiros Passos

Marlene Oliveira, João Aiveca, Ricardo Dias

Universidade de Évora, Portugal
m11327@alunos.uevora.pt, m10712@alunos.uevora.pt, m11246@alunos.uevora.pt

Resumo Hoje em dia a Robótica tem um papel cada vez mais dominante na sociedade. Nesta área a exigência técnica é elevada, com grande abrangência de domínios como física, mecânica, entre outros. Procura-se obter uma abordagem simples para a introdução a esta temática, indiferente ao conhecimento prévio, utilizando pequenos robôs que seguem linhas.

Keywords: 3pi, Robótica, Robô, Calibração, Controlador

1 Introdução

A Robótica, "ramo da tecnologia que lida com o design, construção, operação e aplicação de robôs"[20], está cada vez mais presente no quotidiano de todos nós. Carros, eletrodomésticos, smartphones e PC's, são construídos por robôs ou contêm componente(s) relacionado(s) com robótica.

Segundo o *Robot Institute of America*[17], um robô é "um manipulador multifuncional, reprogramável concebido para mover material, partes, ferramentas, ou dispositivos especializados através de vários movimentos programados para o desempenho de uma variedade de tarefas"[14], existindo vários tipos (robô com rodas, veículos aéreos, etc). Estes agentes desempenham algumas tarefas que são perigosas, aborrecidas, repetitivas, stressantes ou trabalhosas para os humanos (por exemplo, robôs aspiradores que aspiram a casa, poupando tempo)[15]. Os robôs são ainda utilizados em investigação, existindo inclusive diversas universidades portuguesas, como o Instituto Superior Técnico (IST), e estrangeiras, como Carnegie Mellon University, que possuem equipas de investigação dedicadas a temáticas relacionadas com Robótica. Para além disto, existem também algumas competições de Robótica, como a *Robocup*, que visam aproximar o público geral da Robótica. Algumas empresas[19] estão também a apostar mais neste ramo, desenvolvendo produtos inovadores. Na Universidade de Évora não existe atualmente uma equipa dedicada à Robótica, porém alunos e professores juntaram esforços para começar a explorar este tema.

Os *3pi* são robôs seguidores de linhas, isto é, são robôs móveis guiados automaticamente através de sensores óticos capazes de seguir uma linha com uma cor distinta da cor da superfície onde se encontram[3]. A origem do nome deste robô produzido pela Pololu deve-se ao seu diâmetro de, aproximadamente, três vezes a medida matemática[13] π. O *3pi* é normalmente utilizado na introdução de iniciantes aos paradigmas da robótica, em competições para a resolução de labirintos e para seguir linhas[12].

A *Pololu* tem à disposição um kit[6] que contém o robô *3pi*, já pronto a utilizar, e um programador *Alf and Vegar RISC*(AVR)[4] indispensável para, tal como o nome indica, ser possível programar o *3pi*. Este será o modelo objeto de estudo.

O *3pi* pode ser melhorado através da instalação de alguns componentes extra (denominados "shields") que lhe conferem mais funcionalidades, que podem ser adquiridos em conjunto com o robô ou separadamente, por exemplo, placas de expansão e módulos para comunicação wireless.

Existem algumas alternativas ao *3pi*, tais como *Microbot Robot Kit*[16] e o *Lego mindstorms*.

2 Especificações Técnicas

Existem vários componentes no *3pi* que permitem obter e processar informação. Estes robôs incluem um microcontrolador com uma frequência de 20MHz: ATmega328p [2].

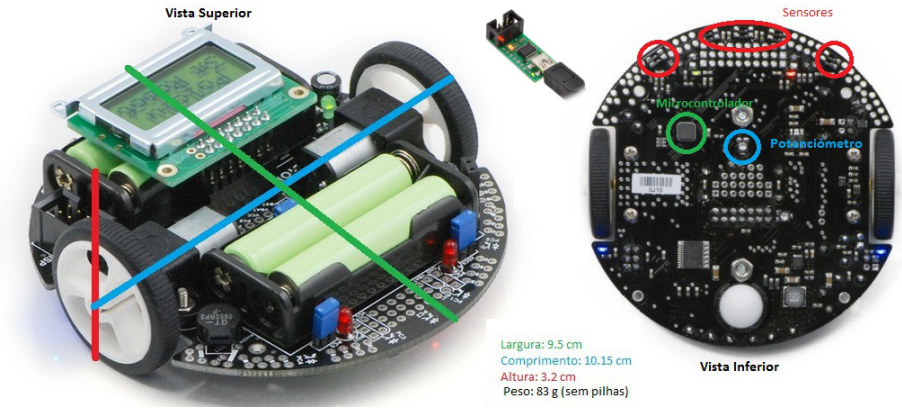


Figura 1. Dados físicos e localização dos componentes mais relevantes do *3pi* e AVR.

O *3pi* que possui o microcontrolador ATmega328p tem as seguintes características no que respeita a memória: 32KB de memória flash de programas, 2KB de RAM e 1KB de memória persistente *EEPROM**[1]. O microcontrolador está assinalado a verde na vista inferior do robô, que consta da figura 1.

Na zona inferior do *3pi* existem cinco sensores que indicam o nível de luz reflectido sob a forma de um valor, sendo zero totalmente branco e 4000 totalmente preto[11]. Os sensores estão assinalados a vermelho na vista inferior do robô, que consta da figura 1.

O *3pi* possui ainda um potenciómetro, que pode ser ajustado para corrigir pequenos problemas. O potenciómetro está assinalado a azul na vista inferior do robô, que consta da figura 1.

O ecrã do *3pi* é um LCD de 8x2 caracteres que está localizado na zona superior. Sob este ecrã existe um compartimento para colocar duas das quatro pilhas AAA que alimentam o *3pi*. A localização do ecrã e das pilhas, bem como as características físicas podem ser observadas na figura vista superior do robô, que consta da figura 1.

3 Instalação e Configuração

O ambiente de desenvolvimento para o *3pi* tem por base o IDE *Arduino*[7]. Após o *download* do executável (versão 1.0.5 na escrita deste documento), devem seguir-se os passos no ecrã. Dos componentes a instalar, são obrigatórios o USB driver e o Arduino Software, sendo as três outras opções facultativas.

Após instalação, o IDE fica disponível; não está, no entanto, pronto para comunicar com o *3pi*. Para tal é necessário incluir as bibliotecas disponíveis no site da Pololu[7], versão 120914. O conteúdo do ficheiro obtido deverá ser extraído para o local onde foi instalado o IDE (por defeito, em *Windows*, *C:\Program Files (x86)\Arduino*, daqui em diante referido como "diretório do *Arduino*"), na pasta *libraries*. Pode verificar-se a correta instalação das bibliotecas acedendo a *File>Examples>Polulu3pi* (podendo ser necessário reiniciar o IDE caso a cópia das bibliotecas se tenha efetuado enquanto este estava a ser executado). É ainda necessário editar o conteúdo dos ficheiros *board.txt* e *programmers.txt*, localizados no diretório do *Arduino*, na pasta *hardware/arduino*. Aos ficheiros *board.txt* e *programmers.txt* devem ser acrescentadas as seguintes linhas de código apresentadas na figura 2. Utilizadores de *Windows* devem ter em conta que, se utilizaram a directoria pré-definida de instalação, necessitam de executar um editor de texto como administrador para conseguir editar o conteúdo dos ficheiros. De notar que este código se aplica apenas ao modelo *328p*, devendo ser diferente quando aplicados a outros modelos[5]. O *3pi* está pronto para ser ligado à porta USB do computador. Durante alguns segundos, o *3pi* é instalado. Quando terminado, se a luz intermitente do programador AVR estiver laranja, está pronto a receber *uploads*

[1] Electrically ErasableProgrammable Read-Only Memory

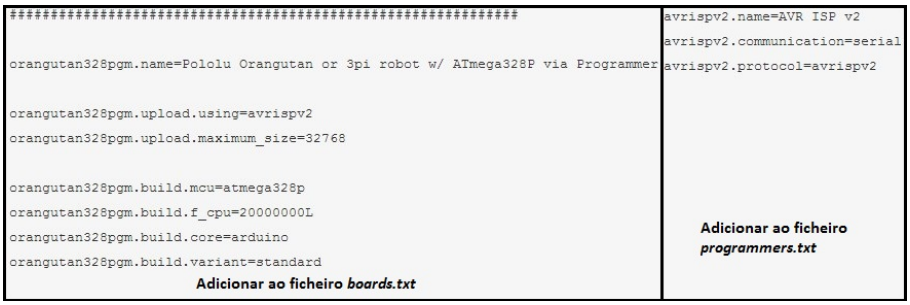


Figura 2. Linhas que devem ser acrescentadas aos ficheiros *board.txt* e *programmers.txt*.

de programas do utilizador. Se a luz intermitente se apresentar vermelha, não deve ser feito *upload* a nenhum programa. Atenção: o computador ligado ao *3pi* deve ter uma fonte segura de energia. Quebras de corrente durante um *upload* podem causar danos irreversíveis no equipamento. Devem definir-se as opções para preparar um *upload*. No menu *Tools>Board*, a opção correta é *Polulu Orangutan or 3pi robot w/ Atmega328P via Programmer*. No menu *Tools>Serial Port*, a opção correta é a correspondente à porta COM identificada como *Pololu USB AVR Programmer Programming Port*. Para identificar esta porta, deve procurar-se no Gestor de Dispositivos (*Windows*), na subpasta Portas (COM e LPT) como está na figura 3. Com todas as definições finalizadas, o dispositivo está pronto a receber código. Na barra de ferramentas, o primeiro botão (✓) permite detectar erros de sintaxe e compilar o código. O botão (→), além de compilar o projecto, envia o

Figura 3. Passos para verificar a porta COM.

código para o dispositivo. A cor dos LEDs do programador AVR alterna durante o *upload*, como pode ser verificado na figura 1.

O *3pi* está agora preparado para ser utilizado, faltando apenas criar uma pista, cujas linhas serão percorridas pelo robô. Sugere-se o uso de uma cartolina branca e fita isoladora preta para construir a pista, sendo pouco aconselhável o uso de uma configuração com curvas demasiado apertadas, dado que isso pode afectar o desempenho do *3pi*[10].

4 Calibração

A calibração consiste num processo que permite melhorar a precisão na localização do robô, fazendo uso de software e não alterando o design ou estrutura mecânica do robô [21]. O *3pi* possui duas rodas independentes[8], o que permite um tipo de locomoção denominado *differential drive*. Descrever uma curva quando este é o tipo de locomoção faz-se movendo os dois motores a velocidades diferentes[8]. Uma vez que as caraterísticas físicas variam ligeiramente de robô para robô e que pode existir desgaste do equipamento, é necessário proceder à calibração de modo a que seja possível ajustar, no código dos programas, a velocidade dos motores para que estes se movam a velocidades muito semelhantes, fazendo com que o movimento do robô seja relativamente uniforme[21]. A não realização deste processo poderá levar a que o *3pi* não se mova em linha reta[8]. Por exemplo, se o motor da direita tiver uma velocidade menor que o da esquerda, o *3pi* curva para a direita.

4.1 Metodologia

Começou-se por programar o robô para manter um motor parado enquanto o outro se encontra em funcionamento durante um determinado intervalo de tempo fixo, dado que assim o robô giraria sobre si próprio. Para efetuar a calibração dos dois motores utilizou-se o seguinte procedimento:

- Colocar uma marca em cima da qual ficará a roda cujo motor se encontra parado;
- Colocar uma marca em cima da qual ficará a roda cujo motor se encontra em funcionamento;
- Ligar o robô e contar quantas vezes a roda do motor que se encontra em funcionamento passa na marca criada anteriormente, isto é, medindo quantas voltas sobre si próprio o robô descreve;
- Repetir o processo com várias velocidades diferentes para o motor atual.

Os dados recolhidos permitem calcular a distância percorrida pelo *3pi* durante um dado intervalo de tempo, isto é, a velocidade real a que o robô se desloca. O tempo pode ser abordado de duas formas: fixo para todos os testes (voltas incompletas são descartadas; dado que esta alternativa acrescenta algum erro, é aconselhável utilizar tempos mais elevados (acima de 10 segundos) para minimizar o seu impacto), ou cronometrar voltas completas (também incorre algum erro do tempo de resposta humana, mas menor que a outra alternativa).

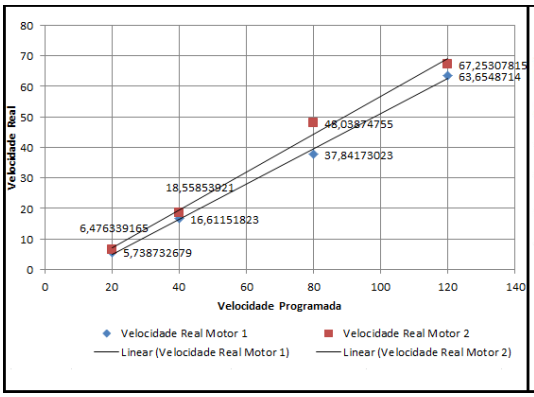
Dado que o perímetro da circunferência descrita pela roda em movimento corresponde à distância percorrida numa volta ($D(n)$), utiliza-se a fórmula de cálculo do perímetro da circunferência para obter este valor, como se verifica na fórmula1 (com o raio do robot (r) igual a 4.75 cm, ou seja $0,0475\,m$, e n correspondente ao número de voltas).

$$D(n) = 2 * \pi * r \tag{1}$$

A distância percorrida (Dt) foi calculada utilizando 2.

$$Dt = n * D(n) \tag{2}$$

Com a informação referente à distância percorrida e ao tempo, foi então possível calcular a velocidade real de cada motor, que difere um pouco da velocidade programada. O passo seguinte foi calcular as médias destas velocidades e marcar os pontos correspondentes num gráfico velocidade programada / velocidade real e traça-se a reta de regressão. As velocidades reais obtidas estão presentes na tabela da figura 4, o gráfico resultante também pode ser observado nesta figura.



Velocidade Programada	Média Velocidade Real (Motor 1)	Média Velocidade Real (Motor 2)
20	5,738732679	6,476339165
40	16,61151823	18,55853921
80	37,84173023	48,03874755
120	63,6548714	67,25307815

Figura 4. Gráfico velocidade real/velocidade programada e retas de regressão para as velocidades reais dos motores e tabela das médias das velocidades reais.

Como os motores não são inteiramente homogéneos, as linhas da regressão linear sobrepor-se-iam, com declives idênticos. Há também ruído nas medições; as linhas da regressão linear deveriam ser paralelas. Para obter a diferença entre os motores, divide-se o motor mais rápido pelo mais

lento; pode ser necessário algum ajuste manual caso a diferença entre os declives seja demasiada. A velocidade do motor 2 deverá ser dada pela fórmula 3, em que v_1 é a velocidade medida no primeiro motor e d_1 e d_2 são os declives das retas.

$$v2 = v1 * Coef\ (com\ Coef = \frac{d_1 + d_2}{d_1}) \tag{3}$$

5 Controladores

O *Proporsional Integral Derivate* (*PID*) é um controlador ideal para uma primeira abordagem com o *3pi*, podendo experimentalmente atingir valores satisfatórios[18].

5.1 Componente *Proportional*

A componente Proporcional controla a resposta do sistema à diferença entre os valores medidos e os valores obtidos; um valor excessivo descontrola o sistema, e um valor demasiado baixo não permite que este se ajuste a tempo às flutuações entre o valor indicado pelos sensores e o valor real. Um ganho K, multiplicado pelo erro P (a diferença entre o valor real e o medido), vai definir o ajuste à velocidade dos motores. A nossa implementação [1] considera apenas esta componente.

5.2 Componentes *Integral* e *Derivativo*

Uma extensão do algoritmo P, o PID, considera duas novas componentes: *Integral*, e *Derivativo*. De forma simplificada, procura compensar os erros decorrentes da leitura de sensores com maior precisão que o P. Numa implementação em *Arduino* para o *3pi* [9] (No IDE do *Arduino*, o exemplo encontra-se em *File->Examples->Pololu3pi->PID3piLineFollower*, na linha 226), os valores dos parâmetros são ajustados na equação 4.

$$powerDifference = \frac{proportional}{20} + \frac{integral}{10000} + derivative * \frac{3}{2} \tag{4}$$

Os valores $\frac{1}{20}$, $\frac{1}{10000}$ e $\frac{3}{2}$ são ajustáveis, dando maior ou menor peso a cada componente. A resposta do sistema ao aumento do valor destes parâmetros pode resumir-se em quatro medidas de desempenho: o tempo que o sistema demora a atingir o objetivo, o quanto o sistema passa além do objetivo, o tempo de estabilizar, e o erro quando num estado estável. No geral, aumentando a componente *P*, descem o erro num estado estável e o tempo que o sistema leva a atingir o objetivo, subindo o quanto o sistema ultrapassa o objetivo [22]; aumentando *I*, obtém-se o mesmo efeito no tempo que o sistema demora a atingir o objetivo e o quanto passa além do objetivo; aumenta-se ainda o tempo de estabilização, mas elimina-se o erro quando está num estado estável [22]. Finalmente, aumentando *D*, descem o quanto o sistema passa além do objetivo e o tempo de estabilização [22].

6 Conclusão

O objetivo deste artigo consistiu em fornecer a um utilizador relativamente inexperiente um meio que seja simples o suficiente para que este consiga configurar e utilizar pequenos exemplos com o *3pi*. A possibilidade de inovar com um sistema de controlo tão simples como o PID é bastante larga. É suficientemente desafiante, uma vez que alterações às variáveis são complexas e bastante voláteis. Passadas as dificuldades iniciais de instalação/calibração, é imediato o *feedback* do *3pi*, e os resultados são simples de replicar. No entanto, há que realçar que é necessário um conhecimento mais aprofundado para continuar em frente, quer a nível de modelação matemática do controlo PID, quer a nível de progressão para algoritmos mais complexos.

No futuro, seria interessante criar uma base de conhecimento mais aprofundado, capaz de fomentar a continuação de projetos nesta área. Tutoriais com diversos níveis de dificuldade seriam

interessantes para proporcionar continuidade a potenciais leitores.

A nível de aprofundamento pessoal, e como pesquisa para a realização do tutorial, seria interessante a criação de um simulador 3D capaz de proporcionar um ambiente de desenvolvimento mesmo àqueles que não disponham de um *3pi*. Seria também interessante reforçar o conhecimento teórico dos algoritmos de controlo, visto que uma generalização matemática é necessária para garantir independência do equipamento, bem como abordar cinemática e dinâmica. Para um nível ainda mais avançado, podiam ser explorados os módulos de expansão do *3pi*, ou outras alternativas a este equipamento.

Agradecimentos

Os autores agradecem ao Professor Doutor Miguel Barão por todo o tempo e dedicação dispensados durante as sessões e por partilhar o seu vasto conhecimento na área da Robótica. Finalmente, os autores desejam agradecer ao Professor José Duarte por todo o tempo dispensado durante as sessões de Robótica e por toda a motivação para a escrita deste artigo.

Referências

1. Algoritmo p. `http://tinyurl.com/q5vryyh`, 2014. Consultado em: 16-01-2014.
2. Atmel. Atmega328p. `http://tinyurl.com/pg44cc3`, 2014. Consultado em: 12-01-2014.
3. Robotics Bible. A simple line following robot. `http://tinyurl.com/og2ummo`, Outubro 2011. Consultado em: 13-01-2014.
4. Atmel Corporation. Atmel avr 8-bit and 32-bit microcontrollers. `http://www.atmel.com/products/microcontrollers/avr/`, 2014. Consultado em: 10-01-2014.
5. Pololu Corporation. 3. configuring the arduino environment. `http://www.pololu.com/docs/0J17/3`, 2014. Consultado em: 09-01-2014.
6. Pololu Corporation. 3pi robot + usb programmer combo - description. `http://www.pololu.com/product/1306`, 2014. Consultado em: 10-01-2014.
7. Pololu Corporation. 5. arduino libraries for the orangutan and 3pi robot. `http://www.pololu.com/docs/0J17/5`, 2014. Consultado em: 09-01-2014.
8. Pololu Corporation. 5.c. motors and gearboxes. `http://www.pololu.com/docs/0J21/5.c`, 2014. Consultado em: 10-01-2014.
9. Pololu Corporation. 7.c. advanced line following with 3pi: Pid control. `http://www.pololu.com/docs/0J21/7.c`, 2014. Consultado em: 12-01-2014.
10. Pololu Corporation. Building line following and line maze courses. `http://www.pololu.com/docs/0J22/all`, 2014. Consultado em: 16-01-2014.
11. Pololu Corporation. Digital inputs and sensors. `http://www.pololu.com/docs/0J21/5.d`, 2014. Consultado em: 12-01-2014.
12. Pololu Corporation. Pololu 3pi robot - description. `http://www.pololu.com/product/975`, 2014. Consultado em: 08-01-2014.
13. Pololu Corporation. Pololu 3pi robot - faqs. `http://www.pololu.com/product/975/faqs`, 2014. Consultado em: 12-01-2014.
14. Kevin Dowling. What is robotics? `http://tinyurl.com/pb39l7w`, 1996. Consultado em: 10-01-2014.
15. Vikram Kapila. Introduction to robotics. `http://mechatronics.poly.edu/smart/pdf/Intro2Robotics.pdf`. Consultado em: 11-01-2014.
16. microbric. Microbot robot kit. `http://www.microbric.com/p/4002074/microbot-robot-kit.html`, 2014. Consultado em: 13-01-2014.
17. BK Nagpal and A Lewis. Robotics-the scope for growth. *Production Engineer*, 63(9):26–28, 1984. Consultado em: 13-01-2014.
18. Society of Robots. Programming - pid control. `http://www.societyofrobots.com/programming_PID.shtml`, 2014. Consultado em: 15-01-2014.
19. João Pedro Pereira. Um pequeno robot para começar a explorar o oceano. `http://tinyurl.com/nud7r62`, Agosto 2013. Consultado em: 08-01-2014.
20. Oxford University Press. Robotics. http://tinyurl.com/qyg667e.
21. ZVIS Roth, B Mooring, and Bahram Ravani. An overview of robot calibration. *Robotics and Automation, IEEE Journal of*, 3(5):377–385, 1987. Consultado em: 12-01-2014.
22. Jinghua Zhong. Pid controller tuning: A short tutorial. *class lesson, Purdue University*, 2006. Consultado em: 14-01-2014.

Extração de Informação e Classificação de Textos em Língua Natural

Nuno Miranda

Universidade de Évora

Resumo O presente trabalho é um levantamento de conceitos e do estado da arte das principais abordagens e metodologias envolvidas na extração de informação e na classificação de textos em Língua Natural. Este trabalho ainda fará uma breve análise nas áreas de Representação Ontológica e de Aprendizagem Automática. Este levantamento é um estudo prévio de extrema importância para a realização de futuros trabalhos, mais complexos, na área de classificação e de extração de informação.

1 Introdução

Actualmente a nossa sociedade auto designa-se por "A sociedade da informação". No entanto tal designação é frequentemente usada num sentido lato e de pura mediatização sem grande análise em profundidade sobre a qualidade/quantidade e a real utilidade da informação que dispomos.

É certo que nos últimos 50 anos acumulamos e coleccionamos mais informação do que na restante história da humanidade. No entanto surgem dúvidas, quanto ao real proveito de tais quantidades massivas de informação. Pois pior do que não ter informação é ser inundado por informação e não saber "navegar"nela.

- De que interessa ter uma enorme biblioteca com milhares de obras se estas não se encontrarem devidamente identificadas, organizadas, catalogadas?
- De que interessa ter uma enorme Biblioteca se não soubermos ler o seu conteúdo?
- De que interessa ter uma enorme Biblioteca onde os livros se encontram dispersos numa entropia tão elevada que é impossível estabelecer qualquer relação ou ligação entre conceitos e conteúdos das informações neles contidas?

Tais questões não se aplicam apenas a uma Biblioteca desorganizada, aplicam-se também a todo o conhecimento humano. Pois mais importante do que ter apenas dados em qualquer contexto, é sobretudo, ter acesso aos conceitos provenientes do cruzamento desses mesmos dados, obtendo-se assim informação e não apenas dados.

1.1 Objectivos

Os objectivos propostos e estipulados para este trabalho é de ganhar conhecimentos em:

- Formas de representação de conhecimento recorrendo a ontologias;
- Algoritmos de classificação existentes;
- Formas de extracção de informação a partir de bases textuais.
- Analise de outros trabalhos já desenvolvidos nestas áreas.

2 Conceitos

2.1 Ontologias e Web Ontology Language

Uma ontologia é um modelo de dados que representa um conjunto de conceitos dentro de um determinado domínio bem como as relações entre esses conceitos, as suas propriedades e também pode representar as restrições dos conceitos, hierarquia, relações e propriedades nesse domínio.

Desta forma uma ontologia permite que sejam feitos levantamentos estruturados sobre os mais diversos campos do conhecimento humano de uma maneira hierárquica e organizada.

As ontologias podem ser representadas em diversas linguagens, sendo a OWL uma das mais difundidas e utilizadas. A sua sigla têm origem no inglês *Web Ontology Language* e é um *standard* do *World Wide Web Consortium* (W3C) [1], e como o seu nome indica, foi desenvolvida para ser utilizada na Web Semântica. No entanto, pode ser utilizada facilmente noutros domínios para representar qualquer ontologia sobre um determinado domínio.

O OWL possui três dialectos; o OWL Full, o OWL DL e o OWL Lite. A diferença entres eles está no grau de expressividade que permitem associar aos conceitos e relações do domínio em estudo, sendo o OWL Lite o de expressividade mais simples, seguido pelo OWL DL, e pelo OWL Full que é o mais complexo e expressivo.

Os três dialectos estão contidos uns nos outros, podendo ser vistos como extensões de expressividade do dialecto anterior. Isto significa que uma ontologia definida em OLW Lite é valida em OWL DL, e por sua vez uma ontologia definida em OWL DL também é válida em OWL Full. O inverso destas relações já não se verifica.

As ontologias são geralmente constituídas por quatro elementos básicos:
- Classe - Grupos ou colecções abstractas que tanto podem conter ou agrupar outras classes ou instâncias de classes.
- Instância de classe - São elementos concretos de uma determinada classe em que os atributos tomam valores concretos. Não são entidades abstractas mas objectos concretos e objectivos.
- Atributo - São características que descrevem propriedades das classes, e que podem tomar diferentes valores nas várias instâncias de uma determinada classe.
- Relações - Como o nome indica, são relações entre classes, instâncias de classes e atributos. As relações podem ter ou não restrições.

2.2 Aprendizagem Automática Supervisionada.

Na aprendizagem supervisionada, os algoritmos de aprendizagem automática tem acesso a uma entidade externa que quase pode ser vista como um "Professor".

Essa entidade externa vai ser responsável por fornecer ao algoritmo bons exemplos para que ele sobre esses exemplos possa criar os seus modelos de aprendizagem e então criar regras indutivas para generalizar a partir do conjunto limitado fornecido pelo "professor".

O "professor"é que vai dispor do conhecimento da classificação dos objectos, e vai saber atribuir para determinado conjunto de atributos uma determinada classe classificadora do objecto. Assim o algoritmo de aprendizagem automática irá aprender baseado nos "conhecimentos"do "professor".

Mas esta aprendizagem "Aluno - Professor"tem sempre em vista o objectivo que o algoritmo não fique restrito a saber classificar apenas casos iguais aos apresentados pelo "professor", mas que tenha alguma inteligência indutiva para saber classificar eventuais novos casos que surjam diferentes dos apresentados pelo "professor".

Nos casos práticos, o papel de "professor"é efectuado por humanos que classificam previamente conjuntos de dados seguindo um conjunto de regras obtidas através da observação e do raciocínio lógico humano, ficando assim esses algoritmos "viciados"pelos "professores".

Esta técnica é utilizada por exemplo em vários domínios de classificação automática, tais como classificação de imagens e de textos, em que são apresentados exemplos previamente classificados e rotulados para depois o algoritmo generalizar essa classificação e automatiza-la a novos casos.

2.3 Aprendizagem Automática Não-Supervisionada.

A aprendizagem não-supervisionada é uma aprendizagem que não tem qualquer tipo de entidade externa que ensine e faculte exemplos de aprendizagem ao algoritmo de aprendizagem automática. Isto pode parecer estranho à primeira vista, pois se o algoritmo de aprendizagem automática não tem qualquer conhecimento prévio ou exemplos de como agrupar os objectos correctamente como pode ele agrupar o que quer que seja correctamente?

É precisamente na resposta à pergunta anterior que reside a motivação de existir a aprendizagem automática não-supervisionada, pois a aprendizagem não-supervisionada tem um objectivo bastante diferente da aprendizagem supervisionada. A aprendizagem supervisionada parte de um conjunto de objectos pré-classificados e induz para novos casos. A aprendizagem não-supervisionada, parte de início sem nenhuma "ideia"pré-adquirida e vai então agrupar os objectos em classes ditas abstractas, a partir das propriedades dos objectos.

O objectivo é permitir que quando se tem grandes quantidades de objectos complexos e aparentemente caóticos de serem classificados por humanos, tornando-se assim impossível de existir uma entidade com o papel de "professor", pois os humanos com o papel de "professor", não têm capacidade de análise e síntese das propriedades dos objectos devido à sua elevada complexidade.

Mas com a aprendizagem não-supervisionada é possível que os algoritmos de aprendizagem automática efectuem a criação de um conjunto de classes classificativas para uma família de objectos, que de outro modo seria impossível obter devido à complexidade dos objectos em estudo.

2.4 Algoritmos de Classificação Automática

Os algoritmos de aprendizagem automática supervisionada agrupam-se em várias famílias, conforme os seus mecanismos base de funcionamento. Dentro de cada família, o princípio geral de funcionamento é semelhante, variando apenas alguns pontos ou afinações.

Árvores de Decisão Os algoritmos de aprendizagem automática baseados em árvores de decisão, são uma das família mais fáceis de perceber conceptualmente o seu funcionamento. Baseiam-se em simples árvores de decisão onde cada nó é uma condição e cada folha é um resultado final. A Figura 1 apresenta um exemplo para determinar se um dia é indicado ou não para jogar ténis.

O funcionamento da árvore é muito simples. Parte-se da raiz, que é o primeiro nó e onde se encontra a primeira condição, depois segue-se caminho conforme o nosso atributo cumpre essa condição. Cada ramo da árvore corresponde a um dos valores possíveis do atributo do nó de onde partem esses ramos. Segue-se sucessivamente para o nó seguinte até chegar às folhas da árvore. Cada folha tem a classificação final, podendo haver várias folhas com o mesmo resultado.

Desta descrição é possível concluir que uma árvore de decisão não passa de uma disjunção de conjunções lógicas sendo os ramos as conjunções e os nós as disjunções.

As TI são complexas. A Governança e Gestão de TI não têm de ser!

COBIT®Sessions

28 de Fevereiro de 2014

Esta edição das Jornadas conta com uma sessão especial "COBIT®Sessions", promovida pelo ISACA Lisbon Chapter, e dedicada às boas práticas de governança e gestão de TI, em particular as boas práticas da ISACA como o COBIT 5.

Esta sessão será dedicada ao tema "As TI são complexas. A Governança e Gestão de TI não têm de ser!" e tem o seguinte programa:

- ISACA Lisbon Chapter & COBIT 5 Overview [Bruno Horta Soares, Presidente ISACA Lisbon Chapter]
- Casos práticos da adoção de boas práticas de governança e gestão
  - EDIA - Empresa de Desenvolvimento e Infraestruturas do Alqueva, S.A.
  - Grupo Nabeiro - Delta Cafés
- Mesa redonda "As TI são complexas. A Governança e Gestão de TI não têm de ser!"com a participação de:
  - Professor Rui Quaresma, Universidade de Évora
  - Luís Estevens, Information System Manager da EDIA
  - Fernando Gonçalves, Global IT Manager no Grupo Nabeiro - Delta Cafés
  - Brundo Horta Soares, Presidente ISACA Lisbon Chapter
  - Moderador - Francisco Guimarães, ISACA Lisbon Chapter

Índice de Autores