Dissertação/Estágio

Mestrado em Engenharia Informática

Detecção de Eventos através de Padrões Comunicacionais

João Miguel de Oliveira e Silva Marques

Orientador

Paulo Gomes

Departamento de Engenharia Informática

Faculdade de Ciências e Tecnologias

Universidade de Coimbra

2013/2014

Resumo

Actualmente, muitas são as investigações em torno do novo contexto social da web. Desde organizações que pretendem saber ao certo quais os interesses dos consumidores sobre determinado assunto, à extracção de informação de uma forma automatizada como até à previsão de acontecimentos num futuro próximo.

Neste trabalho pretende-se construir um sistema que, de forma autónoma, recolha uma elevada quantidade de informação sobre eventos portugueses. Esta informação é proveniente de dois meios distintos: as publicações dos utilizadores no Twitter e as notícias online. É pretendido que o sistema consiga avaliar o conteúdo presente nas várias fontes de informação, e extrair os dados que considerar tanto verdadeiros como relevantes (datas, locais, entidades, termos e tipo de evento). Deverá ser compreender em primeira instância que publicações detém informação relativa a eventos, interpretar datas e locais presentes no texto de forma a complementar os dados, e compreender que publicações se referem ao mesmo evento e agregá-las. Será necessária realização de métodos de extracção de entidades, informação temporal, aquisição de um grande volume de dados para que o tipo de evento atribuído se torne o mais fiel possível. No final o sistema deverá compilar, sobre a forma de um único objecto, o conjunto de todos os atributos relativos a cada evento identificado. O protótipo deverá disponibilizar tanto pesquisa como recomendação semântica para facilitar a apresentação de conteúdo ao utilizador.

Keywords: Processamento de Linguagem Natural, Extracção de Informação, Reconhecimento de Entidades Mencionadas, Web Semântica, Pesquisa Semântica, Recomendação Semântica, Detecção de Eventos

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrónimos

AC Aprendizagem Computacional

API Application Programming Interface

CRUD Create Read Update Delete

CSV Comma Separated Values

EI Extracção de Informação

HTML Hyper Text Markup Language

IA Inteligência Artificial

MVC Model View Controller

PD Prospecção de Dados

PLN Processamento de Linguagem Natural

RDF Resource Description Framework

REM Reconhecimento de Entidades Mencionadas

RI Recuperação de Informação

RSS Really Simple Syndication

SREM Sistema de Reconhecimento de Entidades Mencionadas

WS Web Semântica

Capítulo 1

Introdução

Actualmente vivemos numa era onde grande parte da sociedade estabelece o seu contacto com as novidades recorrendo à Internet, através dos mais variados dispositivos. Estas novidades podem referir-se a vários tipos de acontecimentos, a ocorrer numa qualquer parte do mundo, tanto de pequena como de grande escala. Esta proliferação de conteúdo informativo surge sobre o formato de vídeos, fotografias, músicas, desenhos, aplicações ou jogos mas principalmente, textos. O registo textual é o mais utilizado para as várias finalidades da comunicação na Internet, uma vez que é também o mais simples de executar. Com os serviços disponíveis online nos dias de hoje, o utilizador pode comunicar e publicar todo e qualquer conteúdo para todo o mundo.

A quantidade de informação disponível online é gigantesca. Estima-se que nas redes sociais de topo as publicações realizadas num só dia atinjam os 500 milhões[1]. A partir daqui é possível perceber a quantidade de dados que poderá ser usada para este estudo. A aquisição de conhecimento sobre estes registos pode tornar-se vantajosa já que ela ocorre em toda a parte do planeta e de uma forma quase instantânea à ocorrência dos acontecimentos. Um dos principais problemas consiste na grande quantidade de informação crua e não estruturada, palavras mal construídas, palavras inexistentes, erros ortográficos, ou até a utilização de várias línguas numa única publicação que derivam das redes sociais. Da parte das notícias, o registo linguístico utilizado é muito mais cuidado e devidamente estruturado, mas em contrapartida, as notícias são disponibilizadas com um certo intervalo de tempo do próprio evento, são muito mais escassas, e nem sempre descrevem a verdade.

Tomando como exemplo uma manifestação na capital, sobre um qualquer tema da actua-

[1] http://www.internetlivestats.com/twitter-statistics/

lidade, onde o povo mostra o seu descontentamento perante algumas das medidas implementadas pela tutela, é grande o número de publicações, comentários e/ou notícias que emergem na Internet. O problema é que muito deste conteúdo se encontra disperso pelas várias páginas da web, o que dificulta o acesso a toda a informação existente sobre essa temática.

Congregando algumas das áreas da informática, como a Web Semântica (WS) (Berners-Lee et al., 2001) e a Inteligência Artificial (IA) (Russell and Norvig, 2009), com as potencialidades inerentes à Internet, surge a ideia de identificar eventos através da análise dados textuais presentes na web. Para retirar a maior quantidade possível de conhecimento do texto, técnicas como o Processamento de Linguagem Natural (PLN) (Jurafsky and Martin, 2008) e a Extracção de Informação (EI) (Andersen et al., 1992) tornam-se indispensáveis. Aqui será procurado relacionar muitas das tecnologias actuais de comunicação tendo em vista um objectivo muito particular: a identificação de eventos portugueses e agregação de todas as suas informações num só local de forma automatizada. Como tal, são necessárias fontes de dados que possuam uma constante actualização do seu conteúdo. É condição que estas permitam o acesso livre às suas informações.

Com o intuito de albergar um bom número de acontecimentos, as fontes de informação que serão submetidas a exploração serão de 2 diferentes tipos: redes sociais e sites de notícias. É também objectivo do projecto que o sistema seja específico para a língua portuguesa, não só pelo ainda escasso desenvolvimento de aplicativos deste ramo em português, bem como pela própria dimensão do trabalho a desenvolver.

De modo a tirar partido dos diferentes dados presentes nos dois meios e ao mesmo tempo contornar os respectivos problemas, o sistema a desenvolver deverá possuir diferentes abordagens de Extracção de Informação, não só para o tipo de fonte em estudo (rede social ou site de notícias) mas também para o tipo de evento a analisar (político, natural, desportivo,...). Para conferir bons resultados na classificação, o sistema deverá ser previamente treinado com recurso a uma grande quantidade de informação manualmente catalogada, com o objectivo de aprender o mais próximo possível do real. A procura de *keywords* específicas, análise temporal e consequente determinação da categoria do evento através de métodos de classificação, análise geográfica e comparação com eventos já conhecidos serão práticas que farão parte das abordagens a desenvolver para tornar o sistema o mais fidedigno possível.

Em complemento à realização desta dissertação, é implementado um sistema que adquire, estrutura e organiza uma série de dados referentes a eventos, num só local. Os dados po-

dem ser acedidos através de procuras simples ou de técnicas de WS. Consequentemente, é necessário criar uma ontologia (Gruber, 1993) de forma a catalogar os vários tipos de eventos da vida real. Ao recorrer a métodos de aprendizagem computacional (Carbonell et al., 1984), todo o estudo realizado deverá compreender o conhecimento de *background* necessário à implementação de sistemas de que adquiram conhecimento sobre os dados, nomeadamente, para textos com vários tipos de registos diferentes. Outra das contribuições é criação de um conjunto de dados[2] do qual fazem parte pequenos textos oriundos das fontes supracitadas, manualmente catalogados seguindo a estrutura da ontologia desenvolvida. As várias fases de experimentação bem como resultados e conclusões obtidas são fortuitas para estudos posteriores que se dediquem à Extracção de Informação a partir de textos. O interface desenvolvido, que agrega todo o resultado da análise do sistema, possibilita a pesquisa e a recomendação semântica sobre os dados dos eventos encontrados. Este interface é completamente autónomo e é actualizado em tempo real, permitindo a visualização das novidades à medida que nas redes sociais ou páginas de notícias, sejam abordados tais assuntos.

No capítulo 2 é realizado um levantamento das áreas e conceitos presente no projecto, e serão apresentadas algumas técnicas essenciais para a sua realização. No capítulo seguinte (3) são exploradas algumas ferramentas disponíveis actualmente que permitem dar suporte a sistemas de WS ou PLN, como também são demonstradas algumas das investigações já realizadas nestas áreas. No capítulo 4 são apresentados alguns dos casos de uso da utilização da ferramenta a implementar, são analisados e detalhados os requisitos do sistema, e por fim, é apresentada a arquitectura da aplicação. No capítulo 5 é explicado com detalhe cada uma das secções do sistema, a forma como estes comunicam e transformam os dados existentes. No capítulo 6 serão realizados 3 tipos de testes sobre a classificação e agregação de dados processados, e serão apresentados os seus resultados. Por fim, o capítulo 7 contém as várias fases do planeamento do trabalho e as considerações finais depois da conclusão do trabalho.

[2] O termo em inglês é *dataset*

Capítulo 2

Conceitos Fundamentais

Neste capítulo é abordado todo o grau de conhecimento das áreas de estudo envolvidas, desde os conceitos de linguística às técnicas de análise e extracção de informação. Sendo a base do trabalho o Processamento de Linguagem Natural (PLN), o capítulo começa com uma introdução sobre este tópico, passando pelas várias etapas que compõem o processo. Estão presentes exemplos que ajudam a compreender tanto toda a sequência do processamento, bem como os problemas que dele poderão advir.

2.1 Processamento de Linguagem Natural

Desde há alguns anos que vêm a ser criadas (essencialmente em filmes e outros relatos fictícios ou futuristas) entidades inteligentes como robots, que dispõem de raciocínio, operam, possuem sentidos e conseguem comunicar com os humanos através da sua linguagem natural. São relatos futuristas uma vez que a tecnologia actual ainda não permite compilar todo este conjunto de acções ou funcionalidades numa simples máquina. Pegando no exemplo da comunicação, já é possível comunicar de modo simplificado com uma máquina, recorrendo à nossa linguagem natural, seja de modo escrito ou falado. As máquinas capazes de interpretar uma qualquer frase ou conjunto de palavras fornecido por um ser humano recorrem a técnicas de PLN (Jurafsky and Martin, 2008).

A linguagem pressupõe a escrita ou a fala para ser bem sucedida pelo que são estes os campos onde se centra este tipo de processamento. Para este projecto, é necessária a tradução de contexto linguístico para o contexto conceptual, o que significa que a máquina necessitará de compreender o valor presente nos textos em análise. Tudo isto implica

a utilização de algumas técnicas que serão abordadas nesta mesma secção. Como será abordado mais adiante, esta análise lida com vários obstáculos, uma vez que a linguagem pode revelar-se ambígua de várias formas diferentes.

Sendo o inglês a língua dominante em todo o mundo, a maioria do trabalho de I&D[1], serviços, produtos e respectivo suporte, estão escritos nessa mesma língua. Com a expansão de sistemas que recorrem ao PLN, surgiu a necessidade de os moldar a línguas diferentes, dando assim cobertura a um maior número de registos (artigos, livros, documentos,...). Isto leva a uma adaptação destes sistemas, já que cada língua contém as suas próprias raízes e cultura, que consequentemente influenciam as suas regras gramaticais, dependências e mesmo o léxico. O PLN português surge neste sentido, dedicando-se única e exclusivamente à língua portuguesa.

Uma das grandes comunidades de desenvolvimento e alocação de recursos relacionados com esta temática é a Linguateca[2]. Esta dedica-se à disponibilização de informação relevante ao desenvolvimento de novas ferramentas, à partilha dos recursos existentes e avaliação dos mesmos através da comunidade. Existem mais alguns conjuntos de software livres para o PLN como o LX-Center[3] ou o Word.NET[4] para o caso da língua inglesa, mas estes serão abordados mais adiante.

2.1.1 Análise Morfológica

A análise morfológica dedica-se ao estudo da forma e estrutura de cada palavra presente no léxico. Apesar de cada elemento da frase (palavra) fazer parte de uma construção gramatical que lhe proporciona um sentido, a análise morfológica centra-se em cada uma das palavras. Deste modo, é através desta análise que são identificadas as categorias morfológicas de cada palavra , determinação do seu *lemma* que pode ou não coincidir com a sua palavra primitiva, e as suas propriedades como o número, género e modo (para o caso dos verbos). O *lemma* é geralmente a palavra base de género masculino e de número singular para o caso dos substantivos, enquanto que para os verbos o *lemma* é normalmente a forma verbal no infinitivo.

Embora nos encontremos no primeiro nível de análise do PLN, e tal como foi supracitado, um dos grandes problemas é o lidar com situações ambíguas. Na Tabela 2.2 encontram-se

[1] Investigação e Desenvolvimento
[2] www.linguateca.pt
[3] http://lxcenter.di.fc.ul.pt/tools/pt/LXChunkerPT.html
[4] http://wordnet.princeton.edu/

Substantivos	Verbo	Lemma
programa		
programas	programar	programa
programação		

Tabela 2.1: Exemplos de diferentes palavras com o mesmo *lemma*.

dois exemplos de casos que podem ocorrer a este nível.

"Ele disse que estava na **sede**", onde *sede* representa o lugar onde se concentra o poder ou a administração.
"Ele disse que tinha **sede**", onde *sede* representa a necessidade ou vontade de beber.

Tabela 2.2: Exemplo de situações ambíguas ao nível morfológico.

Inserida neste tipo de análise podem encontram-se várias tarefas distintas. De seguida serão descritas algumas delas:

- **Separação por *tokens*[5]** - é a tarefa de dividir toda a frase pelos vários elementos que a compõem. O objectivo principal centra-se na delimitação de cada palavra e de toda a frase. Problemas podem advir com a pontuação já que esta poderá tanto pôr um fim a uma só palavra como a toda a frase.

- **Divisão de frases[6]** - é o processo pela qual se identifica os vários extremos de uma frase, i.e., o início e o seu fim. Recorre essencialmente à pontuação para delimitar cada uma das frases em análise.

- **Detecção de erros** - a tarefa de identificação de erros morfológicos encontra-se nesta análise, já que é nesta fase que identificamos a estrutura e forma da palavra. A detecção de palavras incompletas, mal escritas ou com uso indevido de maiúsculas e minúsculas, é conseguida neste processo.

- **Lematização[7]** - consiste no processo de normalização de palavras, onde o objectivo é descartar toda a informação redundante como prefixos ou sufixos, e encontrar a representação lexical de um conjunto de palavras com o mesmo sentido (ver Tabela 2.1). Nem sempre o *lemma* corresponde à raiz da palavra.

[5] O termo em inglês é *tokenization*.
[6] O termo em inglês é *sentence splitting*.
[7] O termo em inglês é *lemmatization*.

- ***Stemming*** - tarefa que corresponde à redução de palavras complexas à sua base, raiz ou *stem*, que contem toda a informação semântica da palavra e que se encontra na forma mais simplificada possível. Não tem de ser necessariamente igual ao lema da palavra.

- **Reconhecimento de Entidades Mencionadas**[8] - o intuito desta tarefa é identificar entidades presentes na frase em análise, geralmente classificando-as quanto ao tipo de entidade em questão (pessoa, local, organização,...).

2.1.2 Análise Sintáctica

Sintaxe é o ramo da linguística que estuda a estrutura das frases, a disposição e relação entre as suas palavras. Daqui advêm orações simples ou compostas. Esta análise tem grande relevância na interpretação do sentido correcto da frase já que, apesar das palavras possuírem o seu próprio significado, este último depende tanto da classe da palavra na respectiva frase como também das outras palavras em seu redor. É necessário o conhecimento de todas as classes gramaticais e suas regras para que o processo seja bem sucedido.

Para a identificação de cada uma das classes de palavras, é realizado um processo de marcação ou etiquetagem[9] de cada um dos elementos presentes na frase. É elaborada uma estrutura hierárquica, composta por categorias e subcategorias que abrangem cada uma das palavras, dependendo da sua posição, função bem como das palavras vizinhas. A árvore sintáctica é resultante da aplicação das regras gramaticais da língua em questão (ver Figura 2.1).

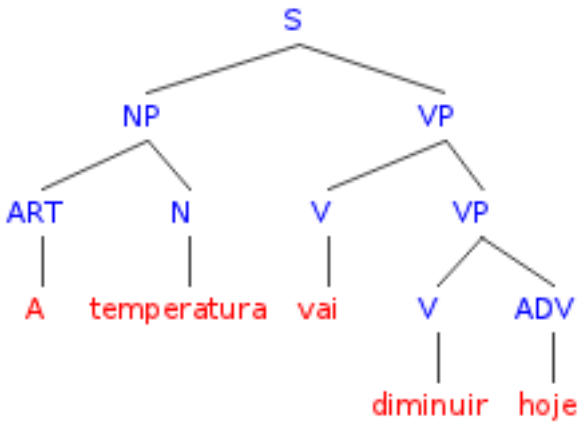


Figura 2.1: Exemplo de árvore de *parsing*.

[8] O termo em inglês é *Named Entity Recognition*.
[9] O termo em inglês é *POS Tagging*.

S → NP VP
NP → ART N
VP → V VP
ART → A
N → temperatura
V → vai
VP → V ADV
V → diminuir
ADV → hoje

Tabela 2.3: Regras de *parsing* aplicadas ao exemplo da Figura 2.1

Estas técnicas baseiam-se em gramáticas livres de contexto que permitem identificar relações, funções e dependências de todos os constituintes da frase. Existem também gramáticas de dependências onde as diferenças residem nas relações de um para um nas palavras ou orações. Apesar de todo este processo de categorização e tal como na análise morfológica, também nesta fase podem surgir erros ou situações ambíguas. Na frase: "A **rapariga** no **carro** que precisava de **água**, está à espera", é possível notar a ambiguidade que pode ocorrer neste nível. Para este caso, não é possível determinar se é a rapariga ou o carro que precisa de água, gerando dois cenários possíveis de interpretação. Até a própria leitura humana poderá não ser bem sucedida nesta tarefa sem a presença de mais elementos na frase.

2.1.3 Análise Semântica

Após a execução da análise morfológica e verificação das regras gramaticais através da análise sintáctica, é necessário retirar o verdadeiro nexo da frase, o seu significado, a informação nela contida. Esta função diz respeito à análise semântica, o último dos processos de interpretação e transformação da informação do contexto linguístico para o contexto conceptual. Como as máquinas não possuem capacidade para extrair o conteúdo informativo das frases, são necessárias algumas técnicas que ajudam à aquisição e interpretação de conceitos e respectivas relações presentes em cada frase. Para isto é necessário representar o conhecimento através de formalismos, como a lógica de predicados (Smullyan, 1995) (ver Figura 2.2), grafos direccionais (ver Figura 2.3) ou *frames* semânticas (Fillmore, 1982) (ver Tabela 2.4).

Uma vez que a frase constitui um conjunto completo de várias partes, faz sentido dizer

do-tipo(lançamento-do-album, evento)
tem(evento,data)

Figura 2.2: Exemplo da representação de conhecimento pela lógica de predicados que consiste na representação do conhecimento através de funções e os seus argumentos.

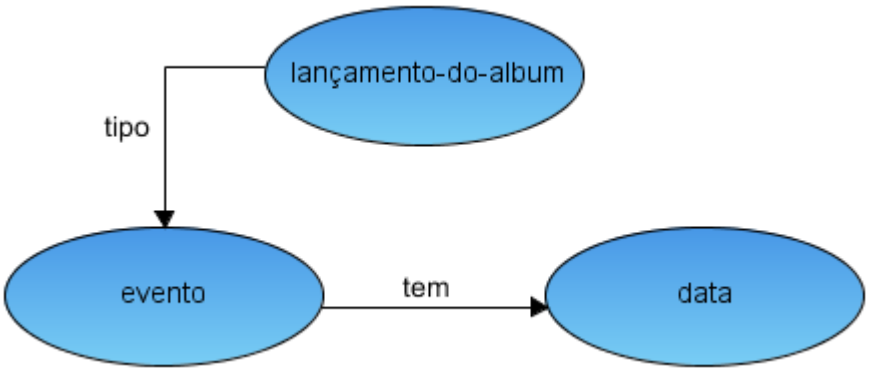


Figura 2.3: Exemplo de um grafo direccional.

que o verdadeiro sentido da frase é retirado a partir do sentido de todas essas mesmas partes. Posta a análise sintáctica, este tipo de processamento poderá sentir necessidade de recorrer a gramáticas semânticas, onde estão presentes muitas das regras e restrições da língua, ajudando à melhor interpretação das relações estabelecidas entre as palavras, pese embora muitas delas estejam limitadas a um domínio ou área específica.

Com a existência de palavras homógrafas e homónimas, é possível a existência de ambiguidades também a este nível já que, uma palavra poderá conter vários significados distintos para o mesmo tipo de função sintáctica na frase. Um exemplo é a palavra **colher**: tanto pode representar o utensílio utilizado para comer (substantivo), como pode significar o acto de apanhar ou recolher (verbo). Para estes casos, é necessário o tratamento através de um processo de Desambiguação[10], que consiste exactamente na identificação do sentido próprio da frase, quando esta possui mais do que uma possibilidade de interpretação.

2.1.4 Recuperação de Informação

Existem vários propósitos para utilização do PLN. A Recuperação de Informação (RI) é uma das tarefas que pode surgir neste sentido e diz respeito à pesquisa de material não estruturado que satisfaça a necessidade de informação a partir de grandes armazenamentos de informação (Manning et al., 2008). Este material é geralmente texto, a pesquisa é realizada a partir de *queries* o que pressupõe a existência de computadores para servir de

[10] O termo em inglês é *Word Sense Desambiguation* (WSD).

lançamento-do-álbum:
do-tipo: evento
tem: data

Tabela 2.4: Exemplo da representação de conhecimento pela *frame* "lançamento do álbum"

suporte.

Antigamente este tipo de tarefa era desempenhado por pessoas que se dedicavam única e exclusivamente à pesquisa de informação em grandes colecções como bibliotecas ou artigos. Com a massificação de informação, a aparição dos sistemas de armazenamento digital e a Internet, foi necessário adaptar a RI. Pesquisa de palavras-chave num e-mail ou num artigo científico é o tipo de novas necessidades que readaptaram a RI.

É considerado um documento a um conjunto de palavras independentes e estes são considerados termos. Uma vez que podem existir vários documentos, e que o mesmo termo pode estar presente em mais do que um documento, é costume agregar-se o par documento-termo e atribuir-lhe um peso, consoante a sua relevância no documento em questão. Existem vários modelos ou estratégias de RI, e dependem do material em análise, do tipo de procura e do tipo de dados a obter.

Dependendo da *query* inserida pelo utilizador, o objectivo é encontrar a grau de importância de cada documento e listá-los por preferência. De modo a avaliar o desempenho dos modelos aplicados, foram criadas algumas técnicas[11], e estas pressupõem o conhecimento tanto dos documentos e dos termos neles existentes.

- ***precision*** - é a percentagem de documentos devolvidos que satisfazem a necessidade do utilizador. A sua expressão é $precision = \frac{relevantes \cap devolvidos}{devolvidos}$;
- ***recall*** - é a percentagem de documentos devolvidos que satisfazem a *query* introduzida. A sua expressão é $recall = \frac{relevantes \cap devolvidos}{relevantes}$;
- ***F1-score*** - é um teste de exactidão e consiste na média harmónica da *precision* e da *recall*. É também conhecido por *F-score* ou *F-measure*. A sua expressão é $f\text{-}score = 2 \times \frac{precision \times recall}{precision + recall}$;

[11] "relevantes" diz respeito aos documentos de interesse para o utilizador; "devolvidos" são os documentos devolvidos pelos sistemas de RI.

2.1.5 Extracção de Informação

De forma a interpretar os dados presentes no corpo linguístico, a Extracção de Informação (EI) (Jurafsky and Martin, 2008) consiste no processo automatizado de obtenção de informação estruturada a partir de fontes de informação não estruturadas, da qual fazem parte várias técnicas. Geralmente, tratam-se de documentos de linguagem natural, mas a EI também pode ser utilizada em documentos multimédia como imagens, vídeo ou áudio. Um objectivo mais complexo é a inferência de novo conhecimento a partir de premissas conhecidas à partida[12], podendo ser descobertas novas relações entre os dados.

Na secção 2.3 serão abordadas algumas das técnicas de EI mais comuns. Algumas das áreas que se inserem na EI são o Reconhecimento de Entidades Mencionadas (REM), a extracção de emoções, extracção de informação temporal ou a extracção de relações (Ling and Weld, 2010).

Tal como acontece na vida real, a não percepção da informação seja através do discurso ou da escrita, leva a erros de interpretação e, consequentemente, a comunicações mal sucedidas. De modo a evitar tal acontecimento, é necessário que a informação se encontre estruturada através das regras gramaticais. Hoje em dia, e olhando para a Internet, muitas são as fontes de informação disponíveis, o que leva a que não seja possível garantir que exista tanto a estruturação de informação, bem como a veracidade da mesma. Estas e outras limitações serão certamente encontradas na implementação deste protótipo.

Informação Semântica

Segundo o princípio da composicionalidade (Szabo, 2004), o significado de qualquer expressão linguística complexa é resultado do conjunto de todas as expressões linguísticas que a compõem. Isto significa que, embora cada palavra ou expressão tenha uma função específica na frase (complementos, verbos, sujeitos,...), no conjunto é produzido um único sentido à frase. É a partir deste ponto que se inicia o processo de extracção de informação, uma vez que, a função ou posição relativa de cada palavra na frase, influencia o seu sentido. Assim cada palavra não pode ser vista como apenas uma palavra, mas sim com uma peça essencial deste puzzle linguístico.

[12] O termo em inglês para esta prática é *reasoning*.

Informação Específica

Geralmente, o teor do texto em análise para a EI é conhecido, o que significa que é possível especializar o sistema. Imaginando um artigo científico sobre determinada espécie de répteis, é muito provável que apareçam referências a locais que representam os *habitats* ou locais de descoberta de registos, nomes de espécies, referências temporais sobre a evolução das mesmas, entre outros elementos. Através deste conhecimento *a priori*, é possível desenhar o sistema para este tipo de ocorrências em particular. Como será abordado na próxima secção, o REM (Poibeau and Kosseim, 2001), uma das sub-áreas de EI, consiste na identificação de entidades presentes no texto, o que possibilita, para o exemplo fornecido, a identificação de espécies e locais com uma maior taxa de sucesso. Dependendo do objectivo, poderá ser ou não necessário dar ênfase a certas classes de palavras como as referências espaciais ou temporais, o que ajuda no processamento da EI. Assim, é possível dizer que a especificidade do sistema depende não só do domínio onde está inserido, mas também do conjunto de elementos que desse domínio fazem parte.

Conhecimento Léxico-Semântico

Quando existe a necessidade de uma maior abrangência do domínio de informação a retirar de um texto, como a extracção de eventos ou a resposta automática a perguntas, surge o conhecimento léxico-semântico. Este tira partido das relações semânticas existentes entre palavras, para de uma melhor forma interpretar o seu significado. As vantagens são um maior domínio do conhecimento, o que resulta numa redução de situações de ambiguidade já que existe mais informação sobre cada palavra. Em contrapartida, e ao contrário do que acontecia na EI específica, esta grande quantidade de informação sobre toda a língua e respectivas relações estabelecidas entre as suas palavras, exige um exaustivo e demorado trabalho prévio. Este pode ser executado tanto manualmente (como é o caso da Word.NET) como automaticamente (Oliveira and Gomes, 2010). Outra das desvantagens é abordagem sobre domínios específicos de conhecimento ou a tradução para uma outra qualquer língua que não seja o inglês, já que as relações entre palavras são diferentes.

Classificação

Como em qualquer sistema de processamento de dados, a classificação da informação revela-se uma fase importante. Esta classificação actua sobre as várias expressões presentes no corpo de texto em análise, e não sobre o todo, uma vez que o texto pode abranger

classes diferentes. Estas classes poderão estar sujeitas a uma taxonomia o que determina a hierarquia estabelecida entre todas elas. Comparando as abordagens supracitadas, a EI pode revelar-se específica ou genérica: para o primeiro caso, o modelo de extracção pode e deve conter regras próprias da classe em questão, sendo bastante restritivo e fechado, enquanto que para o segundo caso, as regras deverão ser abrangentes e heterogéneas, visto não ser possível antever a classe a que cada expressão pertence.

Técnicas de Extracção de Informação

Ao nível de técnicas utilizadas para EI, existem algumas possibilidades como o Reconhecimento de Padrões que, tal como o próprio nome indicia, procura encontrar padrões nos objectos em estudo de modo a categorizá-los; Aprendizagem Supervisionada, que consiste no processo de treino do sistema através de um *input* com classificação, o que confere aprendizagem auxiliada ao sistema; Aprendizagem Não Supervisionada, que consiste na aprendizagem sobre dados não catalogados, apenas por interpretação e análise dos diferentes resultados. Todas estas técnicas serão abordadas em maior detalhe na secção 2.3.

2.1.6 Reconhecimento de Entidades Mencionadas

O Reconhecimento de Entidades Mencionadas[13] é uma das sub-áreas da EI. É responsável pela identificação e classificação de nomes próprios (que representam as entidades) como pessoas, locais, organizações, eventos, podendo mesmo tratar-se de abstracções. Podem tanto recorrer a regras gramaticais da língua bem como a modelos estatísticos para a detecção destes elementos no léxico. Estes modelos estatísticos exigem dados previamente classificados que serão fornecidos ao sistema para servirem de base para a comparação.

A título de exemplo podemos analisar a frase "O **João Marques** é estudante da **Universidade de Coimbra** desde **2008**", onde é possível encontrar 3 entidades de 3 tipos distintos:

- "João Marques" - Pessoa;
- "Universidade de Coimbra" - Organização;
- "2008" - Valor;

[13] O termo em inglês é *Named Entity Recognition*.

É comum encontrar sistemas de EI dedicados a um tema específico de modo a aumentar a sua precisão. Desta forma, este pode ser afinado para o respectivo domínio através do tipo de discurso, de palavras próprias ou expressões, facilitando a identificação de entidades (Ritter et al., 2012). Em contrapartida, este tipo de abordagem restringe o domínio do sistema, já que, por estar especializado num tema em concreto, não tem capacidade para de ir para além do espaço de conhecimento onde se encontra.

2.2 Ontologias

As ontologias são representações formais e explícitas dos termos e suas relações num determinado domínio de conhecimento (Gruber, 1993). O objectivo das ontologias é a transformação do conhecimento presente no mundo linguístico para o meio conceptual, permitindo a partilha de informação num certo domínio (Noy and Mcguinness, 2001). Alguns dos benefícios da utilização das ontologias são:

- partilha de informação devidamente estruturada entre várias partes, sejam elas pessoas ou máquinas.
- a possibilidade de reutilização de conhecimento já desenvolvido sobre um domínio.
- analisar o conhecimento do domínio.

As ontologias são representadas por uma estrutura hierárquica composta por classes (ou conceitos) e as respectivas propriedades (ou papeis). As classes representam os conceitos presentes na própria ontologia: a título de exemplo, podemos idealizar uma ontologia de meios de transporte, onde algumas das classes poderiam ser "automóvel", "barco", "avião", entre outros. As propriedades são atributos da classe, e poderão estar sujeitas a restrições da própria classe ou mesmo da propriedade. Pegando no mesmo exemplo da ontologia dos "meios de transporte", exemplos de propriedades poderiam ser "número de lugares", "tipo de combustível", "peso", entre outras. Classes descendentes herdam todas as propriedades das classes acima, já que pertencem a essa mesma categoria, mas são de um tipo mais específico e necessitam de albergar mais conhecimento. Para a ontologia de meios de transporte, era possível criar uma classe hierarquicamente superior à classe "carro" que fosse "meio terrestre", que poderia ter propriedades específicas como "número de portas", já que nem a "barcos" ("meio aquático") nem a aviões ("meio aéreo") esta propriedade interessa. Sendo uma estrutura em árvore, poderão ser desenvolvidas através de uma abordagem *top-down*

(da raiz para as folhas), *bottom-up* (das folhas para a raiz) ou híbrida (em ambos os sentidos até se cruzarem).

Distinguem-se essencialmente 2 tipos de ontologias que dependem do seu alcance e do seu objectivo:

- **ontologias de domínio** - este tipo de ontologia é aplicado em áreas específicas, onde o domínio se encontra bem delimitado e restrito a esse mesmo contexto;
- **ontologias gerais** - como o próprio nome indica, este tipo de ontologia é aplicado a uma grande variedade de domínios, o que lhe confere generalidade e uma abstracção mais vincada, de forma a estar em conformidade com todas as áreas abrangidas;

O termo não é propriamente recente, várias são as ontologias construídas até aos dias de hoje até que estas começaram a ser aplicadas na estruturação de dados na própria web. Com isto, surgiram várias bases de dados específicas para alocar e disponibilizar esta formalização de conhecimento.

Para a construção de uma ontologia, e sendo esta responsável por toda a estruturação e formalização do conhecimento, existem vários princípios a ter em conta:

1. **definição do seu domínio e objectivo** - em primeira instância, deverá ser bem definido o alvo da ontologia, até onde esta deverá chegar, as suas fronteiras e qual o seu propósito. Alterações e actualizações futuras deverão ser tidas em conta de forma a construir correctamente a ontologia;
2. **reutilização ontologias já construídas** - com a existência de bases de dados de ontologia, e com a implementação destes formalismos na web, provavelmente já existem ontologias que cobrem o domínio em tratamento de forma parcial ou até total. É recomendado então que se realize um a procura prévia no sentido de saber se existem já opções para o nosso problema. Este princípio deve ser adoptado sempre que a aplicação irá comunicar com outros sistemas do género. Existindo troca de dados, é sempre mais fácil executar estas tarefas se ambos se regerem pelas mesmas regras;
3. **identificação de termos importantes** - como ponto de partida, deverão ser apontados termos, conceitos ou considerações importantes a ter em conta tanto no processo de concepção como para a própria estrutura da ontologia. Não existe grande necessidade de organização nesta fase, existe sim a necessidade de indicar pontos que necessitam de estar presentes;

Triplo	Sujeito	Predicado	Valor
1	pessoa1	temNome	Miguel
2	pessoa1	temIdade	23
3	pessoa2	temNome	Sofia
4	pessoa2	temNamorado	pessoa1

Tabela 2.5: Exemplos de triplos e das relações que estes podem conter.

4. **definição das classes** - a partir deste ponto, é iniciado a construção da estrutura hierárquica que constitui a ontologia. São identificadas as classes e as suas relações. A taxonomia deverá ser equilibrada de modo a que não existam classes nem com excesso (> 12) nem com escassez de filhos (< 2).

5. **definição das propriedades** - construída a árvore que formaliza o conhecimento através de classes, é realizada a enumeração de propriedades para cada uma das classes. É necessário ter em conta que classes são afectas a determinada propriedade, o que pode levar a que estas fiquem dispostas em diferentes localizações da taxonomia.

De forma a dar suporte a estas estruturas, existem *frameworks* cuja função é o armazenamento, suporte e interacção de dados para estes formalismos. Distinguem-se das bases de dados relacionais uma vez que os dados são guardados através de **triplos** em formato Resource Description Framework (RDF). Os triplos são um formato de dados que contém 3 componentes principais:

- **sujeito** - consiste no nome do recurso, e é meramente identificativo;
- **predicado** - expressa a relação entre o sujeito e o objecto, pode conter o nome de propriedades ou outros aspectos que tornam relevante o valor do objecto;
- **objecto** - contém o valor do triplo ou a resposta ao predicado;

2.3 Aprendizagem Computacional

Com a gigante difusão de informação disponível na Internet, tanto pela diversidade de páginas web bem como pelos vários tipos de formatos que os dados podem adoptar, tornou-se indispensável criar processos autónomos que fossem capazes de extrair essa mesma informação. Não é possível do ponto de vista biológico adquirir e memorizar tamanha

quantidade de dados, de modo que se tentou incutir a aprendizagem em máquinas para posteriormente realizarem determinadas tarefas por nós.

A Inteligência Artificial (IA) é o ramo da ciência computacional que se dedica à criação de agentes inteligentes com capacidade de interacção com o meio ambiente onde estão inseridos e com os outros agentes em seu redor. A definição de IA tem vindo a progredir à medida que a investigação avança, e acarreta sempre algum caracter subjectivo: é válido dizer que a IA se centra na reprodução do comportamento humano em máquinas, como na introdução de raciocínio próprio nestes sistemas (Russell and Norvig, 2009). Algumas das áreas inseridas neste ramo são Visão Computacional, Planeamento Automatizado, PLN e a Aprendizagem Computacional (AC).

A AC(Bishop, 2006) é uma área que se dedica ao estudo e concepção de sistemas que aprendem a partir de dados. São desenvolvidos algoritmos que sofrem alteração do comportamento à medida que os dados lhes são injectados (treino) e, consequentemente, com o reconhecimento de padrões nos dados, vão afinando a sua precisão do resultado a devolver aos utilizadores. Para concretizar, estes são alguns exemplos de problemas que podem ser resolvidos com recurso à AC:

- pesquisa de informação na web através dos avançados algoritmos presentes nos motores de busca;
- recomendação de produtos numa loja online através da detecção de padrões de interesses do utilizador;
- detecção e reconhecimento facial através de câmaras;
- detecção e reconhecimento de caracteres;
- filtragem de emails (*spam* e não *spam*);
- classificação de tipos de notícias como "desporto", "política", "cultura", entre outros;
- estacionamento automático de carros após observação do comportamento humano;
- previsão meteorológica;
- previsão das variações do mercado bolsista;
- diagnóstico médico de pacientes sobre determinada doença;
- detecção de fraudes em transacções bancárias;

Actualmente, está presente em muitas das áreas da ciência e tecnologia, e é provável que utilizemos produtos que recorrem a AC sem que tomemos consciência disso mesmo.

Existem vários tipos de abordagens em AC que dependem dos problemas com que estes lidam:

- Supervisionada[14] - consiste no tipo de aprendizagem por acompanhamento. É concedido ao programa um valor de entrada e o resultado esperado, de forma a que este consiga desenvolver uma função que permita obter novos resultados perante cenários desconhecidos. Exemplos de algoritmos com aprendizagem supervisionada: Retropropagação, *k-NN* ou classificador Bayesiano;

- Não Supervisionada[15] - consiste no tipo de aprendizagem sem recompensa ou conhecimento prévio sobre os dados. Estes chegam de forma crua ao sistema, sem qualquer tipo de classificação. A aprendizagem centra-se na procura e reconhecimento de padrões, para que seja possível classificá-los. Exemplos: *clustering* e cadeias de Markov;

- Por Reforço[16] - tal como o próprio nome indica, consiste no tipo de aprendizagem por reforço, onde apenas são conhecidos os valores de entrada e, ao contrário da aprendizagem supervisionada, não são fornecidos os resultados esperados. A aprendizagem consiste na tomada de decisões do sistema que depois são avaliadas positiva ou negativamente, concedendo uma recompensa ou uma punição respectivamente. Os algoritmos genéticos são um exemplo prático deste tipo de aprendizagem uma vez que recebem o reforço através do ambiente.

- Semi-supervisionada[17] - consiste na aprendizagem com recurso a uma mistura de dados catalogados e não catalogados. A tarefa de catalogar todos os dados de treino pode ser extenuante, pelo que, através desta aprendizagem, o sistema é capaz de desenvolver um modelo de classificação depois de analisados dados catalogados e dados não catalogados. Podemos encontrar algoritmos de auto-treino e máquinas de vector de suporte neste tipo de abordagem;

[14] O termo em inglês é *Supervised*
[15] O termo em inglês é *Unsupervised*
[16] O termo em inglês é *Reinforcement*
[17] O termo em inglês é *Semi-supervised*

2.4 Web Social

Web Social (Rheingold, 2000) é um termo que só nasceu depois do novo milénio. Diz respeito à componente social da web 2.0, que visa criar, melhorar e recomendar as interacções sociais entre utilizadores de Internet. Refere-se à nova geração de páginas, serviços e funcionalidades disponíveis na rede, que em muito se distinguem das que existiam nos primeiros tempos da Internet. Enquanto que na web 1.0 o conteúdo disponível era estático, as fontes de informação eram restritas, as funcionalidades eram limitadas e a integração de múltiplos sistemas era uma tarefa difícil. Nesta nova era o conteúdo é dinâmico, as atenções centram-se no utilizador já que este é o principal interveniente na rede, existe uma enorme quantidade de serviços disponível que facilmente interagem uns com os outros graças às Application Programming Interface (API)'s (Currier, 2008).

Uma vez que hoje em dia o suporte para a grande maioria das interacções sociais é essa grande rede virtual, este é um conceito em pleno desenvolvimento. A principal ideia é facilitar a utilização, aumentar a satisfação dos utilizadores, permitir e estreitar relações sociais. Dentro do ramo da Web Social, estão compreendidas áreas como as redes sociais, jogos ou compras online, Micro-blogging[18], pesquisa ou ofertas de emprego, realização de eventos, discussões sobre tópicos da actualidade ou a partilha selectiva de conteúdo.

Dado este cenário, é gigante o conhecimento que se pode adquirir estudando e conjugando todos estes dados. Mais uma vez é possível obter os padrões e interesses de utilizadores relativamente a modos de vida, a *hobbies*, locais, músicas ou filmes preferidos, rotinas ou trajectos do dia-a-dia, realizar pesquisas de informação através da Web Semântica[19], em tempo real e de forma muito mais usável. Ao nível empresarial, este é o tipo de informação que se pretende adquirir e manter actualizada, de forma a operar no sentido preferencial dos consumidores, perceber as tendências de mercado e conhecer o estado dos concorrentes.

2.4.1 Twitter

O Twitter[20] é uma rede social que permite a publicação de *tweets* (mensagens/comentários de texto) com um limite de 140 caracteres, que assenta no objectivo inicial da companhia de criar a "SMS para a Internet". Criado em 2006, já atingiu os 600 milhões de utilizadores[21]

[18] http://www.webopedia.com/TERM/M/microblog.html
[19] http://www.w3.org/2001/sw/
[20] https://twitter.com/
[21] http://www.statisticbrain.com/twitter-statistics/

e, tal como o Facebook[22], é um dos meios de comunicação mais bem sucedidos em todo o Mundo, o que justifica a presença no top 10 de sites[23] mais acedidos nos dias de hoje.

A nível de funcionalidades que se revelam importantes para este projecto, existem os próprios *tweets* onde os utilizadores relatam, informam ou comentam um qualquer tipo de informação; a adição de localização aos *tweets* que informa qual o local do utilizador, os *followers* (ou seguidores) que se tratam de utilizadores que "assinam" todo o tipo de conteúdo de terceiros, recebendo-o em tempo real na sua própria página, as *hashtags* que referenciam uma outra página no Twitter, e a partilha de fotos.

Sendo esta uma das redes sociais de topo, também ela possui uma API que permite o acesso a todo o tipo de dados da parte dos *developers*, que é utilizada em muitas das aplicações e sites, para efeitos de autenticação e acesso a dados, e que também se encontra em constante desenvolvimento. O Twitter possui 2 API's diferentes para distribuição de dados para as aplicações:

- **API REST** - o típico processo de *request-response* entre cliente e servidor, onde os parâmetros de entrada do cliente servirão de objecto de pesquisa sobre os dados do Twitter. Assim que o resultado é encontrado e transformado no formato de recepção pretendido pelo cliente, o servidor envia a resposta. Os resultados poderão pertencer a *tweets* dessa mesma altura como do passado.
- **API de *Streaming*** - tipo de ligação que necessita de um processo destacado só para a recepção de *tweets*. Isto acontece porque, ao contrário da API REST, aqui não existe um a ligação *request-response*, mas sim um fluxo de informação unidireccional e assíncrono do servidor para o cliente[24]. Não é possível antecipar quando surgirá um novo *tweet*, daí que seja necessária um tipo de ligação persistente.

Apesar de tudo, existem algumas semelhanças entre estas 2 API's: ambas permitem realizar *queries* sobre a base de dados, embora a API REST permita *queries mais complexas* já que não se cinge aos *tweets* em tempo real; ambas permitem obter uma boa amostra de *tweets* embora se estime que a API de *Streaming* possa devolver até cerca de 180.000 *tweets* por hora[25], enquanto que a API REST pode responder com até 72.000, para o mesmo período temporal.

[22] https://www.facebook.com/
[23] http://www.alexa.com/topsites
[24] https://dev.twitter.com/docs/streaming-apis
[25] http://140dev.com/twitter-api-programming-tutorials/aggregating-tweets-search-api-vs-streaming-api/

2.4.2 Problemas Típicos das Redes Sociais

Apesar do potencial de conhecimento encerrado nas redes sociais, a verdade é que o seu registo linguístico tende a oferecer alguns problemas a nível de PLN e EI. Enquanto que em cenários noticiosos como jornais, artigos ou livros, onde é exigido que a escrita siga todas as regras gramaticais, o que acontece nos *streams* sociais é que cada utilizador é livre para escrever sobre um qualquer tema de uma qualquer forma, deixando ruído aos sistemas que os analisam.

1	"JÁ ESTÁ! Já lá estamos! Jamor 2012"
2	"Estou de férias e não é que estou muuuito melhor assim!?"
3	"Malta! Acabei de lançar a minha nova app para Android!! Q acham?"
4	"Acabei de ver o Despicable ME. Demais! It's so fluffy im gonna die!!!"

Tabela 2.6: Alguns exemplos da linguagem praticada nas redes sociais.

Problemas como a má utilização de maiúsculas/minúsculas, as conhecidas abreviaturas, má formação de palavras ou uso de várias linguagens num só corpo de texto, estão largamente presentes não só nos *tweets* mas em todo o tipo de comentários realizado pelos utilizadores. Não possuindo qualquer tipo de obrigação nas regras de escrita (o que acaba por um ponto atractivo destes meios), o registo adoptado é o coloquial. Na Tabela 2.6 estão descritos alguns exemplos de obstáculos à tradução do conhecimento nas redes sociais: má utilização de maiúsculas/minúsculas de palavras, palavras mal escritas, abreviaturas e o uso de várias línguas diferentes. Como tal, são necessárias abordagens que a ajudem tanto a identificar todas estas notações bem como para contornar a falta de informação presente nestes corpos de texto.

2.4.3 RSS Feeds

A televisão, nos últimos anos, tem sofrido o efeito de digitalização: os espectadores não se encontram mais limitados à restrita gama de programas e canais como acontecia no século XX. É possível escolher o conteúdo a transmitir no televisor, como se de uma ementa se tratasse. O mesmo fenómeno tem acontecido com as notícias: os interessados não necessitam de comprar um ou mais jornais para se manterem a par das últimas do país e do mundo. Desde aplicações para *smartphones*, páginas web a serviços de agregação

de conteúdo, existem várias formas de ler as notícias e ao mesmo tempo seleccionar que notícias ler.

Um destes serviços é o Really Simple Syndication (RSS) desde a versão 2.0. Tem como objectivo a subscrição de *feeds*, que não são nada mais do que tópicos de informação. Sites como serviços de RSS geralmente publicam conteúdo com regularidade, e exploram este serviço de modo a partilhar todo esse conteúdo com os subscritores. É adoptado o formato XML (*eXtensible Markup Language*)[26] ou variantes deste, visto ser universal. Os dados que geralmente são transmitidos são notícias e seus constituintes como título, categoria, data, descrição e ligações externas. Trata-se de uma opção bastante eficaz e ligeira de transmissão de informação pela web.

Para o protótipo desenvolvido, este serviço disponibiliza as notícias oriundas de jornais portugueses. Através deste, será possível obter o título, uma breve descrição, a data e um *link* para a notícia completa. Ao contrário do texto proveniente das redes sociais, este deverá estar bem estruturado e sem erros ortográficos (salvo raras excepções). Deste modo, ajudará a equilibrar a balança na procura por notícias referentes a eventos do presente.

[26] http://en.wikipedia.org/wiki/Xml

Capítulo 3

Estado da Arte

Neste capítulo serão demonstrado os actuais avanços tecnológicos nas áreas de incidência do projecto, com apresentação de sistemas de suporte existentes no mercado; e serão também exploradas algumas das investigações já realizadas na extracção de informação tanto para as redes sociais como para as notícias.

3.1 Ferramentas e Recursos

Já existem vários tipos de sistema que prestam suporte ou recorrem às áreas já apresentadas, uma vez que estas não são propriamente recentes. Existem no entanto plataformas comerciais e não comerciais. Para este projecto apenas serão abordadas as opções não comerciais. São várias as opções existentes, mas apenas será explorado um reduzido número delas para conhecer o actual estado da arte, uma vez que apenas uma destas será incorporada no projecto.

- Processamento de Linguagem Natural:
  - Mallet; [1]
  - OpenNLP; [2]
  - Natural Language Toolkit (NLTK); [3]
  - Stanford NLP; [4]

[1] http://mallet.cs.umass.edu/
[2] http://opennlp.apache.org/
[3] http://nltk.org/
[4] http://nlp.stanford.edu/software/index.shtml

Ferramentas	Mallet	OpenNLP	NLTK	Stanford NLP
Linguagem	Java	Java	Python	Java
Etiquetagem	Não	Sim	Sim	Sim
Separação por *tokens*	Não	Sim	Sim	Sim
Topic modelling	Sim	Não	Não	Não
Classificadores	Naive Bayes, MaxEnt, Decision Tree	OpenNLP Categorizer	MaxEnt, Naive Bayes, Decision Tree	Stanford Classifier

Tabela 3.1: Tabela de comparação de alguns dos sistemas de PLN existentes.

- Prospecção de Dados:
  - Weka; [5]
  - RapidMiner; [6]
  - R; [7]
  - Mahout; [8]
  - Orange; [9]

Mallet

Mallet é uma biblioteca construída em Java que agrega funções de Aprendizagem Computacional (AC) como o PLN, *clustering*, extracção de informação, modelação, classificação entre outros. Possui uma grande variedade de algoritmos usados para os cenários já descritos: *Naïve Bayes*, *Hidden Markov Models*, *Maximum Entropy*, *Latent Dirichlet Allocation*, *Decision Trees*. Foi desenvolvido em Java num ambiente académico com o contributo de estudantes das Universidades de Massachusetts e de Pennsylvania. Consiste numa opção para utilizadores mais experientes nestas áreas, porque apesar bastante capaz, não possui muita documentação nem muitos módulos. A sua licença de utilização é a CPL (*Common Public License*) que permite a utilização total ou parcial do código, bem como a sua distribuição, e ainda a integração com outros programas que se não rejam pela mesma licença.

[5] http://www.cs.waikato.ac.nz/ ml/weka/
[6] http://sourceforge.net/projects/rapidminer/
[7] http://www.r-project.org/
[8] https://mahout.apache.org/
[9] http://orange.biolab.si/

OpenNLP

O Apache OpenNLP é uma biblioteca de ferramentas para PLN. Contém uma série de funcionalidades desenvolvidas na linguagem Java, que possibilitam várias operações ao nível do PLN como: separação por *tokens*, etiquetagem, *chunking*, *parsing* e extracção de entidades. A sua licença de utilização é a *Apache License 2.0*, que permite a utilização livre do software, total ou parcial, tanto para fins pessoais, institucionais ou até comerciais. Não permite a distribuição de qualquer parte do software sem as devidas referências. Embora seja limitado ao PLN, encontra-se bem documentado.

NLTK

NLTK é um sistema gratuito desenvolvido para analisar o texto através de técnicas de PLN. Começa por ser diferente das outras opções apresentadas já que não foi construído em Java mas sim em Python. Isto facilita a interacção com a plataforma mas torna a sua *performance* mais lenta comparativamente a algumas outras soluções. Dispõe de módulos de separação por *tokens*, etiquetagem, Reconhecimento de Entidades Mencionadas (REM) e vários classificadores como Naive Bayes ou Maximum Entropy.

Stanford NLP

Este é um grupo de investigadores, programadores e estudantes que trabalham activamente no desenvolvimento de software para aquisição de informação através de várias línguas existentes. Actualmente este grupo já construiu várias tecnologias de PLN para algumas das mais faladas línguas do mundo: *POS Tagger*, *parsing*, reconhecedor de entidades, classificadores, *english tokenizer*, entre outros. Rege-se segundo a licença e política de utilização GPL-2, que apesar de ser um licença pública, apenas permite a interacção de código regido pela mesma licença.

Em suma, estes são alguns dos sistemas de PLN. Tanto estes como outros não apresentados são válidos para a tarefa. O critério de decisão deve prender-se nas necessidades do utilizador, visto que uns possuem mais módulos que outros, uns contém interface gráfica outros são apenas controlados por comandos na consola, existem também as questões da linguagem de implementação. Como será mais abordado adiante, neste projecto iremos trabalhar com o Mallet para proceder à classificação, já que esta ferramenta é das que dispõe de mais algoritmos implementados, o que nos permite realizar vários testes e escolher o que melhor se enquadra ao problema.

RapidMiner

From: NNP	**From: IN**
where: WRB	where: WRB
thou: PRP	**thou: JJ**
art: VBP	**art: NN**
,: ,	,: ,
why: WRB	why: WRB
should: MD	should: MD
I: PRP	I: PRP
haste: VB	**haste: NN**
me: PRP	me: PRP
thence: NN	**thence: VB**
?: .	?: .
Till: NNP	**Till: IN**
I: PRP	I: PRP
return: VBP	return: VBP
of: IN	of: IN
posting: VBG	posting: VBG
is: VBZ	is: VBZ
no: DT	no: DT
need: NN	need: NN
.: .	.: .

Figura 3.1: Comparação de resultados após etiquetagem entre o Stanford NLP e o sistema NLTK para a frase "From where thou art, why should I haste me thence? Till I return of posting is no need.".

RapidMiner é um sistema código aberto para AC, nomeadamente *PD*, *text mining* e análise de negócios. Encontra-se desenvolvido em Java e permite uma série de operações de controlo, manipulação e visualização dos dados. Algumas das suas principais vantagens são:

- Software código aberto (compatível com todos os principais sistemas operativos) que possibilita manipulação de dados eficiente;
- Ferramentas poderosas para a construção gráfica dos dados manipulados;
- Recorre a esquemas de aprendizagem do sistema Weka [10], e a modeladores estatísticos do sistema R;
- Compatibilidade com uma grande quantidade de fontes de dados como o *Excel*,

[10] http://www.cs.waikato.ac.nz/ml/weka/

Access, *Oracle*, *IBM DB2* e *MySQL*;

A sua licença de utilização é a *AGPL*, licença baseada na conhecida *GPL-2*.

R

O R é um ambiente de programação para computação estatística e geração de gráficos. Baseia-se na linguagem S e embora existam diferenças, o código deste é compatível para o R. A licença de utilização desta linguagem é a GPL-2.

O R dispõe de várias funcionalidades de análise de dados e Prospecção de Dados (PD) como análise de séries temporais, modelação linear e não-linear e classificação e *clustering*. De um modo geral, este sistema inclui:

- uma eficaz manipulação e armazenamento de dados;
- um conjunto de operações para cálculos em diferentes estruturas de dados, mas principalmente, operações matriciais;
- um grande conjunto de ferramentas integradas para análise de dados;
- funções para construção de gráficos para amostragem dos dados;
- uma linguagem de programação simples e bem estruturada, que integra todas as instruções básicas conhecidas de outras linguagens;

Mahout

O Apache Mahout é uma outra opção de biblioteca para desenvolvimento de programas de ML. Possui com grande eficácia quando utilizado em paralelo com o Apache Hadoop, tornando-se uma solução escalável para grandes quantidades de dados, mas não se restringe a esta opção. Dispõe de soluções como *clustering*, *collaborative filtering*, recomendação baseada no utilizador ou no item, *decision tree's*, classificadores *Naive Bayes* e o LDA *Latent Dirichlet Allocation* (Duda et al., 2000).

Este produto é distribuído de acordo com a *Apache License, version 2* que permite a integração deste sistema com terceiros que não estejam vinculados a esta mesma licença.

Linguagem Ruby

A linguagem de programação Ruby[11] é uma linguagem de alto nível desenvolvida por Yukihiro "Matz" Matsumoto, criada em 1995. Esta é livre e foi desenvolvida a pensar

[11] https://www.ruby-lang.org/pt/about/

nalgumas fraquezas de outras linguagens existentes na altura. Tudo em Ruby é tratado como um objecto, permitindo a tipos de variáveis como os números ou *boolean* ter métodos próprios. Possui uma longa e detalhada documentação, e é constantemente actualizada.

Hoje em dia, tem vindo a ser bastante utilizada graças à *framework* para desenvolvimento web Ruby on Rails. Existe também uma longa lista de módulos implementados para as mais variadas tarefas, uma vez que o sistema é livre e qualquer um pode dar a sua contribuição. Estes módulos têm o nome de *gems*.

Para este projecto foram utilizadas algumas dessas *gems* de modo a evitar implementar algo que já existe até porque, estando estas disponíveis online, mais facilmente são corrigidas e melhoradas, o que torna este tipo de solução mais vantajosa do que a própria implementação. Assim, são destacadas algumas das *gems* essenciais para este protótipo:

- Chronic[12] - módulo de interpretação de datas para a língua inglesa escrito na linguagem Ruby;
- Feedjira[13] - módulo que permite extrair *feeds* RSS de uma fonte RSS introduzida, que depois converte os *feeds* para objectos Ruby;
- Twitter[14] - módulo que permite extrair *tweets* sobre determinados parâmetros (língua, localização, *keywords*) e que depois os converte para objectos Ruby;
- RDF[15] - módulo totalmente desenvolvido em Ruby que permite a criação de modelos RDF, a criação de grafos, triplos, pesquisas através de *queries* e comunicação com *Triple Stores*;

3.2 Trabalho Relacionado

Nesta secção serão relatados alguns exemplos de investigações e trabalhos já desenvolvidos sobre a área de extracção de eventos e outros dados, a partir de redes sociais (maioritariamente o Twitter) como também de notícias. Como as redes sociais são um cenário do presente, os estudos apresentados são necessariamente recentes. Serão explicadas sucintamente as abordagens utilizadas bem como os resultados obtidos.

[12] https://github.com/mojombo/chronic
[13] https://github.com/feedjira/feedjira
[14] https://github.com/sferik/twitter
[15] https://github.com/ruby-rdf/rdf

3.2.1 Extracção de Informação do Twitter

O sistema (*TwiCal*) (Ritter et al., 2012) foi criado com o objectivo de classificar o tipo de evento sobre um determinado *tweet* (Desporto, Política, Produto, Encontro,...). Este aprende através dos dados, sem qualquer tipo de ajuda ou reforço (aprendizagem não supervisionada), uma vez que se trata de uma rede social onde os categorias de eventos não são lineares, e onde seria necessário uma grande trabalho manual de modo a associar eventos aos tipos. O sistema começa por extrair tuplos de 4 valores por cada *tweet* encontrado, onde são discriminadas entidades, informações do evento, data e tipo. De modo a maximizar a eficácia, a data do momento é cruzada com os dados em estudo de modo a traduzir uma classificação mais fiel, uma vez que vários eventos na mesma data são provavelmente do mesmo género de evento, como também existem certos tipos de eventos que só ocorrem em datas específicas. Na tabela 3.2 estão dispostos os resultados obtidos, onde é possível perceber que para os 100 eventos mais falados no Twitter, o sistema consegue uma boa precisão, e à medida que são adicionados mais eventos e relevância inferior, a precisão cai drasticamente. Foi concluído que a natureza dos dados no Twitter sofre de falta de informação adicional que ajudariam e eliminar o erro no processo de REM.

# calendar entries	ngram baseline	entity + date	event phrase	event type	entity + date + event type
100	0.50	0.90	0.86	0.72	0.70
500	0.46	0.66	0.56	0.54	0.42
1000	0.44	0.52	0.42	0.40	0.32

Tabela 3.2: Resultados da precisão do sistema *TwiCal* (Ritter et al., 2012) para diferentes números de registos.

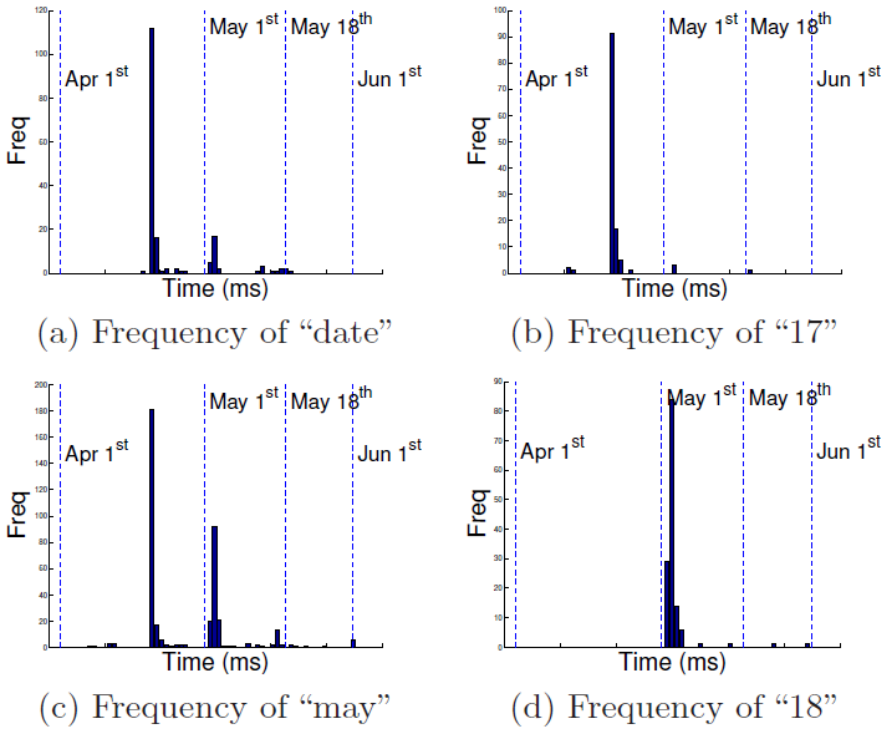
Sobre outra abordagem, foi desenvolvido um *framework* que relaciona *tweets* a nível temporal, já que existe uma grande probabilidade de se referirem ao mesmo evento (Chua and Asur, 2013). Eram fornecidas *queries* para um evento específico em estudo. Posto isto, era realizada uma pesquisa através das *queries* construídas, de modo a encontrar *tweets* que referissem o evento. A tabela 3.3 refere-se a um dos eventos em estudo, neste caso o "Facebook IPO[16]". Na Figura 3.2 é possível visualizar a taxa de *tweets* encontrados ao longo do tempo, que se referem a este evento em particular.No final era esperado compilar um sumário de publicações que abordassem o mesmo acontecimento, de modo a que cada um acrescentasse algo de novo. O resultado obtido demonstrou maior sucesso que o método de

[16] *Initial Public Offer*

Linear Discriminant Analysis (Sá, 2001) ou simplesmente LDA. Foram utilizadas várias fontes de avaliação de resultados, como por exemplo, a *Amazon Mechanical Turk*[17].

(Facebook | FB), IPO
(Facebook | FB), Initial, Public, (Offer | Offering)

Tabela 3.3: Exemplo da representação de conhecimento pela lógica de predicados.



(a) Frequency of "date" (b) Frequency of "17"

(c) Frequency of "may" (d) Frequency of "18"

Figura 3.2: Gráficos com os resultados obtidos em (Chua and Asur, 2013).

TwiNER foi um sistema de REM construído com vista na extracção de entidades de *tweets* (Li et al., 2012). Em primeira instância, os *tweets* eram divididos recursivamente até que apenas restassem segmentos que representavam possíveis entidades. Este processamento recorria à informação presente na Wikipedia[18] e no Web N-Grams[19] para ajudar a encontrar as potenciais entidades. Depois disso, o sistema utiliza toda essa informação de forma não supervisionada, para identificar as entidades encontradas e poder avaliar *tweets* fornecidos pelo *Twitter streaming*. Foram comparados os resultados com conhecidos sistemas de REM (nomeadamente o *Stanford NER* e o *T-NER*) e este obteve resultados semelhantes, o que é satisfatório já que, não existia qualquer tipo de feedback ou auxílio da parte de uma agente humano.

[17] https://www.mturk.com/mturk/
[18] http://dumps.wikimedia.org/enwiki/
[19] http://web-ngram.research.microsoft.com/info/

Outro tipo de classificador foi desenvolvido com o objectivo de detectar entidades em *tweets*, de modo a reconhecer e a diferenciá-las por categorias (Locke and Martin, 2009). O conjunto de dados CoNLL-2003[20] serviu de termo de comparação para com os resultados provenientes do *Twitter*. Como resultado, eram obtidas entidades do tipo *PERSON*, *LOCATION* e *ORGANIZATION*, que correspondem aos 3 tipos de entidades principais. Para o CoNLL-2003 o sistema teve um grande desempenho, enquanto que para o Twitter, sentiu grande dificuldade na identificação de entidades do tipo *PERSON* e *ORGANIZATION*. Foi concluido que devido ao facto de existir uma estruturação de informação, ao contrário do que acontece no *streaming* social, tornou-se imprescindível para a taxa de sucesso atingida. Qualquer tentativa de transferência de conhecimento de um domínio para o outro seria provavelmente fracassada dada a diferença e o rigor presente no registo de linguístico.

Existem também abordagens de *clustering* incremental (Wang et al., 2011). Aqui foi estudada a detecção de eventos através da análise de *features* textuais como visuais. Foram analisados quatro campos essenciais:

- *Timestamp* - se dois eventos distam mais do que uma semana no tempo, a sua similaridade é 0. Caso isto não se verifique, a sua similaridade temporal é calculada segundo a formula $s_t = 1 - \frac{t_1 - t_2}{t_w}$, onde t_w representa o tempo de cada semana em minutos.
- Localização - aqui é calculada a *Great Circle Distance* (of Defence , Navy). Caso esta seja superior a 50 milhas a similaridade é 0, caso contrário a formula aplicada é $s_l = 1 - \frac{GCD}{50}$.
- *Tags* - é calculado o índice de Jaccard (Tan et al., 2005) para conjunto de *tags*.
- Texto - aqui é aplicada similaridade do cosseno (Tan et al., 2005) aos vectores de termo-frequência.

Após o pré-processamento das evidências e obtenção das observações, estas foram combinadas com recurso a pesos de modo a distinguir as mais discriminantes das outras. Depois disto, foi realizada uma fase de filtragem onde foram eliminados os eventos fora intervalo temporal ou espacial estipulado. Os resultados foram razoáveis mas promissores, já que as taxas atingidas eram medianas mas exista espaço de manobra para optimizações. Apesar disso, foi possível determinar que a *precision*, em média, é sempre superior à *recall*.

Em praticamente todos os trabalhos relacionados, foram adicionadas novas funcionalidades, algoritmos de reconhecimento ou até conjuntos de dados de treino para estudar o *corpus*

[20] http://www.cnts.ua.ac.be/conll2003/

presente no Twitter. Foram estudadas mais investigações com abordagens semelhantes às supracitadas como (Zhang et al., 2013), (Tanev et al., 2012), (Becker et al., 2010), (Becker et al., 2011), (Wang et al., 2012) e (Finin et al., 2010), mas que não foram escrutinadas ou por não acrescentar mais valor às já apresentadas ou por não incidirem no problema em questão. Desde lista de entidades, grandes quantidades de comentários para treino, cruzamento de dados com a data da publicação, são algumas das abordagens adoptadas que, acabam por melhorar a precisão na identificação de entidades. O trabalho relacionado foi centrado essencialmente na extracção de dados a partir dos *tweets* uma vez que a quantidade de informação conseguida deste meio é bastante superior às notícias extraídas, como também por ser nestes que residem os maiores obstáculos ao sucesso da EI.

Assim, todo o tipo de informação adicional ou complementar que seja possível encontrar, deverá ser cruzada de modo a colmatar o défice de conteúdo que se regista nos *social streams*. Os sistemas não serão perfeitos a este nível, mas quanto maior for o conteúdo de que estes dispõem para a tomada de decisão, melhor será a eficácia do mesmo.

Capítulo 4

Análise e Especificação

Nesta secção será dado início à abordagem do problema e da solução apresentada. Serão relatados alguns exemplos de cenário da aplicação prática do sistema no quotidiano dos utilizadores. Serão discriminados os requisitos bem como explicada como todos os módulos integrantes , tal como eles se conjugam e cooperam.

4.1 Problema

Era comum os investigadores de Extracção de Informação (EI) dedicarem-se apenas ao domínio noticioso, onde é aplicado o uso devido da linguagem, segundo todas as suas regras gramaticais. Com o crescimento dos *social media*, o domínio foi expandido de modo a albergar a informação presente nas redes sociais, blog's ou e-mail's. O que acontece é que, não existe qualquer imposição para quem efectua um *post* nas redes sociais (razão pela qual existe tanta adesão da parte dos utilizadores). Como tal, novos esforços são realizados de modo a colmatar o défice de formalização nesses registos, por forma a obter a informação neles presente.

Como já foi descrito nos problemas típicos envoltos nestes ambientes, existem vários obstáculos ao Reconhecimento de Entidades Mencionadas (REM): o uso indevido de minúsculas/maiúsculas , abreviaturas, formação de palavras inexistentes, múltiplas línguas num só *post*, falta de pontuação, entre outros. Apesar de todos estes entraves, torna-se importantíssima a exploração e obtenção de informação neste sector já que:

- são bastante populares e activos nos dias de hoje;

- a sua actividade encontra-se em expansão, apesar de já ter atingido os milhões de utilizadores;
- os eventos e as respectivas informações surgem quase instantaneamente e em maior escala nas redes sociais do que nas conhecidas fontes de notícias;
- os emissores e receptores de informação encontram-se em todos os cantos do mundo;
- possibilita o acesso ao *feedback* directo do público sobre determinado contexto;
- possuem variadas características que acrescentam valor ao bloco de texto como associar imagens, localização, *smiles*, *hashtag's* e menções [1];

Tendo por base os sistemas comuns de REM, quando aplicados às redes sociais, a sua taxa de sucesso revela-se fraca. Os sistemas que conseguem analisar correctamente cerca de 90% das frases no contexto jornalístico, podem ter um decréscimo da taxa de sucesso até 50% quando aplicadas em textos mais informais (Poibeau and Kosseim, 2001).

O mundo está em constante mudança. Todos os dias ocorrem novos acontecimentos que alteram o estado do ambiente onde estamos inseridos, porque de alguma forma causam distúrbio. Estes acontecimentos têm um determinado tipo, duração e local e podem envolver pessoas ou organizações. Estes são relatados das mais variadas formas na web dos dias de hoje, através de notícias escritas, vídeos, músicas, jogos ou aplicações.

O problema deste projecto consiste na EI relativa a eventos através de dois meios distintos: as redes sociais e as notícias. De forma a limitar o domínio de resultados, são procurados apenas os eventos portugueses, pelo que serão só explorados textos portugueses. Pretende-se analisar as várias publicações que retratam eventos e agregá-las consoante o seu teor, de modo que possam ser exploradas através da sua categoria e dos seus atributos. São definidos 5 atributos principais para o objecto evento: tipo de evento, data, local, entidades e termos. Não serão diferenciados eventos agendados (eventos desportivos, reuniões, espetáculos) e não agendados (acidentes, conflitos, ...), importa apenas referir que ambos serão contemplados nas várias categorias abrangidas.

O Twitter foi uma das fontes de informação social escolhidas para dar suporte a este problema. Como já foi descrito, é uma das redes sociais de topo dos dias de hoje, com grande adesão por parte da comunidade portuguesa. Para o utilizador comum, é muito fácil publicar conteúdo no Twitter: basta ter uma conta no site, ligação à internet (hoje possível em grande parte dos locais graças aos planos de dados e redes sem fios), e dispositivo com

[1] também conhecido por *mentions*, onde os nomes próprios ou entidades são precedidos pelo caracter "@", indicando que aquela mensagem se destina ao mesmo

ligação à Internet (*smartphone*, *tablet* ou computador). Isto permite que todos os eventos sejam comentados instantaneamente por várias pessoas.

Ao nível do contexto jornalístico, foi escolhido mais do que um único jornal. Ao contrário dos *tweets*, as notícias são muito mais cuidadas na sua escrita, são criadas graças a um pequeno grupo de jornalistas, o que reduz brutalmente o número de publicações comparando como uma rede social de topo. Como tal, foi necessário escolher mais do que um jornal português, não só para aumentar o número de notícias mas também para aumentar a probabilidade de cobrir mais eventos. Estes foram escolhidos consoante o seu prestígio, nível de abrangência e regularidade da tiragem[2]. Assim, foram escolhidos os *feeds* dos jornais: Diário de Notícias[3], Jornal de Notícias[4], Expresso[5] e Público[6]. Para complementar, foram também adicionados alguns *feeds* da página do Google News[7], que é personalizável, e foi configurado para obter notícias portuguesaas a partir de fontes portuguesas.

4.2 Cenários de Utilização

Nesta secção serão abordados alguns exemplos de cenários onde o sistema em desenvolvimento poderá entrar, a forma como este se deverá comportar e de que maneira é que os utilizadores poderão retirar partido das suas funcionalidades no seu quotidiano.

4.2.1 Cenário #1: Manifestação

As manifestações estão na ordem do dia dado o cenário económico que o país atravessa. O descontentamento com o aumento de impostos e os cortes nos vencimentos, levam à rua milhares de pessoas a reivindicar todos os seus direitos e a lutar por melhores condições para sustentar as suas famílias. Estes movimentos são usualmente registados pelos vários meios de comunicação social, como são também relatados em tempo real pelos próprios manifestantes. As redes sociais são um dos principais fóruns de partilha e discussão pública nestas alturas, onde são expressas muitas opiniões acerca do tema ou evento. Paralelamente, surgem várias notícias que relatam todos os acontecimentos, contém testemunhos e tentam narrar o cenário da reivindicação.

[2] http://portadaloja.blogspot.pt/2013/11/imprensa-nacional-tiragens-2013.html
[3] http://www.dnoticias.pt/
[4] http://www.jn.pt/paginainicial/
[5] http://expresso.sapo.pt/
[6] http://www.publico.pt/
[7] https://news.google.pt/

O sistema a desenvolver deve identificar este tipo de ocorrências, apenas através do Twitter ou *feeds* de notícias, já que existem em grande escala. Uma vez que datam de alturas relativamente semelhantes e pertencem todos ao mesmo tipo de acontecimento, o protótipo deve agregar todo o seu conteúdo num só bloco de dados, contendo não só a informação de data, local, tipo e localização, bem como os vários registos interpretados como positivos presentes nas várias fontes de extracção. O interface deve agrupar e disponibilizar todo este bloco de informação ao utilizador, facilitando o acesso a toda a informação referente a um único evento.

4.2.2 Cenário #2: Jogo de futebol

O futebol é o desporto rei em Portugal, move mundos e fundos, e já provou que consegue "parar" o país. Nos dias que antecedem este tipo de espectáculo e claro, no próprio dia do evento, existe tanto uma enchente de notícias como de debates nos *social streams* sobre todo e qualquer detalhe que tenha a ver com os intervenientes no jogo, a história, estatísticas, etc. Quanto maior a dimensão do jogo, maior a afluência e consequentemente o alcance dos comentários efectuados.

Para o sistema em desenvolvimento, torna-se imprescindível alcançar todas essas informações e opiniões prestadas de forma a poder identificar o(s) evento(s) em causa. As datas e localizações são observações que não podem ser descartadas na análise dos dados já que eliminam alguma da ambiguidade presente. Os comentários dos utilizadores que de alguma forma se referem ao evento desportivo, deverão estar associados a este evento complementando a informação já adquirida, ou revelando as várias opiniões acerca de todos as entidades envolvidos. Neste tipo de evento, e uma vez que a dimensão de publicações atingida é bastante significativa, o sistema deverá estar em constante actualização acerca dos pormenores do jogo. Deste modo, o utilizador final poderá aceder a grande parte do conteúdo disponível na web num único local.

4.2.3 Cenário #3: Tempestade

Nem todos os eventos do qual fazemos parte estão previamente agendados. Muitos deles ocorrem de modo inesperado, sem aviso prévio. Muitos deles, acabam por se tornar um fenómeno social ou pela sua dimensão, ou pelo transtorno ou pelo dano causado. Os fenómenos meteorológicos são bons exemplo deste tipo de eventos, que embora sejam capazes de se antecipar, não é possível calcular o seu resultado como é o caso das tempestades. Em

dias de vento, chuva ou trovoada fora do normal, é recorrente aparecerem relatos de habitantes dos locais afectos a à força da Natureza. São publicadas fotografias ou até vídeos para registar e divulgar o acontecimento, mesmo que muitas das vezes estes prejudiquem os meios de transmissão da região. Embora não exista uma localização absoluta neste tipo de ocorrências, mas sim vastas áreas, é possível cruzar todos os dados e perceber se se trata do mesmo evento, aquando das mensagens sobre este mesmo assunto nas redes sociais.

Recorrendo ao protótipo em desenvolvimento, muitos dos conteúdos disponibilizados online como relatos de danos, testemunhos de habitantes afectados ou mesmo dados estatísticos geralmente fornecidos pelos jornais de notícias, deverão concentrar-se num único lugar. Isto acontece porque toda a informação se refere a um único acontecimento. O protótipo deve permitir ao utilizador visualizar os testemunhos extraídas num único local, sem necessidade de uma procura por várias páginas da web.

4.3 Análise de Requisitos

Analisados os casos de uso num sentido mais abstracto, é necessário especificar as funcionalidades presentes no protótipo e descrevê-las. Como será explicado mais adiante, o sistema centra-se na EI a partir de *tweets* e *feeds*. A partir daqui, através de vários processos de REM e classificação (estes procedimentos serão explicados em detalhe no capítulo 6), deverá compreender quais os eventos a ser relatados, agregar todo o seu conhecimento do modo mais possível, e possibilitar a sua visualização ao utilizador através de 3 funcionalidades distintas: *browsing*, recomendação e pesquisa semântica.

1. Extrair notícias online
    - **ID** - 01
    - **Descrição** - Acesso e extracção de *feeds* de vários sites de jornais portugueses, para conseguir extrair as notícias. O sistema verifica periódica e automaticamente se existem actualizações nas referidas fontes.
2. Extrair *tweets*
    - **ID** - 02
    - **Descrição** - Acesso a uma conta do Twitter através da Application Programming Interface (API) para conseguir extrair os *tweets* portugueses. São procurados os *tweets* tanto através de *queries* com *keywords* como através do *streaming*. Aspectos como a data, localização e referências são utilizados como

factor de pesquisa. O sistema extrai automaticamente novos *tweets* para manter os eventos actualizados.

3. Extracção de informação de eventos: notícias online
   - **ID** - 03
   - **Descrição** - Extracção de informação a partir da notícia extraída. A notícia deverá ser referente a um evento português, e caso isto não se verifique, a notícia é descartada. São extraídos dados relacionados com o evento como a data, a localização, os termos, as entidades e classificados os tipos do evento.
4. Extracção de informação de eventos : *tweets*
   - **ID** - 04
   - **Descrição** - Extracção de informação a partir do *tweet* extraído. O *tweet* deverá ser referente a um evento português, e caso isto não ser verifique, o *tweet* é descartado. São extraídos dados relacionados com o evento como a data, a localização, os termos, as entidades e classificados os tipos do evento.
5. *Browsing* de informação
   - **ID** - 05
   - **Descrição** - Pesquisas sobre os dados dos eventos, *feeds* e *tweets* guardados no *Triple Store*. Esta pesquisa é realizada através de *queries* simples onde apenas são fornecidos os valores como *input*, e procurados esses elementos directamente no *Triple Store*. Ao nível do interface gráfico, é possível ao utilizador aceder a listagem de eventos, *feeds* e *tweets* ou acedê-los individualmente para ter acesso a todos os seus dados.
6. Pesquisa semântica de informação
   - **ID** - 06
   - **Descrição** - Pesquisas semânticas sobre os dados dos eventos, *feeds* e *tweets* guardados no *Triple Store*, com recurso a entidades, data, localizações e outro tipo de dados de suporte à pesquisa. Esta funcionalidade está presente no interface gráfico e permite pesquisas inteligentes, já que o utilizador pode inserir qualquer tipo de dados e este será identificado pelo sistema. Os resultados são apresentados por ordem descendente de semelhança para com o *input* inserido.
7. Recomendação de informação
   - **ID** - 07

- **Descrição** - Recomendação de eventos, *feeds* e *tweets* ao utilizador relacionados com o conteúdo em visualização. Só é recomendado novo conteúdo ao utilizador aquando da visualização individual de um item. São portanto recomendados os itens que maior semelhança apresentam com o item em visualização no momento.

8. Alertas sobre eventos
    - **ID** - 08
    - **Descrição** - Notificação ao utilizador da chegada de nova informação relativa a eventos, *feeds* ou *tweets*. É possível definir um alerta em específico através de *keywords*, que será disparados assim que novos itens sejam extraídos com as *keywords* presentes nos valores dos seus atributos.

4.4 Arquitectura

A arquitectura do sistema foi delineada segundo as várias exigências inerentes ao problema. Era sabido que seria necessário construir um sistema de extracção e classificação de conteúdo online. Este mesmo conteúdo seria posteriormente guardado para que pudesse ser acedido e disponibilizado aos utilizadores. Isto pressupõe tanto uma Base de Dados, responsável por guardar os dados e permitir a actualização dos mesmos, um interface que permita a visualização dos dados de forma mais prática e intuitiva, e um servidor sempre disponível que assegurasse a disponibilização destes 2 últimos serviços.

O problema foi pensado do ponto de vista de Engenharia de Software: desde a linguagem de programação a utilizar, que tipo de base de dados e como é que esta seria integrada, as ferramentas a incorporar, e os serviços que lhe poderiam ser adicionados. A solução foi conceber uma aplicação web, desde a sua raiz até ao interface de utilizador. Todas estas particularidades do sistema serão analisadas em maior detalhe ao longo deste capítulo.

4.4.1 *Model-View-Controller*

Como em grande parte das aplicações web desenvolvidas hoje em dia, o sistema necessita de ser construído através de um modelo de software bem conhecido: o Model View Controller (MVC). Tal como o nome indica, este padrão de arquitectura distingue 3 secções principais nos programas: o *Model*, o *View* e o *Controller*. Na figura 4.1 estão esquematizadas as

várias secções deste tipo de arquitectura e a forma como estas comunicam entre si, desde o pedido do utilizador até lhe ser devolvida a resposta.

- *Model* - corresponde a todo o conjunto de funcionalidades que permitem o acesso e alteração dos dados e respectivo valor. Trata-se de uma camada não visível ao utilizador já que não tem controlo directo sobre a Base de Dados;
- *View* - corresponde à camada de visualização do programa que apenas se dedica à apresentação da informação da forma mais adequada e usável possível ao problema, respondendo aos pedidos provenientes do *Controller*;
- *Controller* - toda a camada de controlo de fluxo do programa e de informação. Este é o elo de ligação entre as 3 camadas já que se encontra entre os outros 2, e é também o motor de todo o sistema já que trata da execução de tarefas, requisita os dados ao *Model*, recolhe os resultados e envia para o *View* de modo a ser apresentado ao utilizador.

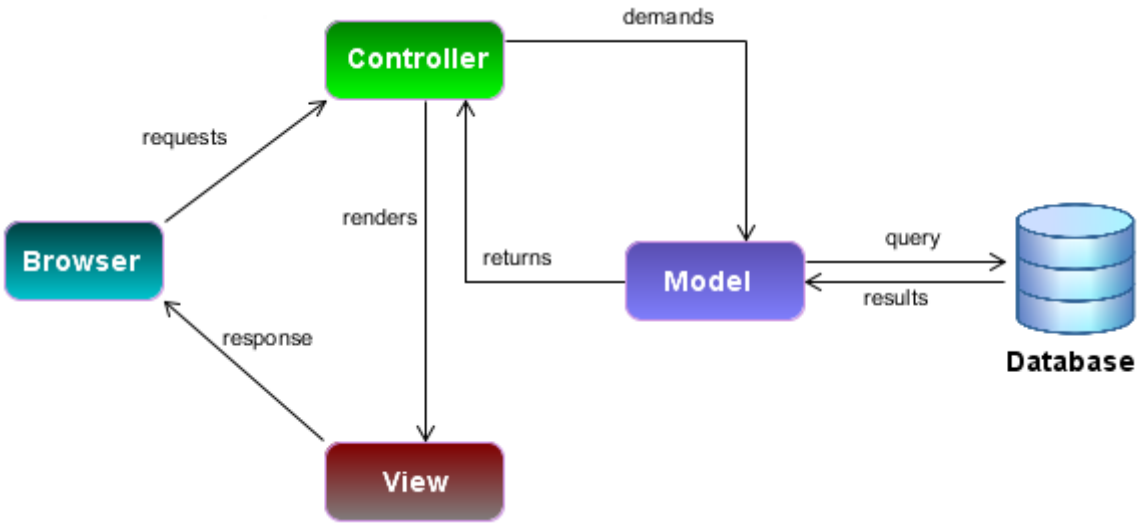


Figura 4.1: Esquema da arquitectura MVC

Hoje em dia, muitas das *frameworks* de desenvolvimento possuem este tipo de arquitectura, o que agiliza o processo de construção de serviços Web. Exemplos como em *Python*[8] com o *Django*[9], *Ruby on Rails*[10], *Struts*[11] do próprio *Java*[12] utilizam este tipo de arquitectura.

Existem vantagens na utilização de *frameworks* para desenvolvimento destes sistemas como a redução do tempo do programação já que incluem funções que "encurtam" o processo,

[8] http://www.python.org/
[9] https://www.djangoproject.com/
[10] http://rubyonrails.org/
[11] http://struts.apache.org/
[12] https://www.java.com

o conjunto de funcionalidades pré-implementadas que expandem o leque de opções, a segurança e a qualidade do sistema seguem normas construídas desde a raíz, a completa documentação de todo o sistema e, o facto de se tratarem de ferramentas actuais que se encontram e constante actualização e discussão, possibilitando a resolução de problemas do modo mais rápido possível.

As desvantagens centram-se no elevado nível da linguagem, que retiram alguma da liberdade para adaptações necessárias ao objectivo do sistema, assim como a perda de escalabilidade e velocidade já que, embora o código se encontre optimizado, não é possível criar a e adaptar o sistema desde o mais baixo nível, de acordo com as suas necessidades.

4.4.2 Descrição do sistema

Os dados no sistema em desenvolvimento encontram-se em primeira instância nas respectivas fontes de informação, sendo depois extraídos para que possam ser trabalhados, armazenamento na Base de Dados/*Triple Store* e finalmente sejam executadas operações de *browsing*, recomendação, pesquisa semântica e alertas sobre os mesmos.

Relacionando estes conceitos com o MVC, podemos concluir que o *Model* diz respeito a todo o sistema de armazenamento de dados e respectivas operações de criação, leitura, actualização e destruição (Create Read Update Delete (CRUD)) para com o sistema de gestão de dados. Na arquitectura apresentada, este refere-se ao Módulo de Gestão de Dados (ver Figura 4.2). Com o objectivo de simplificar a implementação, este é um módulo único que lida com todos os acessos, acabando por não existir problemas de concorrência nem de replicação de dados.

O *Controller* é o motor de sistema que controla todas as operações a executar, recebe alertas para os novos eventos nas fontes de informação, processa a informação e recomenda outros eventos ao utilizador, realiza o reconhecimento de entidades, data e outro tipo de dados. Assim, este sector está presente no sistema nos vários Módulos de controlo: *Browsing*, Pesquisa, Recomendação, Alertas e Detecção de Eventos. Todos estes módulos podem realizar pedidos de leitura aos dados presentes no servidor, sempre através da comunicação com o módulo de gestão de dados.

O *View* refere-se da camada de apresentação, isto é, desde os *layouts* à forma como os dados são dispostos através de texto simples ou gráficos, da forma como são conjugados e mostrados ao utilizador. Diz respeito à primeira de todas as camadas, neste caso ao interface gráfico (GUI) que será o único módulo a que o utilizador terá acesso e pelo qual

este pode interagir com a informação.

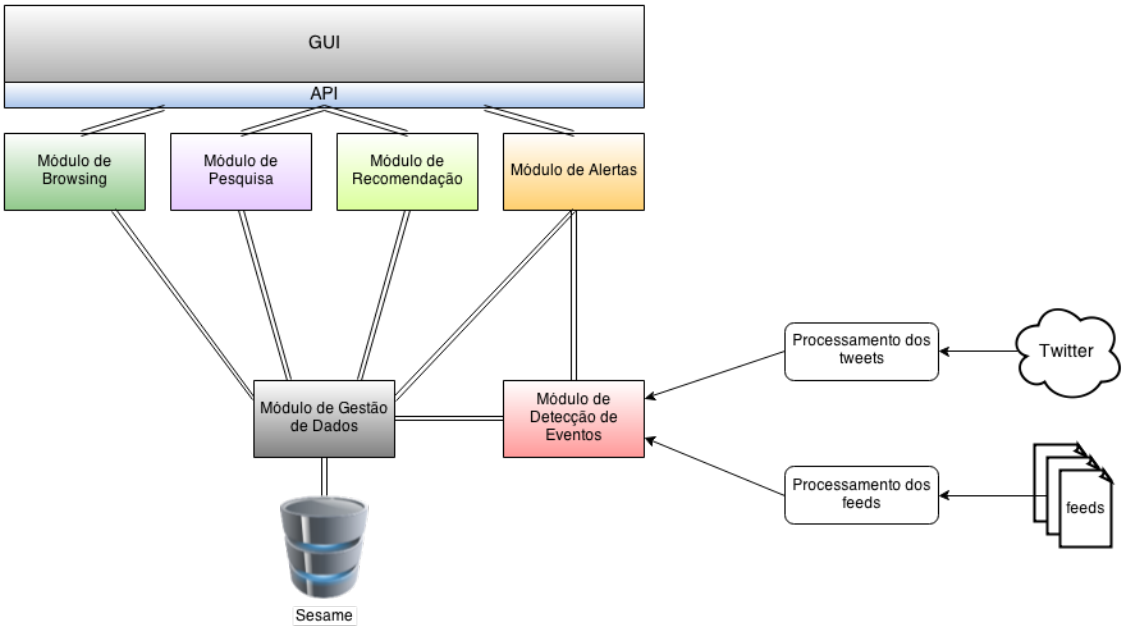


Figura 4.2: Esquema da arquitectura do Sistema

Como também já foi referido, o sistema necessitava de ser alojado num servidor e então foi incorporado na máquina "Mustang" presente Departamento de Engenharia Informática da FCTUC. O link de acesso à aplicação é `mustang.dei.uc.pt:3000`.

Como foi necessário utilizar um *Triple Store*, foi necessário o desenvolvimento de uma ontologia de forma a categorizar o conhecimento extraído. Esta serviu de esquema de armazenamento e relacionamento das várias categorias (classes) e respectivos atributos (propriedades) do conhecimento. O *Triple Store* utilizado foi o *Sesame*[13] que permite a pesquisa e análise de de dados Resource Description Framework (RDF), suporta SPARQL[14] e é mais robusta que muitas outras soluções no mercado.

Sendo esta a secção de arquitectura, apenas foi retratado um esboço da mesma. A descrição detalhada de cada um dos componentes do sistema, a forma como operam e comunicam é abordada no capítulo 5.

[13]http://www.openrdf.org/
[14]http://www.w3.org/TR/sparql11-query/

4.4.3 Sesame

Existem várias opções de armazenamento e pesquisa de dados RDF (ou os chamados *Triple Stores*) como IBM DB2[15], Apache Jena[16], Oracle[17], Garlik 4store[18], YARS2[19], Mulgara[20] e Sesame.

Tal como foi supracitado, foi escolhido este último para este estágio, sistema que inicialmente foi desenvolvido por uma companhia de nome Aduna[21] num projecto que findou em 2001 de nome "On-to-Knowledge". Segue os princípios impostos pelo W3C[22], tratando-se de uma solução bastante robusta, desenvolvida através da linguagem Java, totalmente configurável e que dispõe de várias opções ao nível de inferência, *querying*, armazenamento de triplos em disco ou em memória. Suporta tanto a linguagem SPARQL bem como SeRQL[23]. Confere elevada escalabilidade e uma boa performance nas pesquisas efectuadas sobre os dados armazenados.

Os critérios de escolha desta ferramenta foram a fácil implementação, a possibilidade de acesso aos dados alojados através de uma plataforma própria que facilitava o controlo de erros, a existência de "gems" para Ruby que comunicassem com este tipo de *Triple Store* e o facto de esta ferramenta já se encontrar disponível no servidor "Mustang".

4.4.4 Ontologia Adoptada

Sendo o sistema desenvolvido um agregador de conhecimento sobre o domínio de todos eventos a ocorrer em tempo real, foram procuradas ontologias que satisfizessem o seu objectivo. Foram retiradas várias conclusões das ontologias encontradas de modo a conceber uma específica para este projecto. Até encontrar a ontologia final, existiram vários estados intermédios de ontologias que serão abordadas adiante.

Em primeira instância, os eventos foram categorizados segundo 3 classes de 2ª ordem: casuais, agendados ou outros (ver Figura 4.3). Os eventos casuais eram todos os acontecimentos que ocorrem sem conhecimento prévio da data e hora (ver Figura 4.4). Os eventos

[15] http://pic.dhe.ibm.com/infocenter/db2luw/v10r1/topic/com.ibm.swg.im.dbclient.rdf.doc/doc/c0059661.html
[16] http://jena.apache.org/
[17] http://www.oracle.com/technetwork/database-options/spatialandgraph/overview/rdfsemantic-graph-1902016.html
[18] http://www.4store.org/
[19] http://sw.deri.org/2007/02/swsepaper/iswc2007.pdf
[20] http://www.mulgara.org/
[21] http://www.aduna-software.com/
[22] http://www.w3.org/
[23] http://www.w3.org/2001/sw/wiki/SeRQL

agendados referiam-se a todos os tipos de ocorrências ordinárias, com conhecimento atempado da data e hora (ver Figura 4.5). A categoria "outro" foi colocada para experimentação caso algum caso não estivesse contemplado, mas foi excluída por não ter sentido.

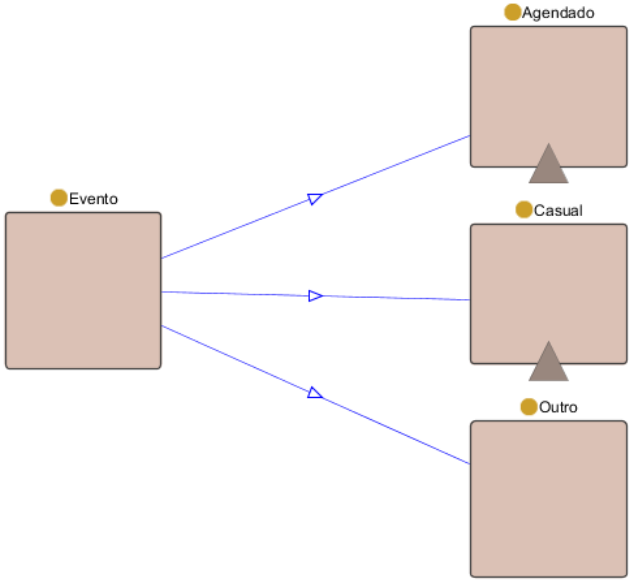


Figura 4.3: Representação da primeira versão de toda a ontologia.

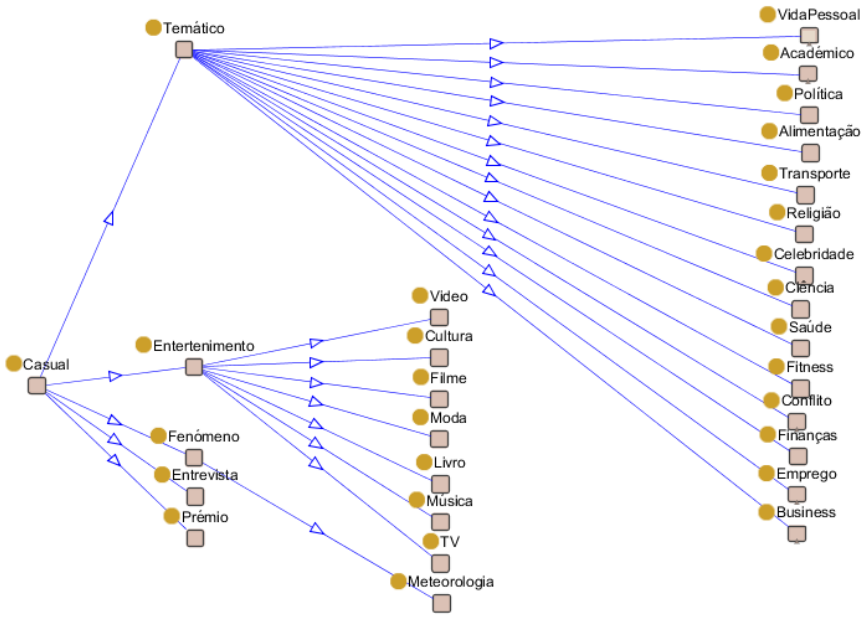


Figura 4.4: Representação da primeira versão da ontologia, a partir da classe "Casual".

Com o início da implementação e como será explicado na mesma secção, foi necessário reduzir o número de classes tanto para eliminar cenários ambíguos como também para melhorar os resultados da classificação de tipos. Foram retiradas as classes "agendado", "não

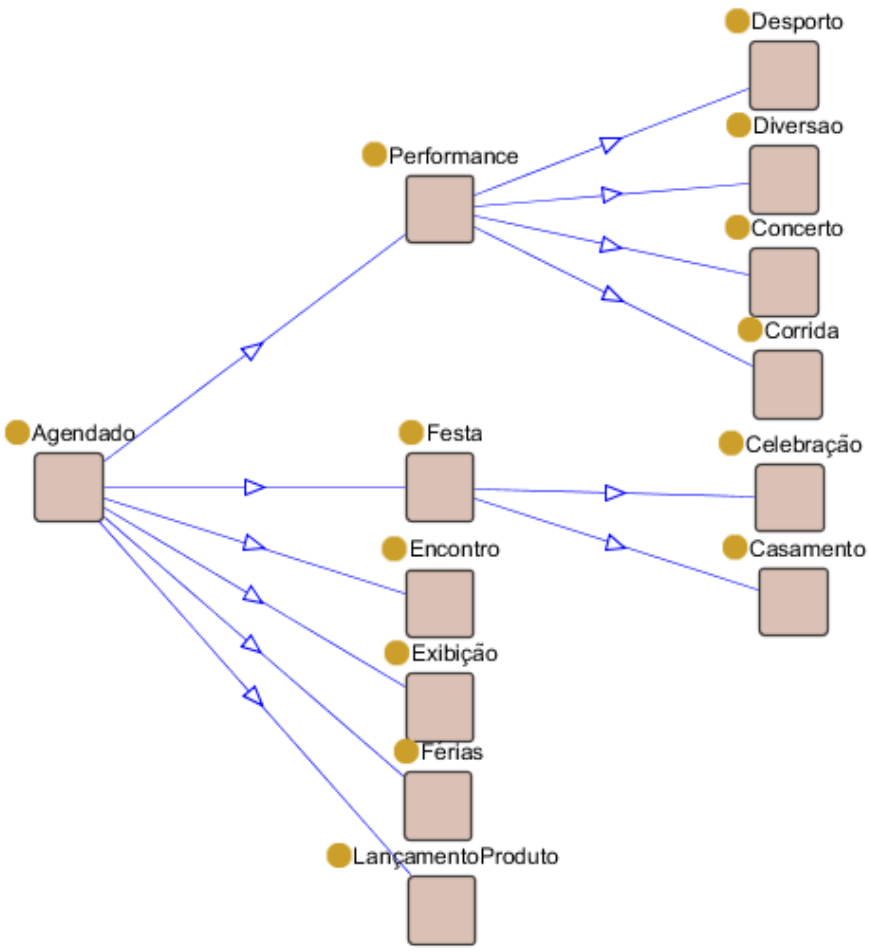


Figura 4.5: Representação da primeira versão da ontologia, a partir da classe "Agendado".

agendado" e "outro" visto não causarem grande relevância para os sub-tipos encontrados e não terem grande sentido de existência. São 12 os diferentes tipos de eventos considerados pelo sistema (ver figura 4.6):

- acidente - acidentes rodoviários, catástrofes naturais, estragos provocados pelo mau tempo;
- celebração - festas de comemoração, romarias, festas de aniversário, Natal, Páscoa, festas académicas;
- cerimónia - missas, casamentos, funerais, bodas de prata ou ouro, condecorações, galas;
- concerto - espectáculos musicais ao vivo, festivais, actuações musicais ao ar livre;
- desporto - todos os eventos desportivos como jogos, corridas, actuações, competições;
- encontro - encontros nacionais, reuniões, conselhos, assembleias, congressos;

- espectáculo - actuações culturais como teatro, magia, musicais, fogos de artifício, circos;
- exibição - galerias de arte, exposições, feiras, demonstrações
- judicial - detenção judicial, julgamentos, libertação ou prisão de reclusos;
- manifestação - manifestações, motins, bozinões;
- natural - estados e previsões meteorológicas, sismos, tornados, tempestades, fenómenos atmosféricos;
- política - tomadas e destomadas de posse, eleições, congressos, discursos políticos;

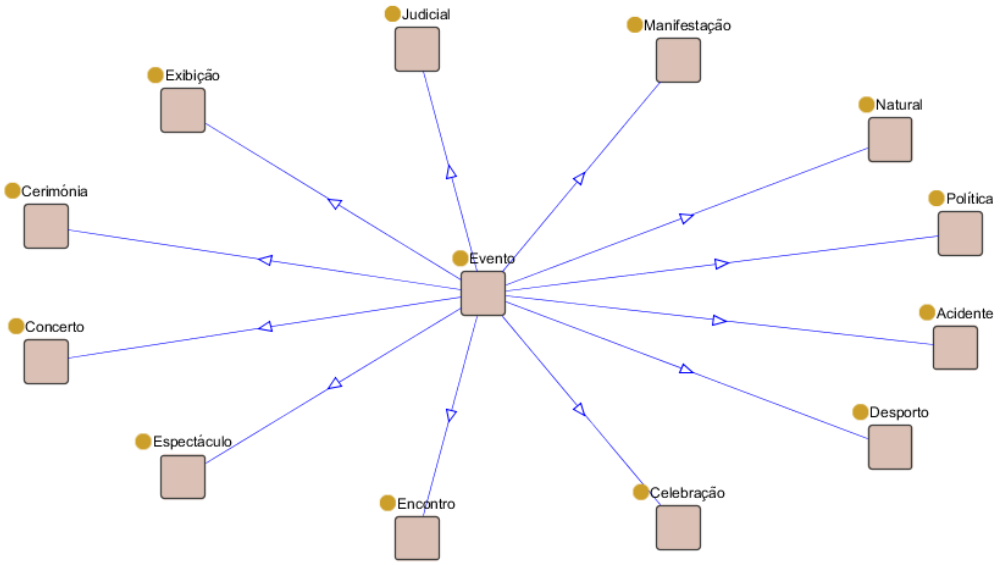


Figura 4.6: Representação da actual ontologia adoptada.

Capítulo 5

Implementação

Neste capítulo será explicado todo o processo de concepção do protótipo, o fluxo do programa e dos dados pelos vários módulos que compõem o sistema, já apresentados na subsecção Arquitectura (4.4). Em primeira instância é explicado todo o processo de extracção de dados, a sua transformação até ao momento em que estes populam o *Triple Store*, percorrendo as várias etapas. São também relatados algumas implementações que funcionam de base para alguns processos como o "Interpretador de Datas" e a "Semelhança de Itens". De seguida são abordados todos os 6 módulos que compõem o sistema bem como o interface gráfico desenvolvido.

5.1 *Workflow* de Extracção de Dados

Como já foi referido, existem dois tipos diferentes de informação: uns provenientes de *tweets* e outros dos *feeds* de notícias. A primeira parte do processo é a aquisição da informação em bruto, de onde serão extraídos os dados que serão tratados pelo sistema. Para esta tarefa foram criados dois extractores diferentes: estes são iniciados por um único comando que faz o *download* das publicações e as devolve na respectiva lista. O *workflow* de extracção de dados é a fase que segue.

Foi construído um processo único e comum para extrair todos os dados às duas classes de objectos. Isto facilitou tanto o processo de desenvolvimento deste sector do protótipo, a correcção de erros bem como permitiu a não repetição de código. Apesar de serem tratados da mesma forma, tanto os *tweets* como os *feeds* continuam a ser dois objectos distintos no final do processo, já que nos interessou manter a sua identidade para as fases futuras.

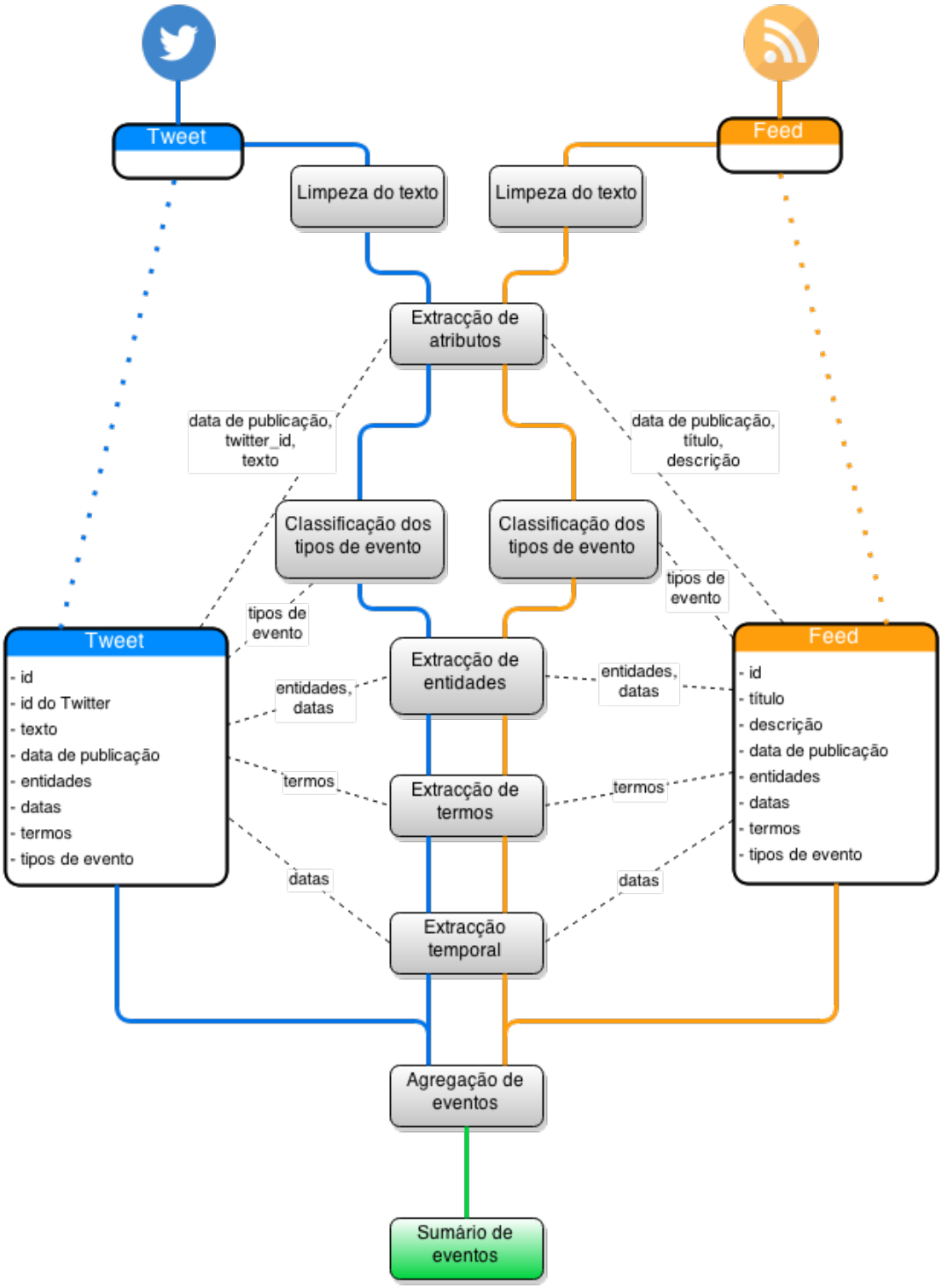


Figura 5.1: *Workflow* do processo de extracção de informação

Na figura 5.1 está esquematizado todo o fluxo do programa, as várias fases, os atributos tratados em cada uma das etapas e as classes dos dois tipos de publicação.

***Tweets* não tratados**	***Tweets* tratados**
acunteçeme a tda a ora http://t.co/SziltiAGZW	acunteçeme a tda a ora
A famosa francesinha do porto. #francesinha #portugal #porto #visitporto #ilovporto #food... http://t.co/MOlNJ5RhjV	A famosa francesinha do porto. ...
@EduardaDiiias porque é que o teu irmão está como head?	EduardaDiiias porque é que o teu irmão está como head?
É assim que se como em Portugal ! (+ sobremesa) #Baptismo #restaurante #ComidaPortuguesa #Bife... http://t.co/hTzQNo0wHR	É assim que se como em Portugal ! (+ sobremesa) ...
Já andamos de tanga... http://t.co/5UuEsVAcTW	Já andamos de tanga...
@PaulaFCFerreira pega qualquer uma tudo fica-te bem :)	PaulaFCFerreira pega qualquer uma tudo fica-te bem :)

Tabela 5.1: Exemplos de *tweets* antes e depois da limpeza de texto

5.1.1 Limpeza do texto

Nesta etapa procede-se à filtragem e tratamento do texto através de expressões de regulares para cada um dos casos:

- *feeds* - são removidos espaçamentos desnecessários, *links* externos ou pedaços de código HTML.
- *tweets* - são removidos espaçamentos desnecessários, *links* externos, pedaços de código HTML, *retweets*, *hashtag's* e o caracter "@" das referências.

Para o caso das referências dos *tweets*, foi decidido apenas remover o caracter "@" já que estas poderão referir-se a entidades ou locais, o que justifica a sua não exclusão por completo. Estão presentes alguns exemplos do tratamento que é exercido ao texto na tabela 5.1.

5.1.2 Extracção de atributos

Nesta fase, são guardados os atributos relevantes dos objectos provenientes da *gems* Feedjira e Twitter:

- *feeds* - é guardado a data de publicação, o título e a descrição
- *tweets* - é guardado a data de publicação, o ID do Twitter e o corpo do texto

5.1.3 Classificação dos Tipos de Evento

Esta é uma das etapas mais relevantes deste processo, já que procede à extracção de um ou mais tipos de evento de cada publicação. Como é possível ver na subsecção da Ontologia Adoptada (4.4.4), na fase final da implementação foram consideradas 12 diferentes categorias de eventos. Para que uma publicação seja aceite pelo sistema, tem de conter pelo menos uma dessas categorias. Várias foram as etapas que antecederam a implementação desta funcionalidade do sistema, isto porque, e tratando-se de classificação, foi necessário um longo processo de recolha de dados de teste, que foram constantemente aperfeiçoados. Tudo isto será explicado em maior detalhe na secção 6.1.

O Mallet foi a ferramenta utilizada para classificar os tipos das publicações. Foram testados diferentes algoritmos disponíveis sobre o conjunto de dados criado para perceber qual gerava melhores resultados. Estes resultados também serão abordados com maior detalhe na Experimentação (secção 6). Assim, foi escolhido o algoritmo *Naive Bayes* para classificar os *feeds* e o algoritmo *Maximum Entropy* para classificar os *tweets*. No total foram criados 24 classificadores distintos: 12 para os *feeds* e outros 12 para os *tweets*.

Depois de implementado e testado o classificador, este foi incorporado no fluxo de extracção de dados. O sistema procede à classificação das publicações através de 12 classificações diferentes, cada uma referente a cada tipo de evento. Uma publicação pode ter entre 0 a 12 tipos de eventos: tanto pode não conter tipos de evento, como pode ser classificada positivamente em todos os tipos. Para as publicações que não obtiveram qualquer tipo de evento atribuído, são descartadas nesta fase.

5.1.4 Extracção de Entidades

Para a tarefa de extracção de entidades, foi incorporado um Sistema de Reconhecimento de Entidades Mencionadas (SREM) português que esteve a ser desenvolvido numa outra dissertação paralela a esta. As entidades podem ser de 5 diferentes categorias, seguidas de alguns exemplos de possíveis entidades desse mesmo tipo:

- Local - “Lisboa”, “Beira Interior”, “Porto Avô”;

- Organização - "Associação Académica de Coimbra", "O Público", "Fundação de Serralves";
- Pessoa - "José Alberto Carvalho", "Álvaro Siza Vieira", "Isabel";
- Tempo - "há muito tempo", "fevereiro", "verão";
- Valor - "euro", "milhões", "anos".

As entidades são classificadas ao nível das palavras, pelo que se existir uma entidade composta por um conjunto de palavras, esta será separada pelo SREM. Isto obrigou à implementação de uma função que agrega palavras consecutivas com a mesma classe de entidade, numa só. Posto isto, estas passariam a fazer parte dos atributos da publicação. Nesta fase é também avaliado se o evento é ou não português através das entidades da categoria "Local": caso possuísse locais e pelo menos um deles estivesse contemplado na lista de terras e regiões portuguesas, este seria aprovado; caso não possuísse qualquer local, também seria aprovado; caso possuísse locais e nenhum estivesse contemplado na lista de terras e regiões portuguesas, este seria descartado.

5.1.5 Extracção de Termos

Uma vez que já era utilizado um SREM para a extracção de entidades, e este já dispunha de um *POS Tagger*, aproveitou-se essa sua funcionalidade para obter os termos presentes nos textos dos *feeds* e dos *tweets*. Foram definidas algumas regras pelas quais seriam procurados os termos. Tal como na extracção de entidades, o *POS Tagger* opera ao nível das palavras o que significa que se existirem termos compostos estes seriam agregados posteriormente, caso a sua composição respeitasse alguma das seguintes regras. É dada prioridade às regras de maior tamanho uma vez que podem existir situações de ambiguidade. De seguida, são apresentadas as referidas regras de extracção de termos com prioridade a diminuir no sentido descendente:

- Termo → PROP PRP PROP PROP;
- Termo → PROP PRP PROP;
- Termo → PROP PROP;
- Termo → PROP;
- Termo → N PRP N;

- Termo → N ADJ;
- Termo → N N;
- Termo → N.

5.1.6 Extracção temporal

Findada a extracção de entidades e já conhecidas as entidades do tipo "Valor", foi necessário traduzir cada uma destas referências temporais para a data a que correspondiam. De modo a melhorar os resultados desta etapa, em complemento à análise realizada pelo SREM, foi construída uma função que realiza uma nova procura de datas segundo algumas expressões regulares, já que em alguns casos, as datas não eram detectadas. Conhecidas todas as referências temporais presentes nos textos, a interpretação das datas fez-se com recurso à modificação de uma *gem* que interpreta datas para a língua inglesa. Este tópico será abordado com maior detalhe na secção 5.2. Se não existirem quaisquer referências temporais na publicação, é assumida a data da própria publicação como a data do evento.

5.1.7 Agregação de eventos

Após toda a extracção de dados das publicações, é gerado o objecto da mesma. Antes de agregar a qualquer evento, é realizada uma verificado se já existe tal publicação no servidor (através do seu texto), de modo a evitar publicações repetidas. Feita a verificação da unicidade, é atribuído um ID único à publicação. Depois procura-se entre os eventos já existentes no servidor a qual deles a publicação mais se assemelha (a função de semelhança é abordada com maior detalhe na secção 5.3). Os eventos são percorridos por ordem de entrada no servidor.

Foi determinado um *threshold* mínimo para uma publicação pertencer a um evento. Encontrado o evento mais semelhante com a publicação, este passa a fazer parte do evento, e são actualizados os seus dados. Se não existir semelhança suficiente com nenhum dos eventos disponíveis, um novo evento é criado e adicionado ao servidor.

Data	Chronic
today	2014-08-16 22:00:00 +0100
tomorrow	2014-08-17 12:00:00 +0100
last year	2013-07-02 13:00:00 +0100
october	2014-10-16 12:30:00 +0100
summer	2014-09-04 00:00:00 +0100
friday 13:00	2014-08-22 13:00:00 +0100
mon 2:35	2014-08-18 14:35:00 +0100
1st of sep	2014-09-01 12:00:00 +0100
in 2 months	2014-10-16 21:17:07 +0100

Tabela 5.2: Exemplos de utilização da *Chronic*

5.2 Interpretador de Datas

Com o objectivo de fazer a tradução de referências temporais para a data absoluta, foi necessário construir um interpretador de datas. Na tentativa de rentabilizar o tempo de destacado para a implementação deste sector do programa, foram procuradas soluções (de código aberto) disponíveis para a linguagem Ruby. Outra das vantagens de não construir o interpretador de raíz foi o facto das hipóteses já existentes serem constantemente actualizadas e corrigidas através de várias contribuições, o que as torna bastante eficazes e robustas.

Foi encontrada uma *gem* para Ruby de nome *Chronic*[1]. Esta recebe como *input* uma referência temporal e converte para uma data absoluta. Por defeito, o dia em questão é tomado como valor de referência, mas é também possível estipular um outro qualquer dia. O sistema consegue converter casos simples de datas como "hoje", "ontem", "dia 31" (onde é assumido o ano e mês actual), "13:00" (onde é assumido o dia, mês e ano actual), "há 2 dias", entre outros. Para casos mais complexos, nem sempre o sistema consegue interpretar o valor da data absoluta, acabando por não devolver nada. Na tabela 5.2 encontram-se alguns exemplos da sua utilização.

Apesar da língua portuguesa e inglesa descenderem da mesma língua materna, estas têm algumas diferenças, o que fez com que a conversão do interpretador de datas de inglês para português, fosse para além de um simples processo de tradução de palavras ou expressões. Para converter a *gem Chronic* para português foi necessário:

[1] https://github.com/mojombo/chronic

Data	ChronicPT
hoje	2014-08-16 22:00:00 +0100
amanhã	2014-08-17 12:00:00 +0100
último ano	2013-07-02 13:00:00 +0100
outubro	2014-10-16 12:30:00 +0100
verão	2014-09-04 00:00:00 +0100
sexta 13:00	2014-08-22 13:00:00 +0100
seg 2:35	2014-08-18 14:35:00 +0100
1º de set	2014-09-01 12:00:00 +0100
10 set	2014-09-10 12:00:00 +0100
daqui a 2 meses	2014-10-16 20:19:09 +0100
4 da manhã	2014-08-16 04:00:00 +0100
esta noite	2014-08-16 18:30:00 +0100
quarta-feira última semana	2014-08-06 12:00:00 +0100

Tabela 5.3: Exemplos de utilização da *ChronicPT*

- traduzir as expressões regulares responsáveis por apanhar os dias da semana, meses, referências temporais relativas ("hoje", "depois de amanhã", "daqui a um ano") e estações do ano;
- traduzir os números cardinais por extenso ("duas horas", "uma dúzia de dias");
- traduzir os números ordinais por extenso ("primeiro dia deste mês");
- ajustamento das regras gramaticais ("last year" para "ano passado").

5.3 Semelhança de Itens

O projecto consiste num "agregador" de eventos que se rege pela semelhança que as várias publicações apresentam. Pode ser comparada a semelhança entre quais quer dois itens: evento, *feed* ou *tweet*.

A distância entre os mesmos dois atributos de dois itens está apresentada na equação 5.3 e estes podem ser os tipos, datas, locais, entidades e termos. Esta distância é realizada através de uma comparação simples dos valores dos atributos, à excepção da data: para efeitos de comparação de datas, e uma vez que se tratam de eventos, quando duas datas estão separadas por mais de 48 horas, são consideradas diferentes; se duas datas estiverem

dentro do mesmo intervalo de 48 horas, são consideradas iguais. Foi estipulado um valor de intervalo de 48 horas, uma vez que a maioria das publicações sobre determinado evento, se concentra nesse curto espaço de tempo.

A semelhança entre dois itens consiste no somatório da distância pesada dos vários atributos, e está apresentada na equação 5.3.

$$dist(A_i, B_i) = \frac{A_i \cap B_i}{A_i \cup B_i} \tag{5.1}$$

$$Sim(A, B) = \sum_{i=1}^{5} w_i dist(A_i, B_i) \tag{5.2}$$

Na secção de agregação de eventos (6.4) estão apresentado os resultados dos pesos presentes na equação.

5.4 Interface Gráfico

Para efeitos de utilização, foi tentado implementar um interface que fosse simplista mas ao mesmo tempo prático. Não sendo este um dos objectivos do projecto, não foi despendido muito tempo na questão do Design do interface. De modo a encurtar o tempo de implementação, foi utilizada a *framework* Bootstrap[2]. Este contém:

- um menu no topo para navegar pelas várias classes de objectos(*browsing*, ver figura 5.2);
- um menu lateral para navegar pelos vários tipos de eventos existentes com contagem automática do número de eventos do referido tipo (*browsing*, ver figura 5.2);
- uma barra de pesquisa no topo (pesquisa semântica, ver figura 5.2);
- uma página de amostragem dos atributos do elemento seleccionado, ver figuras 5.4 e 5.5;

[2]http://getbootstrap.com/

- dependendo do conteúdo em visualização, são recomendados outros itens semelhantes no fundo da página (recomendação, ver figura 5.7);
- *pop-ups* informativos quando um alerta é disparado ou novos dados estão disponíveis (alerta, ver figura 5.11).

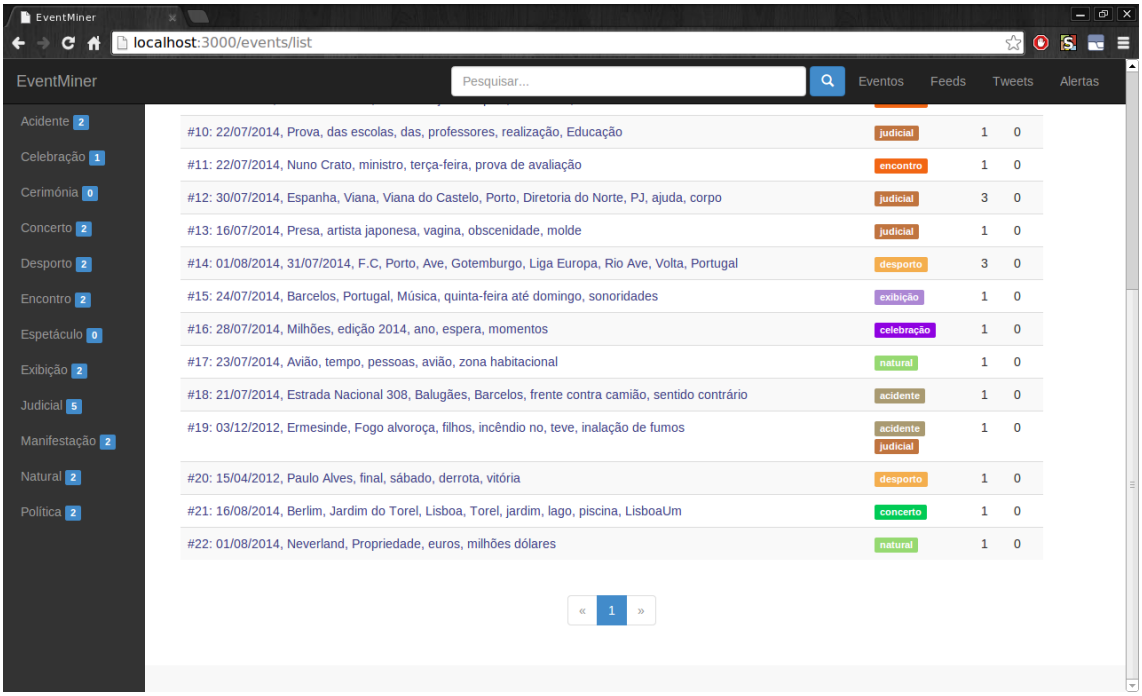


Figura 5.2: *Screenshot* do módulo de *browsing*

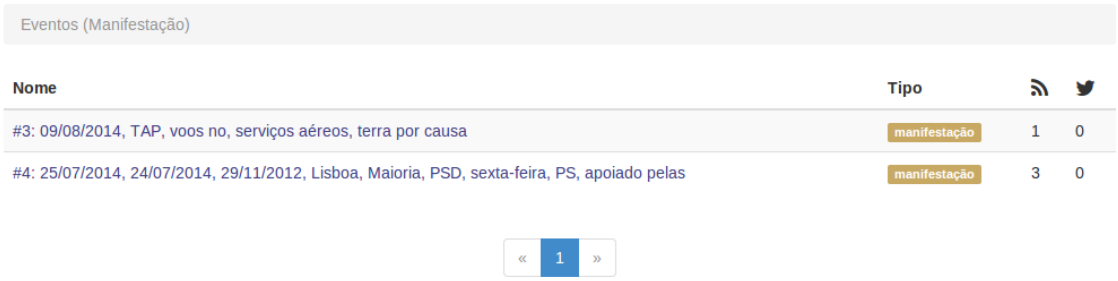


Figura 5.3: *Screenshot* do módulo de *browsing* apenas para eventos do tipo "manifestação"

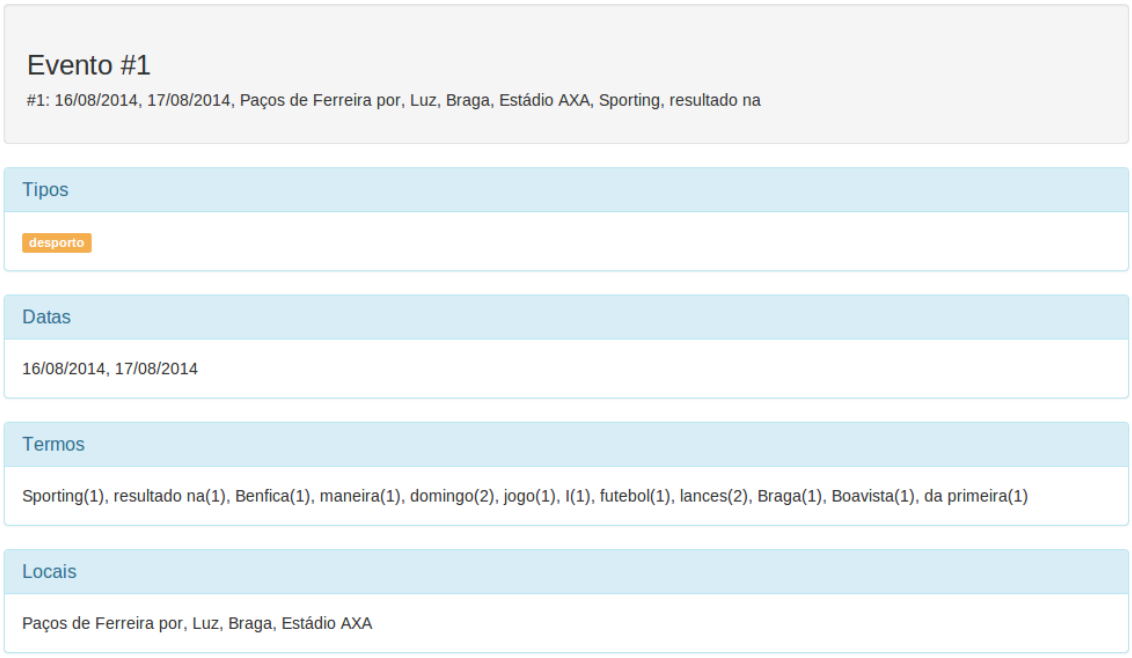


Figura 5.4: *Screenshot* do módulo de *browsing*, com a amostragem dos atributos de um único evento (parte I)

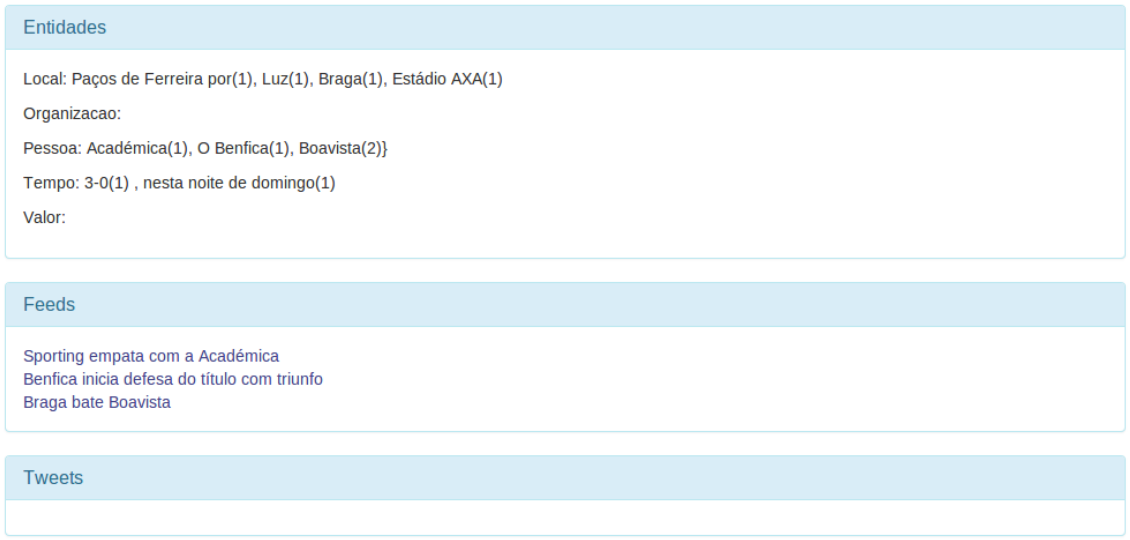


Figura 5.5: *Screenshot* do módulo de *browsing*, com a amostragem dos atributos de um único evento (parte II)

Figura 5.6: *Screenshot* do módulo de recomendação para o *index* do sistema

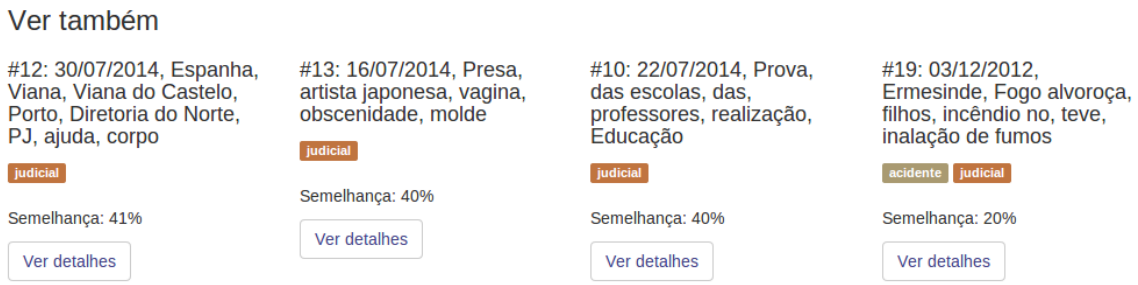


Figura 5.7: *Screenshot* do módulo de recomendação genérico

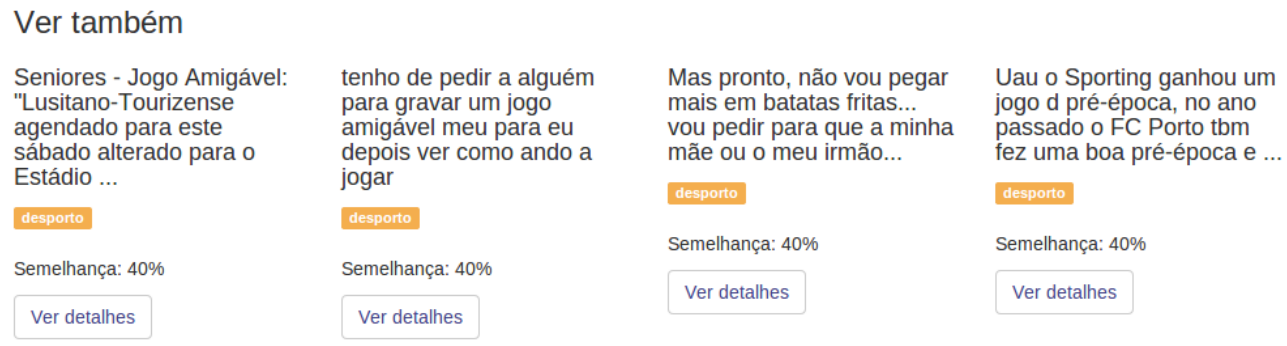


Figura 5.8: *Screenshot* do módulo de recomendação para os *tweets*

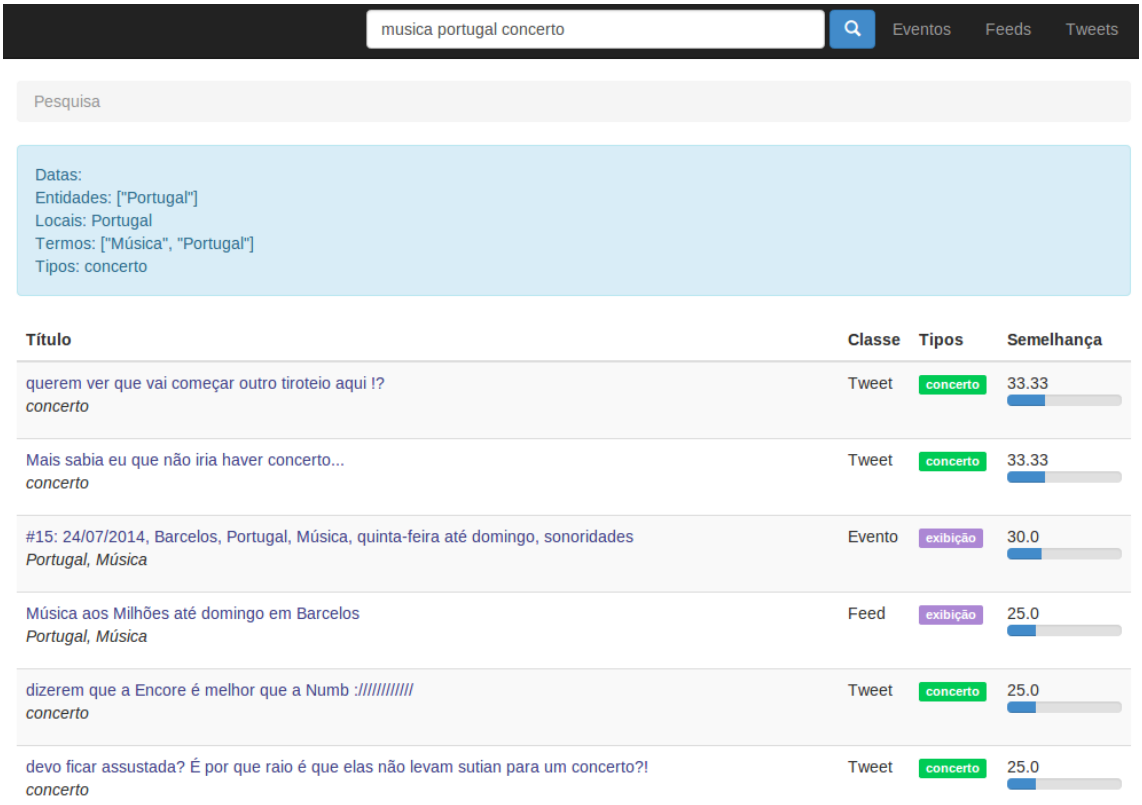


Figura 5.9: *Screenshot* do módulo de pesquisa com a query "musica portugal concerto"

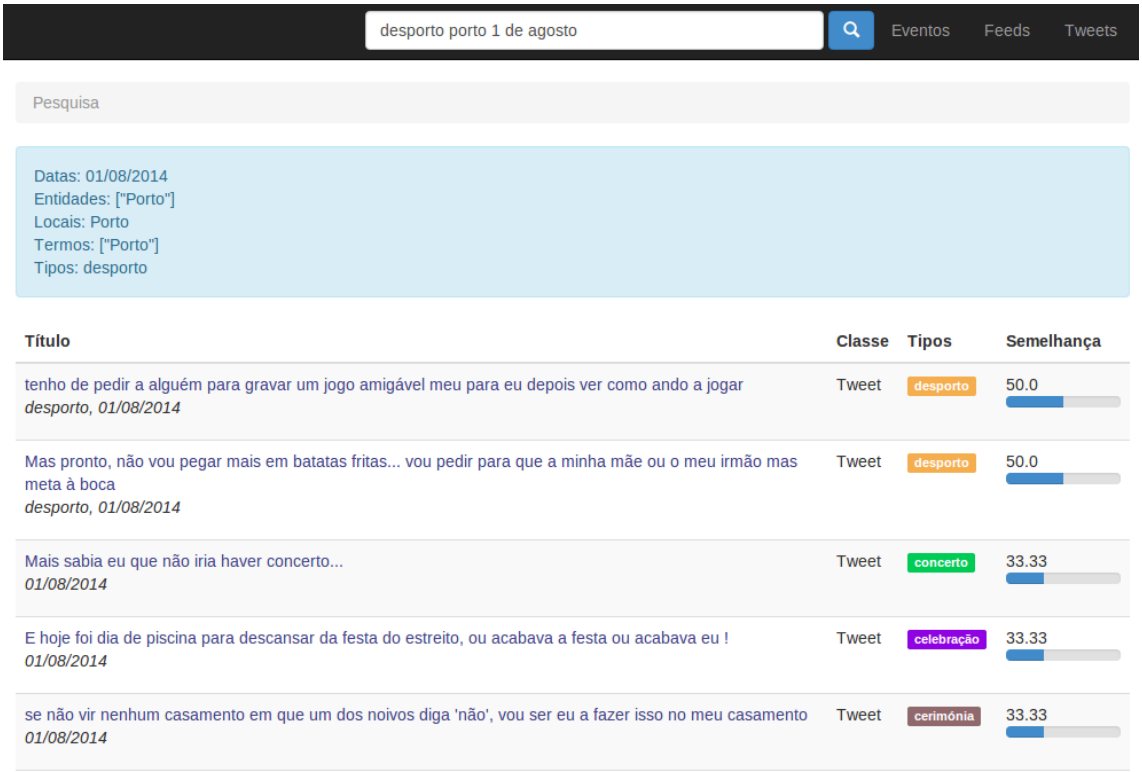


Figura 5.10: *Screenshot* do módulo de pesquisa com a query "desporto porto 1 de agosto"

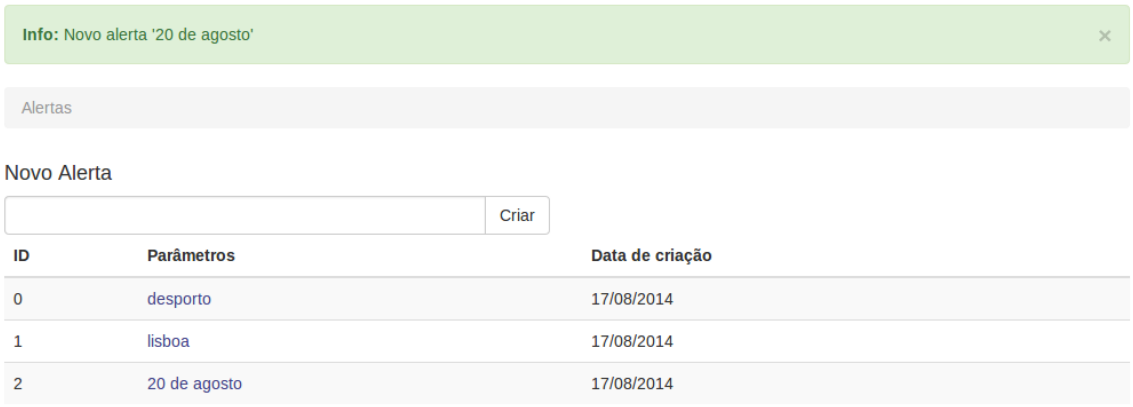


Figura 5.11: *Screenshot* do módulo de alertas

5.5 Módulo de Detecção de Eventos

O módulo de detecção de eventos é o primeiro sector do sistema a receber e tratar a informação. Diz respeito aos requisitos #1, #2, #3 e #4, e tem como principais funções descarregar os *feeds* e os *tweets* sobre eventos portugueses, e seguidamente, extrair todo o tipo de informação segundo os processos definidos pelo *workflow*. Este módulo desperta de hora a hora para fazer algumas operações:

1. verifica se existem registos a mais no servidor, e em caso afirmativo, elimina os mais antigos;
2. descarrega até cerca de 1000 *feeds* portugueses;
3. injecta a lista de *feeds* no *workflow*;
4. transfere a lista de *feeds* já tratados para o módulo de gestão de dados;
5. descarrega até cerca de 1000 *tweets* portugueses;
6. injecta a lista de *tweets* no *workflow*;
7. transfere a lista de *tweets* já tratados para o módulo de gestão de dados;
8. notifica o módulo de alertas;
9. fica em espera durante 1 hora de forma a não descarregar informação repetida, e no final volta ao início do ciclo;

5.6 Módulo de Gestão de Dados

O módulo de gestão de dados trata de toda a comunicação com o *Triple Store*. Todas operações Create Read Update Delete (CRUD) sobre os dados são realizadas aqui, acabando todo o módulo por representar o *model* na arquitectura Model View Controller (MVC) implementada.

Depois de descarregados e extraídos os dados das publicações, cabe a este módulo guardá-los no servidor, criar e actualizar os eventos, e disponibilizar toda a informação nas questões de procura, *browsing* e recomendação.

5.7 Módulo de *Browsing*

Este módulo diz respeito ao requisito de sistema #5. Tem como principal objectivo o acesso a todo o tipo de dados presente no servidor, e apenas requer a navegação pelo interface. Uma vez implementada a arquitectura MVC, foram definidos 3 tipos de controladores: um para os eventos, um para os *feeds* e outros para os *tweets*. Para cada um destes foram implementados os seguintes métodos:

- listagem - lista todos os elementos dessa classe por ordem de crescente de ID's em várias páginas, sendo o limite de registos por página de 25 unidades, de forma a melhorar tanto a procura para o utilizador como a *performance* do método (ver figura 5.2);
- amostragem - selecção de um único elemento e disponibilização numa só página de todos os seus atributos como tipos, datas, locais, entidades e termos. Dependendo da classe do elemento, poderão existir mais alguns atributos (ver figuras 5.4 e 5.5).

No caso do controlador dos eventos, foi implementado um outro método para este módulo. De modo a permitir a filtragem de eventos por tipo, incorporou-se um menu de separadores que realiza esta operação. Este menu dispõe ainda de uma contagem do número de eventos disponíveis até ao momento para cada tipo (ver figura 5.3).

5.8 Módulo de Recomendação

O módulo de recomendação corresponde ao requisito #7. É responsável pela recomendação de conteúdo consoante a actual página. À medida que o utilizador percorre as recorre à amostragem de itens (módulo de *browsing*), são lhe recomendados itens da mesma classe consoante a sua semelhança. De modo a não sobrecarregar a página, são apenas disponibilizados os 4 itens com maior semelhança ao item em visualização (ver figuras 5.7 e 5.8).

Ainda neste módulo, foi inserida recomendação na página principal do interface. Esta escolhe 6 eventos do servidor aleatoriamente e disponibiliza-os para o utilizador, tentando disponibilizar informação do seu interesse sem que este execute algum tipo de procura (ver figura 5.6).

5.9 Módulo de Pesquisa

O requisito #6 diz respeito às funcionalidades do módulo de pesquisa. Foi desenvolvida um tipo de pesquisa semântica onde o utilizador pode inserir qualquer conjunto de palavras a pesquisar pelos atributos de todos os itens disponíveis. Tratando-se de pesquisa, foi também tido em consideração não limitar os resultados ás palavras tal e qual como se encontram no sistema. Assim, a pesquisa implementada:

- é *case insensitive*;
- permite a escrita do caracter "c" ao invés do caracter "ç";
- não é sensível a assentos ortográficos;
- ignora espaçamentos excessivos;
- converte referências temporais para a data absoluta.

Depois de inserido o *input* do utilizador, o programa chama recursivamente uma função que descortina a que tipo de atributo se referem uma ou mais palavras presentes, e em que item ou itens estão disponíveis. Tal como nas entidades e termos, uma vez que toda a *query* pode pertencer a um só atributo, a pesquisa começa por ser feita sobre todas as palavras da *query*. Se nada for encontrado, então começa a ser dividido em *tokens* e o procedimento é chamado recursivamente, até encontrar resultados.

Se o utilizador pretender saber se existiram eventos como manifestações em Lisboa na semana anterior, uma simples *query* como "manifestacao Lisboa semana passada" chega para procurar esses mesmo resultados.

Tal como no módulo *browsing*, são disponibilizados 25 resultados por página por ordem decrescente de semelhança com o *input*, de modo a não sobrecarregar a tarefa do do utilizador nem o próprio sistema (ver figuras 5.9 e 5.10).

5.10 Módulo de Alertas

O módulo de alertas está contemplado no requisito #8. Tal como havia sido descrito, este desempenha a função de comunicar ao utilizador que existem novas informações. Numa abordagem mais simplista, este módulo notifica o sempre que novas publicações forem extraídas, o que gera uma actualização dos eventos disponíveis. Foram também implementados os alertas, onde o utilizador pode definir um conjunto de *tokens*, e caso estes venham a estar presentes nos atributos das publicações futuramente extraídas, o utilizador é novamente notificado pelo interface, actuando como um *trigger* (ver figura 5.11).

Capítulo 6

Experimentação

Neste capítulo será demonstrada toda a experimentação realizada sobre o protótipo, explicada a razão que levou à execução dos vários tipos de testes, comparados os resultados e referenciadas algumas das mudanças que surgiram durante e após este processo. Será explicada toda a recolha de conjunto de dados e as 3 fases que constituiriam a experimentação em si: **Evento *vs* Não Evento** (6.2), **Tipo de Eventos** (6.3) e **Agregação de Eventos** (6.4).

6.1 Dados de Treino e de Teste

Em todo o desenvolvimento, foi constante a necessidade de arranjar dados que contribuíssem de mais variadas formas para o sistema. Várias foram as tentativas de afinação e melhoramento dos dados de treino para que também os resultados fossem melhores. Na tabela 6.1 estão apresentados e descritos todos os conjuntos de dados criados.

Ainda na primeira etapa do processo de classificação, surgiu a necessidade de redefinir toda a ontologia (primeira abordagem descrita nas figuras 4.3, 4.4 e 4.5), visto ser muito vasta e ambígua: se já para um ser humano era complicado classificar alguns eventos nestes moldes, para uma máquina mais complicado seria. Assim, foram redefinidos os tipos de eventos para um total de 20. Foi criado um primeiro conjunto de dados experimental ("DATASET_TYPES"), catalogado manualmente, com cerca de 30 amostras positivas para cada um dos 20 tipos, ou seja, um total de 600 amostras (ainda apenas para os *feeds*). Foi realizada a classificação através de um classificador bayesiano[1] disponível para Ruby.

[1] https://github.com/cardmagic/classifier

Nome	**# Registos**	**Descrição**
DATASET_ALPHA	500	Conjunto de dados que sustentaram a criação da primeira ontologia de tipos de eventos, composto tanto por *feeds* e por *tweets*
DATASET_TYPES	$\simeq$ 600	Conjunto de dados composto por 30 *feeds* positivos para cada um dos 20 tipos de eventos considerados para a ontologia intermédia
DATASET_TYPES_V2	$\simeq$ 600	Conjunto de dados composto por 50 *feeds* positivos para cada um dos 12 tipos de eventos considerados para a ontologia final
DATASET_TYPES_FINAL	$\simeq$ 4500	Conjunto de dados separado em 24 partes, 12 para os dados de treino dos tipos de eventos dos *feeds* e outros 12 para os dados de treino dos tipos de eventos de *tweets*. Cada uma destas partes contém aproximadamente 100 publicações positivas e cerca de 100 publicações negativas
DATASET_EV_NOT_EV	200	Conjunto de dados com 100 publicações referentes a eventos (casos positivos) e 100 publicações não referentes a eventos (casos negativos)
DATASET_EVENT_COMP	150	Conjunto composto por 75 pares de publicações referentes ao mesmo evento (casos positivos) e outros 75 pares referentes a publicações de diferentes eventos (casos negativos), onde foram extraídas várias características para determinar a relevância de cada uma delas
DATASET_EVENT_AGGR	115	Conjunto de dados com 150 *feeds* agregados manualmente em 115 eventos diferentes. Serviu para testar os pesos e o limiar para a função de agregação de eventos

Tabela 6.1: Tabela descritiva dos vários conjuntos de dados criados

Foi importante realizar esta primeira abordagem não para arranjar resultados mas para retirar algumas impressões. Foi entendido que o sistema necessitava de maior robustez, e que para isso era necessário:

- reduzir o número de tipos, visto que alguns deles apresentavam grandes semelhanças e outros eram muito esporádicos;
- aumentar o número de amostras positivas;

Numa segunda fase, foram novamente redefinidos os tipos de eventos e chegou-se aos 12 finais (ver figura 4.6). Foi criado um conjunto de dados de raiz ("DATASET_TYPES_V2") para estar em conformidade com as novas categorias com cerca de 50 amostras de cada tipo, e foi testado novamente o comportamento do classificador. Novas conclusões surgiram depois destes testes. Era necessário:

- incluir amostras negativas, para passar de apenas um classificador a doze classificadores;
- incluir o maior número de exemplos de casos para cada tipo;
- testar vários classificadores com diferentes algoritmos;
- restringir o *corpus* a eventos ocorridos em território português.

Numa última fase de tratamento das questões de classificação, foram então reconstruídos os conjuntos de dados ("DATASET_TYPES_FINAL") que actualmente existem para classificar as publicações. Foram recolhidas e catalogadas manualmente cerca de 100 observações positivas e cerca de 100 observações negativas para cada tipo de evento, tanto para os *feeds* como para os *tweets*, que respeitassem as regras já enumeradas. Assim foram construídos 24 novos conjuntos de dados com cerca de 200 observações cada um. Procurou-se encontrar um número significativo de casos distintos para cada categoria de evento, de forma a melhorar os resultados do classificador. Também foram procuradas notícias com um grande espaço temporal (entre 2005 e 2014).

Também foi necessário criar um conjunto de dados para extrair algumas características importantes a ter em conta na agregação de publicações em eventos. Para isso, foi construído um conjunto de dados composto por 75 pares de publicações referentes ao mesmo evento e outros 75 pares referentes a publicações de diferentes eventos. Para cada par eram extraídas características como "mesma data", "mesmo local", "mesmo tipo", "data do evento contém data da publicação", "data da publicação contém data do evento", "local do evento

contém local da publicação", "local da publicação contém local do evento" cujos os valores poderiam ser "sim", "não" ou "talvez". Existem também características quantitativas como "entidades em comum" e "termos em comum".

Ao longo de toda da experimentação foi ainda construído mais um conjunto de dados para agregação de eventos. Para este caso, arranjaram-se 150 *feeds* de notícias, e foram manualmente agrupados em eventos ("DATASET_EVENT_AGGR"). Este conjunto de dados serviu para avaliar a qualidade das várias funções de agregação de eventos.

6.2 Evento *vs* Não Evento

Foram realizados testes para avaliação da qualidade da classificação das publicações referentes a eventos. Exemplos com crónicas ou simples comentários podem vir a ser extraídos e não conter qualquer informação relativa a eventos portugueses, pelo que a avaliação foi realizada no sentido de apurar a viabilidade desta primeira classificação. Uma vez que o Ruby se trata de uma linguagem de *scripting*, foi construído um *script* que corre e gera estes resultados automaticamente.

	C4.5			**Decision Tree**			**Maximum Entropy**			**Naive Bayes**			**Winnow**		
Classe	exa	des	err	exa	des	err	exa	des	err	exa	des	err	exa	des	err
feeds	0.6350	0.1163	0.0368	0.7250	0.0873	0.0276	0.8100	0.0700	0.0221	0.8200	0.0748	0.0237	0.6600	0.0970	0.0307
tweets	0.6800	0.1054	0.0333	0.7150	0.1074	0.0339	0.7950	0.0879	0.0278	0.7850	0.0808	0.0255	0.6100	0.0735	0.0232
Média	0.6575	0.1108	0.0350	0.7200	0.0973	0.0308	**0.8025**	0.0789	0.0250	**0.8025**	0.0778	0.0246	0.6350	0.0852	0.0269

Tabela 6.2: Tabela comparativa dos vários algoritmos para a classificação "evento *vs* não evento", utilizando o Mallet. A abreviatura "exa" diz respeito à exactidão; a abreviatura "des" diz respeito ao desvio padrão; a abreviatura "err" diz respeito ao erro médio.

O Mallet foi a ferramenta adoptada para a realização destes testes, uma vez que contém vários algoritmos implementados e é de fácil utilização. Os algoritmos escolhidos para criar os diferentes modelos de treino foram o *C4.5*, *Decision Tree*, *Maximum Entropy*, *Naive Bayes* e *Winnow*. Como método de validação foi utilizado o algoritmo *10-fold Cross Validation*: este divide todo o conjunto de dados em 10 segmentos e utiliza 9 dos 10 segmentos para treino e 1 deles para teste, e executa este procedimento 10 vezes variando sempre o segmento de treino de forma a percorrer todos eles. No final é feita a média da exactidão[2], do desvio padrão e do erro, para os 10 testes realizados.

Como é possível ver na tabela 6.2, os dois algoritmos que obtém melhor resultado são o *Naive Bayes* e o *Maximum Entropy*, com um valor médio igual a 80%. Esta avaliação foi realizada com um pequeno conjunto de dados de eventos e não eventos, devidamente classificados, com cerca de 100 amostras para cada caso (DATASET_EV_NOT_EV).

6.3 Tipos de Eventos

Foram também realizados testes para avaliar a qualidade da classificação dos vários tipos de eventos presentes nas publicações. Tornou-se importante discriminar a avaliação para cada tipo como para cada publicação, o que resultou em 24 avaliações diferentes. À semelhança da experimentação realizada para o caso apresentado em 6.2, foi construído um *script* que faz correr todos estes 24 testes no Mallet, utilizando o método de *10-fold Cross Validation*, para todos os diferentes algoritmos, e apresenta os resultados na forma de um Comma Separated Values (CSV).

[2] O termo em inglês é *accuracy* e é calculada através da soma dos casos positivos da população dividida por todos os elementos da população.

	C4.5			Decision Tree			Maximum Entropy			Naive Bayes			Winnow		
Classe	exa	des	err	exa	des	err	exa	des	err	exa	des	err	exa	des	err
Acidente	0.7523	0.1075	0.0340	0.8523	0.0957	0.0303	0.9092	0.0575	0.0182	0.9536	0.0575	0.0182	0.7016	0.1242	0.0393
Celebração	0.7603	0.1059	0.0335	0.6945	0.1156	0.0365	0.7908	0.1072	0.0339	0.8018	0.0638	0.0202	0.6897	0.1610	0.0509
Cerimónia	0.6776	0.1303	0.0412	0.8699	0.0685	0.0217	0.8875	0.0556	0.0176	0.9283	0.0448	0.0142	0.4989	0.1257	0.0397
Concerto	0.7055	0.1591	0.0503	0.8188	0.0961	0.0304	0.8316	0.0766	0.0242	0.8743	0.0674	0.0213	0.7195	0.0674	0.0213
Desporto	0.8581	0.0750	0.0237	0.8096	0.0708	0.0224	0.8890	0.0550	0.0174	0.8945	0.0885	0.0280	0.8022	0.1215	0.0384
Encontro	0.8558	0.0871	0.0275	0.8617	0.0758	0.0240	0.8289	0.0670	0.0212	0.8117	0.1005	0.0318	0.6582	0.1011	0.0320
Espectáculo	0.8893	0.0660	0.0209	0.9140	0.0689	0.0218	0.9199	0.0927	0.0293	0.8471	0.0864	0.0273	0.8346	0.1308	0.0414
Exibição	0.6423	0.1190	0.0376	0.7471	0.1161	0.0367	0.8217	0.0967	0.0306	0.8408	0.0801	0.0253	0.5544	0.1303	0.0412
Judicial	0.6324	0.1139	0.0360	0.7585	0.1094	0.0346	0.8063	0.0919	0.0291	0.8662	0.0983	0.0311	0.6386	0.1035	0.0327
Manifestação	0.6813	0.1324	0.0419	0.8625	0.0612	0.0194	0.8938	0.0688	0.0217	0.8875	0.0375	0.0119	0.6250	0.1398	0.0442
Natural	0.8500	0.0500	0.0158	0.7813	0.0895	0.0283	0.9500	0.0545	0.0172	0.9500	0.0545	0.0172	0.7563	0.1233	0.0390
Política	0.8875	0.1146	0.0362	0.7313	0.1120	0.0354	0.9063	0.0803	0.0254	0.8813	0.0763	0.0241	0.7000	0.1179	0.0373
Média	0.7660	0.1051	0.0332	0.8084	0.0900	0.0284	0.8696	0.0753	0.0238	**0.8781**	0.0713	0.0225	0.6816	0.1205	0.0381

Tabela 6.3: Tabela comparativa dos vários algoritmos para a classificação dos tipos de eventos para os *feeds*, utilizando o Mallet. A abreviatura "exa" diz respeito à exactidão; a abreviatura "des" diz respeito ao desvio padrão; a abreviatura "err" diz respeito ao erro médio.

	C4.5			Decision Tree			Maximum Entropy			Naive Bayes			Winnow		
Classe	exa	des	err	exa	des	err	exa	des	err	exa	des	err	exa	des	err
Acidente	0.8750	0.1008	0.0319	0.7688	0.1048	0.0331	0.8438	0.0699	0.0221	0.8313	0.0688	0.0217	0.7063	0.1084	0.0343
Celebração	0.5375	0.1090	0.0345	0.7375	0.1275	0.0403	0.7813	0.0895	0.0283	0.7250	0.0935	0.0296	0.6438	0.1154	0.0365
Cerimónia	0.5000	0.1152	0.0364	0.7125	0.1016	0.0321	0.6625	0.1090	0.0345	0.6250	0.1083	0.0342	0.5250	0.0976	0.0309
Concerto	0.9438	0.0519	0.0164	0.8125	0.0839	0.0265	0.9063	0.1127	0.0356	0.8750	0.0685	0.0217	0.6500	0.1225	0.0387
Desporto	0.8750	0.0901	0.0285	0.7250	0.1328	0.0420	0.8850	0.0550	0.0174	0.8800	0.0954	0.0302	0.6400	0.1261	0.0399
Encontro	0.8438	0.0978	0.0309	0.7500	0.0884	0.0280	0.8125	0.1046	0.0331	0.7813	0.0576	0.0182	0.6250	0.0791	0.0250
Espectáculo	0.4313	0.1491	0.0471	0.8063	0.0438	0.0138	0.7688	0.0841	0.0266	0.7625	0.0545	0.0172	0.6500	0.0893	0.0282
Exibição	0.8625	0.1146	0.0362	0.7500	0.0839	0.0265	0.8125	0.1281	0.0405	0.8188	0.1354	0.0428	0.5188	0.1399	0.0442
Judicial	0.5313	0.1751	0.5554	0.6625	0.0800	0.0253	0.8063	0.0813	0.0257	0.7813	0.0978	0.0309	0.5813	0.1048	0.0331
Manifestação	0.6938	0.1264	0.0400	0.9063	0.0640	0.0203	0.8688	0.0859	0.0272	0.8313	0.0841	0.0266	0.7563	0.0946	0.0299
Natural	0.5250	0.1658	0.0524	0.8000	0.1111	0.0351	0.8375	0.0848	0.0268	0.7563	0.1063	0.0336	0.7313	0.1048	0.0331
Política	0.4688	0.1612	0.0510	0.8813	0.0653	0.0206	0.8938	0.0886	0.0280	0.8688	0.0813	0.0257	0.7250	0.0800	0.0253
Média	0.6740	0.1214	0.0384	0.7760	0.0906	0.0286	**0.8232**	0.0911	0.0288	0.7947	0.0876	0.0277	0.6460	0.1052	0.0333

Tabela 6.4: Tabela comparativa dos vários algoritmos para a classificação dos tipos de eventos para os *tweets*, utilizando o Mallet. A abreviatura "exa" diz respeito à exactidão; a abreviatura "des" diz respeito ao desvio padrão; a abreviatura "err" diz respeito ao erro médio.

Estes testes foram aplicados sobre os dados que são utilizados para treinar os vários classificadores. Analisando as tabelas (6.3 e 6.4), é possível perceber que os algoritmos com melhores resultados são novamente *Naive Bayes* para o caso dos *feeds* e o *Maximum Entropy* para os *tweets*. Como primeiramente só se estava a utilizar o *Maximum Entropy*, depois de concluídos estes testes, passou-se a utilizar o algoritmo *Naive Bayes* apenas para a classificação dos tipos dos *feeds*. A exactidão média para dos *feeds* ronda os 88% enquanto que para os *tweets* ronda os 82%. Esta diferença compreende-se uma vez que as notícias detêm informação em maior número e correctamente estruturada, enquanto que os *tweets* estão limitados aos 140 caracteres e não possuem qualquer tipo de tratamento de linguagem.

6.4 Agregação de Eventos

Nesta secção é explicada a abordagem da agregação de eventos, ou seja, a função que junta as várias publicações num só evento dependendo do seu conteúdo. Tornou-se na fase da experimentação mais exigente visto ter que ser afinada várias vezes de forma a produzir os melhores resultados possíveis.

Numa primeira abordagem, a função de agregação de publicações a um só evento exigia apenas que duas publicações tivessem pelo menos um tipo de evento em comum, bem como pelo menos 1 outro atributo em comum (data/local/termo/entidade). Esta revelou resultados satisfatórios mas a uma determinada altura, quando um evento detinha um número elevado de publicações, começava a ter o efeito "buraco negro", já que detinha muitos atributos e tornava-se bastante provável encontrar publicações parecidas mas não referentes ao mesmo evento.

Numa segunda abordagem, para que duas publicações pertencessem ao mesmo evento, era exigido pelo menos um tipo de evento em comum, pelo menos uma data em comum e pelo menos um local em comum. Como nem todas as publicações contêm data e muito menos local, esta metodologia limitava o efeito agregador, acabando praticamente todos os eventos por serem constituídos pela publicação que os originou.

Foi iniciada uma terceira abordagem que consistia na utilização da função de semelhança 5.3 entre um evento já existente e uma publicação nova, para determinar se esta pertencia ou não a esse evento. Consoante o resultado e um determinado limiar, a agregação era ou não concedida. Para este caso, foram utilizados os pesos de 0.4 para o atributo "tipos de eventos" e 0.15 para os 4 restantes (datas, locais, entidades e termos). Foram utilizados

estes valores uma vez que, ao longo do processo, notou-se a importância do atributo "tipos de evento" relativamente a todos os outros. Convém referir que só a partir desta fase é que se começaram a realizar testes. Nesta etapa, foi apenas testado o valor ideal de limiar para estes pesos como é possível de verificar na figura 6.1, através do conjunto de dados de agregação de eventos já construído.

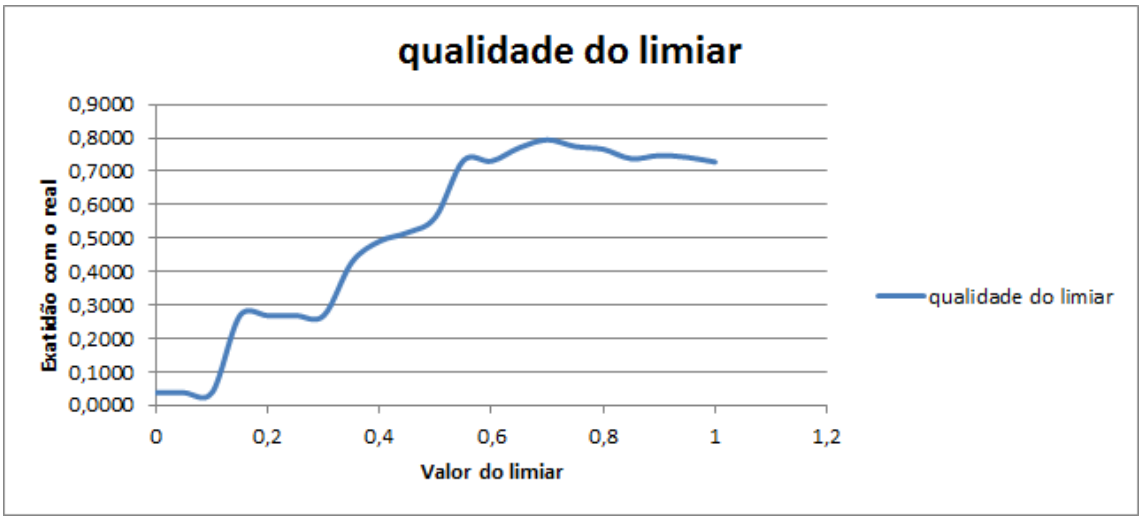


Figura 6.1: Gráfico da qualidade dos vários valores de limiar de semelhança com pesos de valor 0.15 para datas, locais, entidades e termos, e 0.4 para os tipos de eventos.

A quarta e última abordagem consistiu na realização de um algoritmo genético para determinar os valores ideais para os 5 pesos. Um algoritmo genético consiste numa técnica de optimização que procura a solução óptima através de uma evolução dos indivíduos. À semelhança da Natureza, os indivíduos contêm informação genética variável que os distingue uns dos outros, geralmente chamada de cromossomas que são representadas em cadeias de caracteres de 0's e 1's. A população inicial é gerada de forma aleatória. A existência de diferentes indivíduos é o resultado ou do cruzamento de indivíduos ou da mutação que apesar de muito pouco provável, pode ocorrer nos mesmos. A qualidade do indivíduo é o resultado da função de *fitness*, e é aqui que são encontrados os melhores indivíduos de cada geração.

- os cromossomas eram constituídos por 20 bits, 4 destinados a cada um dos pesos, o que possibilitou 16 valores distintos para cada um dos pesos, sendo o mínimo o valor 0 e o máximo 0.9375;
- o rácio de *crossover* foi estipulado nos 0.65, um valor usado normalmente para estas abordagens;
- o rácio de mutação foi fixado nos 0.001, o que significa que existia 0,1% de probabilidade de cada indivíduo sofrer uma mutação;

- a função de *fitness* consistia no cálculo da exatidão para com os eventos presentes no conjunto de dados de agregação de eventos ("DATASET_EVENT_AGGR");
- o limiar de semelhança varia entre 0 a 1, com um passo de 0.05, e o algoritmo genético foi corrido para cada um dos valores que o limiar podia assumir.
- em cada geração é apenas guardado o melhor de todos os indivíduos para a próxima;

Em primeira análise, o algoritmo genético foi testado com 20 indivíduos e 10 gerações para tentar encontrar erros. Como esperado, os resultados não foram bem sucedidos o que se deve à escassez do número de gerações e de indivíduos introduzido. Posto isto, o número de indivíduos foi alterado para 100 bem como o número de gerações. De relembrar que também aqui o algoritmo genético foi corrido cerca de 20 vezes (uma para cada valor do limiar, sendo que $20 \times 0.05 = 1$).

Tratando-se de um sistema pesado, uma vez que para cada indivíduo de cada geração, eram agregadas 150 publicações, com os pesos da função de semelhança definidos pelo cromossoma do indivíduo, estima-se que para cada um dos valores do limiar, o algoritmo tenha demorado mais de 16 horas, pelo que foram necessários vários dias até à determinação da melhor solução. Estes testes foram deixados a correr no servidor *Mustang* também utilizado para alojar tanto o *triple-store* como o próprio site.

De seguida são discriminados os valores óptimos para os pesos e limiar consoante a função de semelhança implementada, que resultaram numa valor de exactidão com o real de 0.83:

- $w_{tipos} = 0.56$;
- $w_{datas} = 0.19$;
- $w_{locais} = 0.13$;
- $w_{entidades} = 0.06$;
- $w_{termos} = 0.06$;
- $limiar = 0.7$;

Depois de realizada esta última fase da experimentação, os pesos e o valor do limiar foram actualizados com vista a optimizar os resultados do site, uma vez que ele se encontra sempre disponível. Os valores possíveis para cada peso oscilavam entre 0 e 0.9375 como já foi explicado. Isto acontecia porque existiam 16 valores diferentes a começar no 0, e

também não faz sentido existir um peso com valor muito próximo e 1.0 já que anula a informação presente nos outros parâmetros.

Pelo resultado obtido, percebe-se claramente que o resultado da classificação do tipo de eventos é bastante significativo para a agregação de eventos, o que nos diz que duas publicações do mesmo tipo são provavelmente do mesmo evento, e duas publicações de diferentes tipos, provavelmente dizem respeito a eventos diferentes. Logo de seguida, os atributos mais discriminantes são a data e o local, já que isolam as publicações no espaço e no tempo, permitindo perceber as similaridades com outras com as mesmas características. Publicações com a data diferente (para este caso são consideradas datas diferentes, datas separadas por um intervalo temporal superior a 48 horas) têm uma boa probabilidade de se referirem a diferentes eventos, uma vez que as publicações sobre determinado acontecimento, por norma, se concentram todas num curto espaço de tempo. Como os termos e as entidades extraídas nem sempre possuem os melhores resultados, e como para estes casos em que o texto não é muito longo o número de termos ou entidades é escasso, os pesos encontrados são de valor reduzido.

O valor do limiar encontrado assegura que o tipo de eventos e publicações deva encontrase em total conformidade já que este constitui 56% dos atributos. Para agregar uma publicação com um evento onde foram classificações dos tipos de eventos são diferentes (pelo menos um distinto), é necessário que todos os outros atributos sejam iguais para que este os considere do mesmo evento.

Comparando os resultados com os da 6.1, é possível perceber que o limiar desceu em função da variância dos pesos. No entanto a exactidão conseguiu passar a margem dos 80% quando testado com o conjunto de dados para agregação de eventos ("DATASET_EVENT_AGGR");

6.5 Análise dos Resultados Obtidos

O sistema implementado conseguiu obter resultados satisfatórios, já que nas várias etapas da experimentação, a taxa de sucesso situa-se na ordem dos 80%. Quando executados no próprio sistema em tempo real, os vários tipos de classificação descem a sua taxa de sucesso residualmente. Isto acontece porque a quantidade de dados em análise é muito superior, o espaço temporal das notícias é mais reduzido o que pode dar azo a erros de análise, uma vez que as datas são consideradas iguais quando o espaço de tempo é diminuto (menor que 48 horas), até porque nem sempre as publicações tinham data ou nem sempre era possível traduzi-la, acabando por ser assumida a data da publicação. Neste caso as datas vão ser

bastante coincidentes já que o sistema se encontra a receber novas publicações de hora a hora.

Os resultados obtidos através da classificação de tipos de eventos demonstram também uma satisfatória taxa de sucesso. Estes ainda têm espaço para melhoria se for aumentado o número de dados de treino para cada uma das categorias, e se forem combatidas algumas ambiguidades. Isto torna-se mais significativo para os *tweets* que sofrem de uma maior escassez de informação quando comparados com os *feeds*, e como também já foi descrito, muita dessa informação não se encontra devidamente empregue, o que explica a diferença para com os resultados dos *feeds*.

Tal como aconteceu para o algoritmo genético, a comparação de publicações com os eventos já introduzidos no sistema torna-se bastante demorada quando o número de eventos se torna elevado (superior a 250). A partir daqui tanto os resultados como a própria *performance* do sistema se encontrava baixa, o que levou à necessidade de apagar eventos antigos no próprio sistema.

O efeito "buraco negro" ocorre quando um determinado evento contém várias publicações, e consequentemente muitos dados, e ainda não foi extinto por completo. Como a probabilidade de encontrar semelhança é maior, é provável que para as funções implementadas, este seja considerado semelhante com novas publicações a agregar, podendo "sugar" todas estas novas publicações. Daqui resultam eventos com agregação errada e cuja solução passa apenas por apagá-los já que estão errados e não permitem que os verdadeiros eventos consigam agregar os dados correctos.

Capítulo 7

Conclusão

Neste capítulo será realizado um apanhado geral de todo o desenvolvimento do projecto. Em primeiro lugar serão apresentados os vários planeamentos (inicial, intermédio e final) e explicadas algumas das causas das suas alterações. De seguida, serão apresentadas as considerações finais onde constam as contribuições do trabalho, algumas das limitações encontradas e sugestões para o trabalho futuro.

7.1 Planeamento

O objectivo inicial proposto para o projecto era detecção de situações anómalas na rede móvel com recurso a dados provenientes de redes sociais. Para isso, fariam parte tanto dados da rede móvel que seriam analisados para perceber o estado da mesma, em localizações distintas. Conseguida a detecção dessas ocorrências, seria necessário adquirir, analisar e cruzar a informação oriunda do Twitter e do Facebook e, caso fosse necessário, do Foursquare[1], de modo a compreender as causas da anormalidade dos acontecimentos, através do relato directo dos utilizadores. O planeamento original consta na Figura 7.1, onde é possível visualizar as várias fases que compunham a ideia inicial. Em paralelo, foi dado início à produção do relatório do projecto.

Com a adaptação do foco do projecto, a componente da rede móvel foi descartada, dando especial evidência aos eventos. Como tal, as fontes de dados passaram a ser o Twitter e fontes de notícias portuguesas, através do serviço de Really Simple Syndication (RSS). Estes servem de base para a análise e percepção de os vários tipos de eventos a decorrer

[1] https://foursquare.com

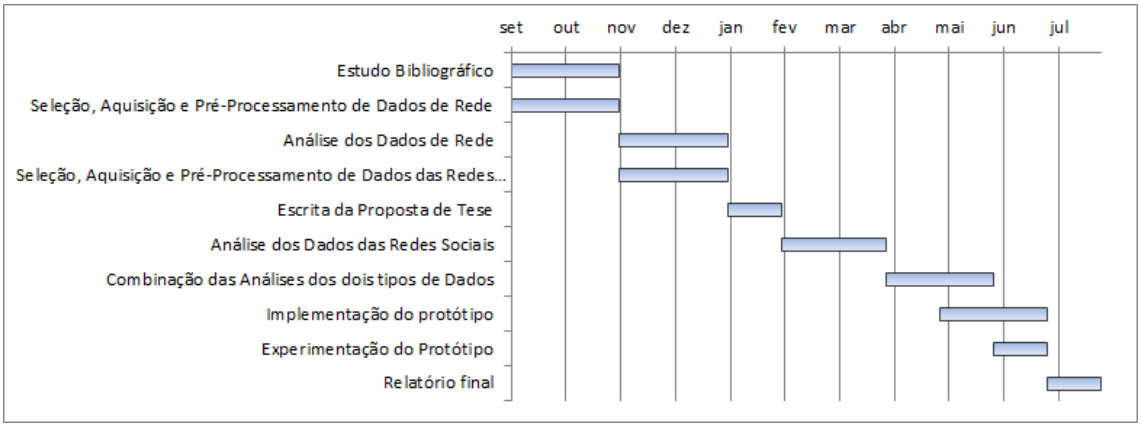


Figura 7.1: Diagram de Gantt com o planeamento inicial.

em tempo-real. O sistema consegue realizar a aquisição, tratamento e criação de eventos de forma autónoma e permanente, sem qualquer interferência do utilizador. Na Figura 7.2 está presente o actual planeamento intermédio de todo o projecto, sendo que a fase após a defesa intermédia era "Pré-Processamento e Análise dos Dados dos *Feeds* de Notícias".

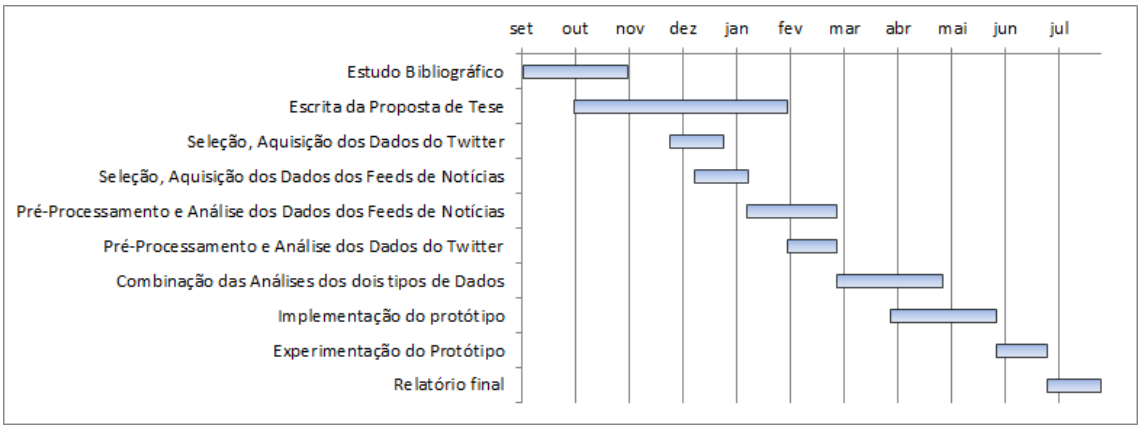


Figura 7.2: Diagrama de Gantt com o planeamento intermédio.

Do planeamento intermédio para o planeamento final também surgiram alterações: desde já foi dedicado bem mais tempo que o previsto até então à fase "Pré-Processamento e Análise dos Dados dos *Feeds* de Notícias". Isto aconteceu porque foram várias as abordagens de classificação, as reformulações da ontologia e a construção dos vários conjuntos de dados que possibilitam a classificação. Esta fase foi bem mais longa do que a homónima para os *tweets* uma vez que as adaptações e as conclusões foram todas tiradas nesta fase.

O início da implementação também surgiu logo após a data da defesa intermédia o que permitiu ir adaptando o sistema consoante as conclusões que eram retiradas da análise dos *feeds*. A partir do momento que se iniciou o "Pré-Processamento e Análise dos Dados dos *Tweets*", também se pôde dar início à seguinte tarefa, cujo o nome é "Combinação e Análise dos dois tipos de Dados", que consistia na afinação das funções de semelhança e

agregação de eventos.

Com o iniciar da experimentação, ainda foram afinados alguns detalhes no protótipo, o que justifica a implementação terminar após o início desta fase. De forma a obter bons resultados nas várias abordagens da experimentação, foi decidido agendar a data de entrega do relatório para setembro. O plano de execução culmina com a conclusão do relatório final.

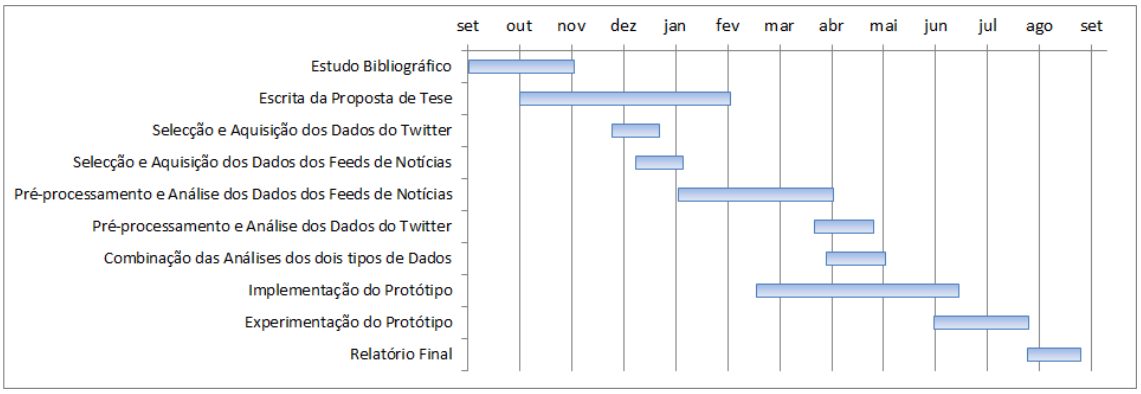


Figura 7.3: Diagrama de Gantt com o planeamento final.

Ao longo do desenvolvimento, foram realizadas reuniões semanais com o orientador para que fosse possível avaliar o estado das várias fases e se pudesse planear a abordagem a tomar para as fases seguintes. Foi utilizada uma abordagem sequencial uma vez que as fases iniciais serviram de base para o trabalho futuro. O objectivo principal foi a criação do sistema de agregação de eventos, e concluída a sua construção, a prioridade passou por afinar a sua taxa de sucesso para as várias categorias encontradas.

7.2 Considerações finais

Conhecido o problema de adquirir conhecimento a partir de texto escrito por agentes humanos nos mais variados tipos de registo, esta dissertação procurou encontrar uma solução para a agregação de informações relativas a eventos a partir de várias fontes de informação online.

Foi constituído o problema, delineados os objectivos e as directrizes para o desenvolvimento do projecto. Através da conjugação das áreas de Processamento de Linguagem Natural (PLN), Web Semântica (WS) e IA foi possível extrair e agregar a informação (disponível no Twitter e em jornais de notícias portuguesas) referente a eventos portugueses.

Várias foram as limitações que surgiram no desenvolvimento do trabalho: desde já a escolha do tratamento exclusivo da língua portuguesa o que implicou que algumas ferramentas tivessem de ser adaptadas; a inexistência de dados de treino para as categorias de eventos portugueses provenientes das fontes referidas, o que obrigou à procura e categorização desta informação de forma manual; o défice de informação presente em algumas publicações exigiu a idealização de novas abordagens tanto de comparação como de agregação de conteúdo.

Olhando para os resultados da experimentação, e uma vez que muitos deles se situam na ordem dos 80%, é possível concluir que os objectivos foram atingidos, já que o sistema consegue identificar eventos portugueses em notícias e *tweets*, tem a capacidade de extrair a informação presente nas publicações analisadas e agrupá-la em blocos de semelhança que constituem os eventos interpretados. Em complemento, ainda detém métodos de WS como a pesquisa e a recomendação semântica que facultam a interacção com os dados de forma mais prática.

É possível enumerar algumas contribuições resultantes deste projecto:

- a ontologia de eventos construída;
- os vários conjuntos de dados de classificação de tipos de eventos nos 2 tipos de publicações;
- o interpretador de datas para a língua portuguesa;
- todo o sistema de classificação e agregação de conteúdos;
- o interface desenvolvido que permite aceder aos dados;

Ao nível de trabalho futuro, serão procurados mais dados para classificação de conteúdo de forma a melhorar os resultados. Outras formas de agregação e exploração de mais características significativas podem ser investigadas, de forma a permitirem tanto melhorar a taxa de sucesso como acrescentar mais informação ao conteúdo trabalhado. Podem também ser exploradas aplicações práticas deste tipo de informação, nomeadamente o cruzamento da informação dos eventos já obtida com informação sobre anomalias em redes móveis; interoperabilidade com serviços online de forma a acrescentar valor aos mesmos; criação de aplicações móveis de acesso aos dados do sistema em tempo real e expandir o sistema para outras línguas existentes.

Bibliografia

Andersen, P. M., Hayes, P. J., Huettner, A. K., Schmandt, L. M., Nirenburg, I. B., and Weinstein, S. P. (1992). Automatic extraction of facts from press releases to generate news stories. In *Proceedings of the Third Conference on Applied Natural Language Processing*, ANLC '92, pages 170–177, Stroudsburg, PA, USA. Association for Computational Linguistics.

Becker, H., Naaman, M., and Gravano, L. (2010). Learning similarity metrics for event identification in social media. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 291–300, New York, NY, USA. ACM.

Becker, H., Naaman, M., and Gravano, L. (2011). Beyond trending topics: Real-world event identification on twitter. In *Fifth International AAAI Conference on Weblogs and Social Media*.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):34–43.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Carbonell, J. G., Michalski, R. S., and Mitchell, T. M. (1984). An overview of machine learning. In Michalski, R. S., Carbonell, J. G., and Mitchell, T. M., editors, *Machine Learning: An Artificial Intelligence Approach*, pages 3–23. Springer, Berlin, Heidelberg.

Chua, F. C. T. and Asur, S. (2013). Automatic summarization of events from social media. In *ICWSM*.

Currier, S. (2008). Metadata for learning resources: An update on standards activity for 2008. (Ariadne Issue 55).

Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification (2Nd Edition)*. Wiley-Interscience.

Fillmore, C. J. (1982). *Frame semantics*, pages 111–137. Hanshin Publishing Co., Seoul, South Korea.

Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., and Dredze, M. (2010). Annotating named entities in twitter data with crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 80–88, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220.

Jurafsky, D. and Martin, J. H. (2008). *Speech and Language Processing (2nd Edition) (Prentice Hall Series in Artificial Intelligence)*. Prentice Hall, 2 edition.

Li, C., Weng, J., He, Q., Yao, Y., Datta, A., Sun, A., and Lee, B.-S. (2012). Twiner: Named entity recognition in targeted twitter stream. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 721–730, New York, NY, USA. ACM.

Ling, X. and Weld, D. S. (2010). Temporal information extraction. In Fox, M. and Poole, D., editors, *AAAI*. AAAI Press.

Locke, B. and Martin, J. (2009). Named entity recognition: Adapting to microblogging. *University of Colorado*.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Noy, N. F. and Mcguinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology. Technical report.

of Defence (Navy), G. B. M. (1995). *Admiralty Manual of Seamanship: 1995 Edition*. Serie Bestuursrecht. H.M. Stationery Office.

Oliveira, H. G. and Gomes, P. (2010). Onto.pt: Automatic construction of a lexical ontology for portuguese. In Ågotnes, T., editor, *STAIRS*, volume 222 of *Frontiers in Artificial Intelligence and Applications*, pages 199–211. IOS Press.

Poibeau, T. and Kosseim, L. (2001). Proper name extraction from non-journalistic texts. In *In Computational Linguistics in the Netherlands*, pages 144–157.

Rheingold, H. (2000). *The virtual community : homesteading on the electronic frontier*. MIT Press, Cambridge, Mass.

Ritter, A., Mausam, Etzioni, O., and Clark, S. (2012). Open domain event extraction from twitter. In 0001, Q. Y., Agarwal, D., and Pei, J., editors, *KDD*, pages 1104–1112. ACM.

Russell, S. and Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition.

Sá, J. P. M. d. (2001). *Pattern recognition : concepts, methods and applications*. Springer, Berlin, Heidelberg, New York.

Smullyan, R. (1995). *First-order logic*. Dover Publications.

Szabo, Z. G. (2004). Review: The compositionality papers. *Mind*, 113(450):340–344.

Tan, P.-N., Steinbach, M., and Kumar, V. (2005). *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Tanev, H., Ehrmann, M., Piskorski, J., and Zavarella, V. (2012). Enhancing event descriptions through twitter mining. In Breslin, J. G., Ellison, N. B., Shanahan, J. G., and Tufekci, Z., editors, *ICWSM*. The AAAI Press.

Wang, X., Gerber, M. S., and Brown, D. E. (2012). Automatic crime prediction using events extracted from twitter posts. In *Proceedings of the 5th International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction*, SBP'12, pages 231–238, Berlin, Heidelberg. Springer-Verlag.

Wang, Y., Xie, L., and Sundaram, H. (2011). Social event detection with clustering and filtering. In Larson, M., Rae, A., Demarty, C.-H., Kofler, C., Metze, F., Troncy, R., Mezaris, V., and Jones, G. J. F., editors, *MediaEval*, volume 807 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Zhang, R., Li, W., Gao, D., and You, O. (2013). Automatic twitter topic summarization with speech acts. *IEEE Transactions on Audio, Speech & Language Processing*, 21(3):649–658.