Mestrado em Engenharia Informática

Dissertação

Relatório Final

Extração de Informação Semântica de Conteúdo da Web 2.0

Ana Rita Bento Carvalheira

arbc@student.dei.uc.pt

Orientador:

Paulo Jorge de Sousa Gomes

pgomes@dei.uc.pt

Data: 1 de Julho de 2014

FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Agradecimentos

Gostaria de começar por agradecer ao Professor Paulo Gomes pelo profissionalismo e apoio incondicional, pela sincera amizade e a total disponibilidade demonstrada ao longo do ano. O seu apoio, não só foi determinante para a elaboração desta tese, como me motivou sempre a querer saber mais e ter vontade de fazer melhor.

À minha Avó Maria e Avô Francisco, por sempre estarem presentes quando eu precisei, pelo carinho e afeto, bem como todo o esforço que fizeram para que nunca me faltasse nada. Espero um dia poder retribuir de alguma forma tudo aquilo que fizeram por mim.

Aos meus Pais, pelos ensinamentos e valores transmitidos, por tudo o que me proporcionaram e por toda a disponibilidade e dedicação que, constantemente, me oferecem. Tudo aquilo que sou, devo-o a vocês.

Ao David agradeço toda a ajuda e compreensão ao longo do ano, todo o carinho e apoio demonstrado em todas as minhas decisões e por sempre me ter encorajado a seguir os meus sonhos. Admiro-te sobretudo pela tua competência e humildade, pela transmissão de força e confiança que me dás em todos os momentos.

Resumo

A massiva proliferação de blogues e redes sociais fez com que o conteúdo gerado pelos utilizadores, presente em plataformas como o Twitter ou Facebook, se tornasse bastante valioso pela quantidade de informação passível de ser extraída e explorada. No entanto, a análise de informação proveniente destas fontes apresenta bastantes desafios, devido, não só, ao curto tamanho das mensagens, mas também ao tipo de linguagem usada, que contém inúmeras abreviaturas, erros ortográficos e conteúdo específico da *media social*, o que dificulta significativamente a tarefa de extração de informação a partir deste texto.

A presente tese visa o desenvolvimento de um conjunto de ferramentas que permitem efetuar a análise e extração de conhecimento a partir de várias fontes da Web 2.0, recorrendo ao uso de diversas técnicas de Processamento de Linguagem Natural e representando esse conhecimento através de tecnologias da Web Semântica.

De forma a realizar este objetivo foi desenvolvida uma biblioteca, constituída por vários módulos que possibilitam a extração de informação semântica a partir de notícias online, blogues e publicações provenientes de redes sociais. Foi também desenvolvido um sistema cujo principal objetivo é demonstrar as funcionalidades providenciadas pela biblioteca, permitindo a realização de pesquisa e navegação sobre a informação extraída e representando-a através de tecnologias da Web Semântica. Importa ainda salientar que a biblioteca suporta unicamente a língua portuguesa (português de Portugal) o que, por si só, representa um desafio, visto existirem relativamente poucos recursos disponíveis para o português.

Palavras-Chave

Língua Portuguesa, Media Social, Pesquisa Semântica, Processamento de Linguagem Natural, Web 2.0, Web Semântica, Web Social

Índice

Lista de Figuras

Lista de Tabelas

Acrónimos

AC/DC	Acesso a Corpus/Disponibilização de Corpus
AJAX	Asynchronous Javascript and XML (Javascript Assíncrono e XML)
API	*Application Programming Interface* (Interface de Programação de Aplicações)
BD	Base de Dados
CRF	*Conditional Random Fields* (Campo Aleatório Condicional)
DSP	Desambiguação do Sentido das Palavras
EI	Extração de Informação
HTML	*HyperText Markup Language* (Linguagem de Marcação de Hipertexto)
IA	Inteligência Artificial
JSON	JavaScript Object Notation (Notação de Objetos JavaScript)
LDA	*Latent Dirichlet Allocation* (Modelo de Alocação Latente de Dirichlet)
OWL	*Web Ontology Language* (Linguagem de Representação de Ontologia Baseadas na Web)
PLN	Processamento de Linguagem Natural
RDF	*Resource Definition Framework* (Framework de Definição de Recursos)
RDFS	*Resource Definition Framework Schema* (Esquema RDF)
REM	Reconhecimento de Entidades Mencionadas
SPARQL	*SPARQL protocol and RDF Query Language* (Linguagem de consulta e protocolo de acesso a dados RDF)
SWRL	*Semantic Web Rule Language* (Linguagem de Regras para a Web Semântica)
URI	*Uniform Resource Identifier* (Identificador Uniforme de Recursos)
URL	*Uniform Resource Locator* (Localizador Uniforme de Recursos)
WS	Web Semântica
WWW	*World Wide Web* (Rede de Alcance Mundial)
XML	*Extensible Markup Language* (Linguagem de Marcação Extensível)

Capítulo 1

Introdução

Inicialmente o conteúdo existente na Web era estático e pouco interativo, constituído maioritariamente por páginas contendo informação não editável que limitava o papel do utilizador a um mero espectador (Cormode & Krishnamurthy (2008)). No entanto, em 2004, surgiu uma nova filosofia associada ao uso da Web, cujo propósito passa por fornecer aos utilizadores uma experiência mais envolvente e colaborativa, onde as suas preferências e opiniões passam a ser tidas em consideração. Esta mudança na forma como a Internet é encarada pelos utilizadores deu origem ao termo Web 2.0 (O'Reilly (2007)), criado pela empresa americana O'Reilly Media[1], referindo-se a uma segunda geração de comunidades e serviços que dá ênfase à colaboração e partilha de informação, a denominada *inteligência coletiva* (O'Reilly (2007)).

Consequentemente, tanto redes sociais como blogues foram ganhando bastante importância e são hoje massivamente usados, criando um ambiente bastante desejado para obtenção e exploração de grandes quantidades de informação textual relativa a eventos, notícias, produtos ou entidades (Vickery & Wunsch-Vincent (2007)). No entanto, a análise de informação proveniente destas fontes apresenta inúmeros desafios, sobretudo devido ao tipo de linguagem usada que, regra geral, é pouco estruturada e contém inúmeras abreviaturas, erros ortográficos e conteúdo específico da *media social* (como os populares *emoticons*[2] ou as *hashtags* e menções) dificultando assim a tarefa de interpretação do texto (Hu & Liu (2012)). Muitas vezes é também imposto um limite ao nível de tamanho da mensagem (140 caracteres no caso do Twitter[3]) o que torna ainda mais difícil a extração de conhecimento, uma vez que dificulta a contextualização do conteúdo proveniente da mensagem.

Por esta razão, um dos objetivos desta tese passou precisamente por permitir que o conteúdo proveniente de diversas fontes da *Web* 2.0 (também conhecida por *Web* Social) pudesse ser analisado e possibilitasse ao utilizador efetuar pesquisa e navegação a partir de diversos tipos de informação extraída destas fontes,

[1] Disponível em *http://www.oreilly.com/*.

[2] Sequência de caracteres que pretende transmitir uma emoção. Por exemplo, ":)" para representar alegria e ":(" para representar tristeza.

[3] Disponível em *https://twitter.com/*.

sendo explorado o uso das várias tecnologias da *Web* Semântica (WS) (Berners-Lee, Hendler & Lassila (2001)) e de técnicas de Processamento de Linguagem Natural (PLN) (Jurafsky & Martin (2008)) para esse fim. Desta forma, foram implementadas, entre outras funcionalidades, a deteção de qual a polaridade associada a uma determinada publicação através de técnicas de análise de sentimentos (Liu & Zhang (2012)), reconhecimento de entidades mencionadas que permitem a correta identificação de organizações, localizações, pessoas, etc. (Chinchor & Robinson (1997)) e ainda, extração de tópicos a partir de um conjunto de notícias (Steyvers & Griffiths (2007)).

Os objetivos definidos para esta tese foram então os seguintes:

- Obtenção de conteúdo textual a partir de várias fontes da *Web* 2.0 em português, quer este seja constituído por texto sintaticamente menos estruturado (proveniente sobretudo das redes sociais), como por texto proveniente de blogues e notícias online;
- Extração de vários tipos de informação semântica a partir dessas fontes, através do uso de técnicas de Processamento de Linguagem Natural, nomeadamente:
  - Extração de tópicos;
  - Reconhecimento de entidades mencionadas;
  - Análise de sentimentos;
  - Extração de termos e expressões multipalavra;
  - Extração de triplos.
- Representação da informação extraída usando tecnologias da *Web* Semântica;
- Pesquisa semântica e navegação sobre a informação extraída;
- Visualização de recomendações baseadas em conteúdo relacionado.

A realização destes objetivos traduziu-se na criação de dois componentes: uma **biblioteca** e um **sistema** cujo objetivo é demonstrar as funcionalidades providenciadas pela biblioteca para um determinado conjunto de dados de teste.

A biblioteca é responsável pelo processo de extração dos vários tipos de informação semântica a partir de conteúdo da *Web* 2.0, sendo constituída por vários módulos, cada um responsável pela realização de uma tarefa específica (módulo de extração de tópicos, módulo de extração de entidades, entre outros). O facto da biblioteca ser desenvolvida de forma independente do sistema visa

sobretudo promover a reutilização e manutenção de código, de modo a permitir a sua posterior inclusão noutros projetos.

O sistema tem como principal intuito usar as funcionalidades disponibilizadas pela biblioteca, permitindo a realização de pesquisa semântica e navegação no conteúdo, de forma a facilitar o acesso aos dados mais relevante por parte do utilizador. Toda a informação é representada através de tecnologias da *Web* Semântica, dada a sua flexibilidade e o poder de expressividade que detêm. Tal como referido anteriormente, o propósito deste sistema é permitir que os utilizadores possam visualizar as potencialidades da biblioteca, ou seja, a partir de um conjunto de dados de teste (isto é, de publicações obtidas a partir de diversas fontes da Web 2.0) consigam facilmente visualizar os vários tipos de conhecimento passível de ser extraído. Com esse intuito, foi desenvolvido um **cliente Web** que fornece uma interface visual das funcionalidades do sistema e, consequentemente, das capacidades da biblioteca. Este cliente Web foi, no entanto, desenhado como um protótipo, não visando alcançar um produto final.

Todos os módulos foram desenvolvidos com o objetivo de suportar a língua portuguesa (português de Portugal) o que, por si só, se revelou um desafio, uma vez que existem poucos recursos disponíveis para esta língua.

Resumidamente, as contribuições deste trabalho são as seguintes:

- Elaboração do **estado da arte** referente ao domínio do Processamento de Linguagem Natural e *Web* Semântica para conteúdo da Web 2.0, com foco na língua portuguesa (português de Portugal);
- ***Crawlers***[4] que permitem a obtenção de publicações e comentários provenientes de diversas fontes noticiosas e redes sociais;
- **Biblioteca** que possibilita a extração de vários tipos de conhecimento semântico a partir de recursos textuais em português;
- **Sistema** que realiza pesquisa e navegação sobre a informação extraída, permitindo igualmente efetuar recomendações, a partir de conteúdo da Web 2.0;
- **Cliente Web** que demonstra as funcionalidades do sistema e consequentemente da biblioteca, a partir de um *dataset* de teste;
- Uso das funcionalidades providenciadas pela biblioteca para participação na **avaliação SemEval-2014**[5] (mais concretamente na tarefa de análise

[4] Software desenvolvido com o intuito de navegar na Web para efetuar recolha de informação de forma automatizada.

[5] Referência em *http://alt.qcri.org/semeval2014/*.

de sentimentos em publicações do Twitter) com consequente aceitação de um artigo científico (Leal et al. (in press));

- **Experimentação** realizada à biblioteca, bem como os vários *datasets* elaborados para realização dos testes;
- Conjunto de **recursos léxicos** (dicionários de pitês[6] e *emoticons*, bem como as listas de entidades, entre outros) para a língua portuguesa;
- **Tese** onde é descrito todo o trabalho realizado.

Quanto à estrutura da presente tese, divide-se em mais 5 capítulos e 4 anexos cujo conteúdo é descrito de seguida:

Capítulo 2 introduz os conceitos fundamentais necessários para a compreensão do trabalho desenvolvido na tese. Uma vez que o material existente neste capítulo já se encontrava presente na proposta de tese, foi aqui elaborado um breve resumo do mesmo e movido todo o seu conteúdo para o anexo A.

Capítulo 3 demonstra todo o processo que conduziu à especificação da biblioteca e do sistema. Mais concretamente: na secção 3.1 são apresentados os vários casos de uso com a especificação das possíveis interações de serem realizadas; a apresentação dos vários requisitos que evidenciam as funcionalidades requeridas no solução encontram-se na secção 3.2; a arquitetura do projeto e de cada um dos seus módulos é explicada mais pormenorizadamente na secção 3.3; por fim, na secção 3.4, são apresentados os protótipos relativos à interface do sistema.

Capítulo 4 descreve a metodologia de trabalho adotada nos desenvolvimentos e detalhes de implementação referentes aos vários módulos da biblioteca e sistema. Na secção 4.1 é apresentada a metodologia e também a lista de todas tarefas realizadas no âmbito da tese; os detalhes de implementação relativos aos módulos desenvolvidos, nomeadamente as funcionalidades suportadas por cada um, bem como as ferramentas e bibliotecas usadas, encontram-se especificadas na secção 4.3 para a biblioteca e na secção 4.4 para o sistema. Por último, nas secções 4.5 e 4.6, são fornecidos breves detalhes acerca das tecnologias adotadas no desenvolvimento dos *Web services* e cliente Web, respetivamente.

[6] Linguagem frequentemente adotada pelos utilizadores das redes sociais, que contém inúmeras abreviações, com o objetivo de permitir uma comunicação mais rápida.

Capítulo 5 apresenta a descrição e respetiva análise de resultados dos vários testes efectuados para avaliação do desempenho dos módulos da biblioteca: o primeiro teste é referente ao módulo de Extração de Entidades, mais concretamente ao extrator de entidades vocacionado para texto estruturado (blogues e notícias online), na secção 5.1; a segunda experimentação realizada diz respeito ao Extrator de Entidades vocacionado para texto não estruturado (redes sociais), na secção 5.2; o terceiro teste pretende avaliar o desempenho do módulo de Análise de Sentimentos, na secção 5.3; o quarto teste pretende analisar os resultados obtidos pelo Extrator de Triplos, na secção 5.4; a quinta experimentação executada incide sobre o módulo de Extração de Tópicos, na secção 5.5; por último é apresentada a especificação dos testes unitários e de validação de requisitos elaborados para o sistema na secção 5.6 e secção 5.7, respetivamente.

Capítulo 6 apresenta um resumo da tese, discute as suas contribuições e providencia ideias para possíveis melhoramentos a realizar no trabalho, bem como direcções para futura investigação.

Anexo A apresenta de forma detalhada os conceitos introduzidos no Capítulo 1. Isto inclui: uma introdução ao Processamento de Linguagem Natural, secção A.1; visão geral da arquitetura e das várias tecnologias da Web Semântica na secção A.2; apresentação da Web Social e da terminologia usada nas redes sociais, secção A.3; descrição de trabalhos já desenvolvidos e relacionados com a presente tese, secção A.4; vários recursos linguísticos e léxico-semânticos úteis no âmbito do trabalho são apresentados na secção A.5; algumas das ferramentas e bibliotecas a utilizar nos desenvolvimentos encontram-se descritas na secção A.6.

Anexo B descreve de forma textual os vários casos de uso referentes ao cliente Web, apresentados na Figura 3.1.

Anexo C mostra de forma pormenorizada cada um dos requisitos funcionais apresentados na Tabela 3.1.

Anexo D explica pormenorizadamente as várias classes e propriedades referentes à ontologia mencionada na secção 4.4.1.

Capítulo 2

Estado da Arte

O presente capítulo tem como objetivo fazer um breve resumo dos conceitos fundamentais necessários para a compreensão e análise do trabalho desenvolvido na tese. Esta introdução será muito breve e sucinta, uma vez que todo o seu conteúdo foi movido para o anexo A, por já se encontrar presente na proposta de tese.

Uma das áreas mais exploradas no presente trabalho diz respeito ao Processamento de Linguagem Natural (Jurafsky & Martin (2008)), que consiste na aplicação de métodos e técnicas que possibilitam extrair a semântica da linguagem humana e consequentemente permitir uma interação entre o homem e a máquina. As principais dificuldades encontradas na aplicação de técnicas de PLN a texto têm que ver essencialmente com a ambiguidade que pode ocorrer a vários níveis: fonético, morfológico, sintático, semântico, pragmático e do discurso. A título de exemplo, algumas das tarefas mais conhecidas de PLN são: Reconhecimento de Entidades Mencionadas (Chinchor & Robinson (1997)), Análise de Sentimentos (Liu & Zhang (2012)), Extração de Tópicos (Steyvers & Griffiths (2007)), Desambiguação do Sentido das Palavras (Ide & Véronis (1998)), entre outras.

Torna-se ainda mais desafiante a aplicação das técnicas de PLN aos dados que circulam na *media social* (Appelquist et al. (2010)), ou seja, ao conjunto de aplicações e plataformas existentes na Web que facilitam o compartilhamento de informação (como o caso das redes sociais Facebook[7] e Twitter[8]). De facto, o tamanho reduzido deste tipo de textos, aliado à inexistência de estrutura sintática na grande maioria dos casos, exige um tratamento "diferenciado" relativamente a textos mais estruturados provenientes de notícias online ou blogues. Regra geral, este tipo de texto contém também inúmeras particularidades como a presença de palavras específicas da *media social* ("lol" ou "ftw"), abreviaturas de palavras ("td bem ctg"), uso de *emoticons* (":-)" ou ":-/"), bem como a existência de *hashtags* e menções. Assim, existe a necessidade de realizar um pré-processamento de forma a conseguir identificar e tirar proveito

[7] Disponível em *https://www.facebook.com/*.
[8] Disponível em *https://twitter.com/*.

deste tipo de informação extra para realização (ou melhoramento) das várias tarefas de PLN.

Outra área em relevo nesta tese diz respeito à Web Semântica (WS), definida como "*uma extensão da Web atual no qual a informação possui um significado bem definido, permitindo a cooperação entre computadores e pessoas*" (Berners-Lee, Hendler & Lassila (2001)). A WS assenta sobre um conjunto de padrões tecnológicos que possibilitam a identificação de recursos na Web (assim como a representação de informações referentes a esses recursos) levando à criação de informação legível para as máquinas e consequentemente permitindo a partilha global de conhecimento e reutilização de dados. Este conjunto de padrões forma as diversas camadas em que se divide a arquitetura da WS, onde cada camada explora e usa as capacidades das camadas inferiores. Das várias camadas existentes, salientam-se as seguintes:

- **Camada RDF**, providencia um modelo de descrição lógica dos dados para representação de informações sobre recursos;
- **Camada RDF Schema (ou RDFS)**, permite expandir a especificação básica da RDF (o qual representa unicamente dados) incluindo o suporte necessário para definir classes e restrições de valores das propriedades;
- **Camada Ontologia OWL**, que possibilita a representação de conhecimento de um dado domínio de forma explícita (uma vez que todos os elementos se encontram claramente definidos de forma a evitar ambiguidades) através da criação de um vocabulário comum para partilha de informação;
- **Camada de Regras SWRL**, que expande a expressividade da OWL através da adição de regras a uma ontologia;
- **Camada de Consulta SPARQL**, que se trata de uma linguagem de consulta de triplos RDF, sendo reconhecida como uma das tecnologias chave da WS.

O conceito de pesquisa semântica (Guha et al. (2003)) surge associado à necessidade de melhorar os tradicionais métodos de pesquisa que não realizam nenhuma interpretação ao nível da *query* introduzida pelo utilizador. Este tipo de pesquisa pretende interpretar a semântica associada aos dados, com o intuito de "perceber" qual o significado do conteúdo introduzido na *query* e com isso apresentar resultados mais significativos para o utilizador. Também a componente de recomendação pode beneficiar da incorporação de conhecimento semântico no seu processo, de forma a melhorar a qualidade dos resultados.

A explicação mais pormenorizada dos vários conceitos referidos neste capítulo, bem como dos trabalhos relacionados, recursos linguísticos, ferramentas e bibliotecas, encontra-se presente no anexo A.

Capítulo 3

Especificação da Implementação

Como referido no Capítulo 1, este trabalho teve como objetivo a criação de uma biblioteca para extração de conhecimento a partir de publicações proveniente de várias fontes da Web 2.0 (notícias online, blogues e redes sociais) – desde análise de sentimentos, a extração de termos e expressões multipalavra, extração de entidades, tópicos e também de triplos. Foi ainda elaborado um sistema cujo principal propósito é permitir demonstrar as potencialidades da biblioteca, ou seja, a partir de um conjunto de publicações provenientes de um *dataset* de teste, possibilita a obtenção de vários tipos de conhecimento, bem como permite a realização de pesquisa e de recomendação de conteúdo, através dessa mesma informação. Todas as funcionalidades disponibilizadas pelo sistema são apresentadas ao utilizador através de um cliente Web.

Uma vez que a extração de conhecimento é composta por vários componentes, (que tal como mencionado anteriormente vão desde a extração de tópicos e entidades a análise de sentimentos), efetuou-se a criação de uma biblioteca detentora de uma arquitetura modular, tentando ao máximo que cada módulo fosse o mais desacoplado possível dos restantes. Cada módulo é assim responsável pela realização de uma tarefa específica com um objetivo bem definido, de forma a promover a reutilização e manutenção de código para posterior utilização noutros projetos.

Este capítulo pretende apresentar todo o processo que levou à especificação da biblioteca assim como do sistema e cliente Web, desde a definição dos vários casos de uso para descrição das interações possíveis de serem realizadas pelo utilizador no cliente Web (secção 3.1), ao levantamento dos requisitos funcionais e não-funcionais que permitiram a determinação das suas funcionalidades (secção 3.2). É também apresentada a arquitetura de toda a solução e explicado cada um dos seus módulos de forma mais pormenorizada na secção 3.3. Por fim, na secção 3.4, são apresentados os protótipos relativos ao cliente Web do sistema, com a finalidade de ilustrar e providenciar um melhor entendimento de como será constituída a página Web disponibilizada ao utilizador.

3.1 Casos de Uso

Os casos de uso representam uma sequência de eventos, realizados por um agente externo (denominado ator), que faz uso do sistema para completar uma determinada tarefa, permitindo desta forma explicitar o comportamento esperado. A sua definição permite a especificação de um conjunto de requisitos alto-nível, que auxiliam depois na identificação mais detalhada dos vários requisitos funcionais (Silva & Videira (2001)).

Os casos de uso definidos no âmbito da atual tese são referentes à interação com o cliente Web. Através desta interface os utilizadores podem visualizar os resultados das várias análises efetuadas pelo sistema, realizar pesquisa semântica e navegação no conteúdo, bem como visualizar recomendações de publicações relacionadas.

Na Figura 3.1 é apresentado o diagrama de casos de uso referente ao cliente Web do sistema. Como forma de complemento à Figura 3.1 é apresentado no anexo B uma descrição textual de cada um dos casos, para melhor compreensão das ações a executar pelo utilizador.

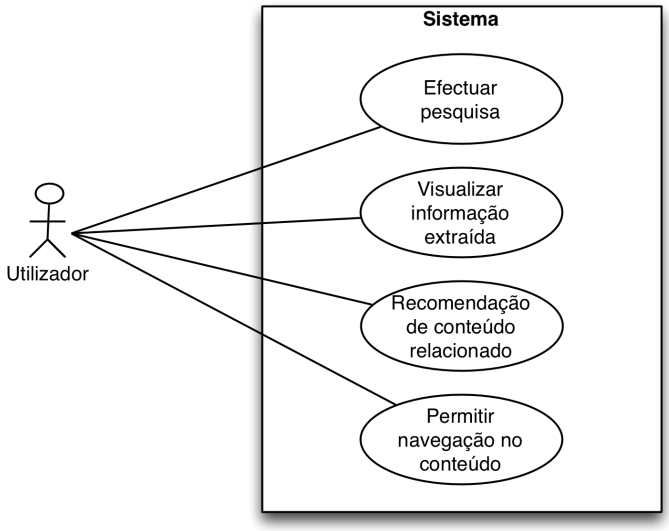

Figura 3.1: Diagrama de casos de uso referentes ao cliente Web do sistema.

3.2 Requisitos

O processo de especificação de requisitos apresenta-se como uma das etapas cruciais no desenvolvimento de um projeto, permitindo identificar quais os objetivos pretendidos e possibilitando, não só definir uma linha orientadora para os desenvolvimentos, como também uma forma de avaliar o sucesso e concretização do projeto.

Nesta secção é então apresentado o resultado do processo de especificação de requisitos, sendo definida uma listagem das funcionalidades requeridas na solução.

3.2.1 Requisitos Funcionais

Os requisitos funcionais fornecem uma descrição daquilo que o sistema deverá realizar, mencionando explicitamente as funcionalidades pretendidas. Os requisitos deverão assim ser coerentes e incluir a descrição de todos os mecanismos e funcionalidades requeridas.

A especificação de requisitos, elaborada no âmbito da tese, é apresentada na Tabela 3.1, sendo no anexo C descritos de forma mais pormenorizada cada um dos 8 requisitos de alto-nível identificados.

Referência	**Descrição**
RF 1	**Obtenção de conteúdo da Web 2.0**
RF 2	**Limpeza e processamento dos dados recolhidos**
RF 3	**Extração de informação a partir dos dados**
RF 3.1	Extração de informação acerca do autor
RF 3.2	Extração de informação relativa ao conteúdo textual recolhido
RF 3.3	Extração de comentários associados a cada publicação
RF 3.4	Extração de dados específicos da *media social*
RF 3.5	Extração de termos e expressões multipalavra
RF 3.6	Reconhecimento de entidades mencionadas
RF 3.7	Extração de tópicos
RF 3.8	Extração de triplos
RF 4	**Representação da informação usando tecnologias da WS**
RF 4.1	Geração de triplos RDF
RF 4.2	Persistência da informação numa base de dados de triplos
RF 5	**Análise de sentimentos/polaridade**
RF 6	**Suporte de pesquisa semântica**
RF 7	**Recomendação de conteúdo relacionado**
RF 8	**Permitir navegação[9] sobre os dados**

Tabela 3.1: Especificação de requisitos funcionais.

[9] Em inglês é utilizado o termo *browsing*.

3.2.2 Requisitos Não Funcionais

Apesar dos requisitos não funcionais descreverem aspetos do sistema que não se encontram diretamente relacionados com o seu comportamento funcional, estes possuem igualmente bastante relevância. Incluem uma ampla variedade de requisitos que se aplicam a diferentes aspetos do sistema, desde usabilidade a performance. No entanto, tendo em conta o facto da presente tese ser de índole mais científica, foram apenas consideradas algumas das categorias apresentadas no modelo FURPS+[10] usado pelo Processo Unificado (Jacobson et al. (1999)).

A especificação de requisitos não funcionais, elaborada no âmbito da tese, é apresentada na Tabela 3.2, sendo depois descritos de forma mais pormenorizada cada um deles.

Referência	Categoria	Descrição
RNF 1	Requisitos de Portabilidade	A biblioteca PLN-PT deverá poder ser executada em vários sistemas operativos.
RNF 2	Requisitos Legais	Uso de licenças de software permissivas.

Tabela 3.2: Especificação de requisitos não funcionais.

Foram assim identificados 2 requisitos de alto-nível explicados mais pormenorizadamente de seguida:

RNF 1: Execução em diversos sistemas operativos

A biblioteca deve poder ser executada em diversos sistemas operativos, nomeadamente Windows[11], Mac OS X[12] e Linux[13]. É então necessário garantir que quer a linguagem de desenvolvimento, quer todas as ferramentas e bibliotecas integradas, são suportadas pelos vários sistemas operativos.

O cumprimento deste requisito foi garantido pelo facto da linguagem adotada para implementação da biblioteca ser Java. Consequentemente todas as ferramentas têm obrigatoriamente que dar suporte a esta linguagem, garantindo assim a sua execução nos diversos sistemas operativos mencionados anteriormente.

[10] Trata-se de um acrónimo que usa a primeira letra de cada categoria de cada uma das categorias de requisitos: Funcionality, Usability, Reliability, Performance e Supportability. O + indica as categorias adicionais. O modelo FURPS foi originalmente proposto por (Grady (1992)).

[11] Disponível em *http://windows.microsoft.com/pt-pt/windows/home.*

[12] Disponível em *https://www.apple.com/pt/osx/.*

[13] Disponível em *http://www.linux.org/.*

RNF 2: Licenças de software permissivas

As licenças de todo o software utilizado devem permitir o seu uso e redistribuição sem qualquer tipo de encargo ou sem acarretar custos adicionais. Fazem parte das licenças consideradas permissivas: LGPL[14], AGPL[15], BSD[16], Apache[17], MIT[18] e Common Public[19].

Este requisito também foi cumprido, uma vez que foram identificadas todas as licenças das várias ferramentas e recursos utilizados no âmbito do projeto, de forma a garantir que apenas fosse incluído software com licença permissiva. Essa análise encontra-se disponível nas várias tabelas presentes no anexo A, na secção de Recursos Linguísticos (secção A.5), bem como na secção de Ferramentas e Bibliotecas (secção A.6).

3.3 Arquitetura

Tendo em conta a análise anterior efetuada ao nível de requisitos e de casos de uso, foi especificada uma arquitetura para a solução composta por 3 camadas, cujo propósito é mencionado de seguida:

- ***Back End*,** onde se encontra toda a lógica da solução. Esta camada é constituída por dois componentes: o sistema e a biblioteca, sendo que o sistema necessita das funcionalidades providenciadas pela biblioteca para cumprir os objetivos de cada módulo.

- **Web *Services*,** disponibilizam serviços que permitem efetuar a comunicação entre o *back end* e o cliente.

- **Cliente,** representa o cliente Web que demonstra todas as funcionalidades providenciadas pela camada de *back end* (sistema e biblioteca).

Na Figura 3.2 é possível visualizar a arquitetura desenhada para a solução, estando identificadas a cinzento cada uma das camadas mencionadas anteriormente. É depois descrito em pormenor cada um dos componentes referidos nas várias camadas (nomeadamente o cliente Web, o servidor que disponibiliza os vários serviços, o sistema e a biblioteca).

[14] Disponível em *https://www.gnu.org/licenses/lgpl.html*.
[15] Disponível em *http://www.gnu.org/licenses/agpl-3.0.html*.
[16] Disponível em *http://opensource.org/licenses/BSD-3-Clause*.
[17] Disponível em *http://www.apache.org/licenses/*.
[18] Disponível em *http://opensource.org/licenses/MIT*.
[19] Disponível em *http://opensource.org/licenses/cpl1.0.php*.

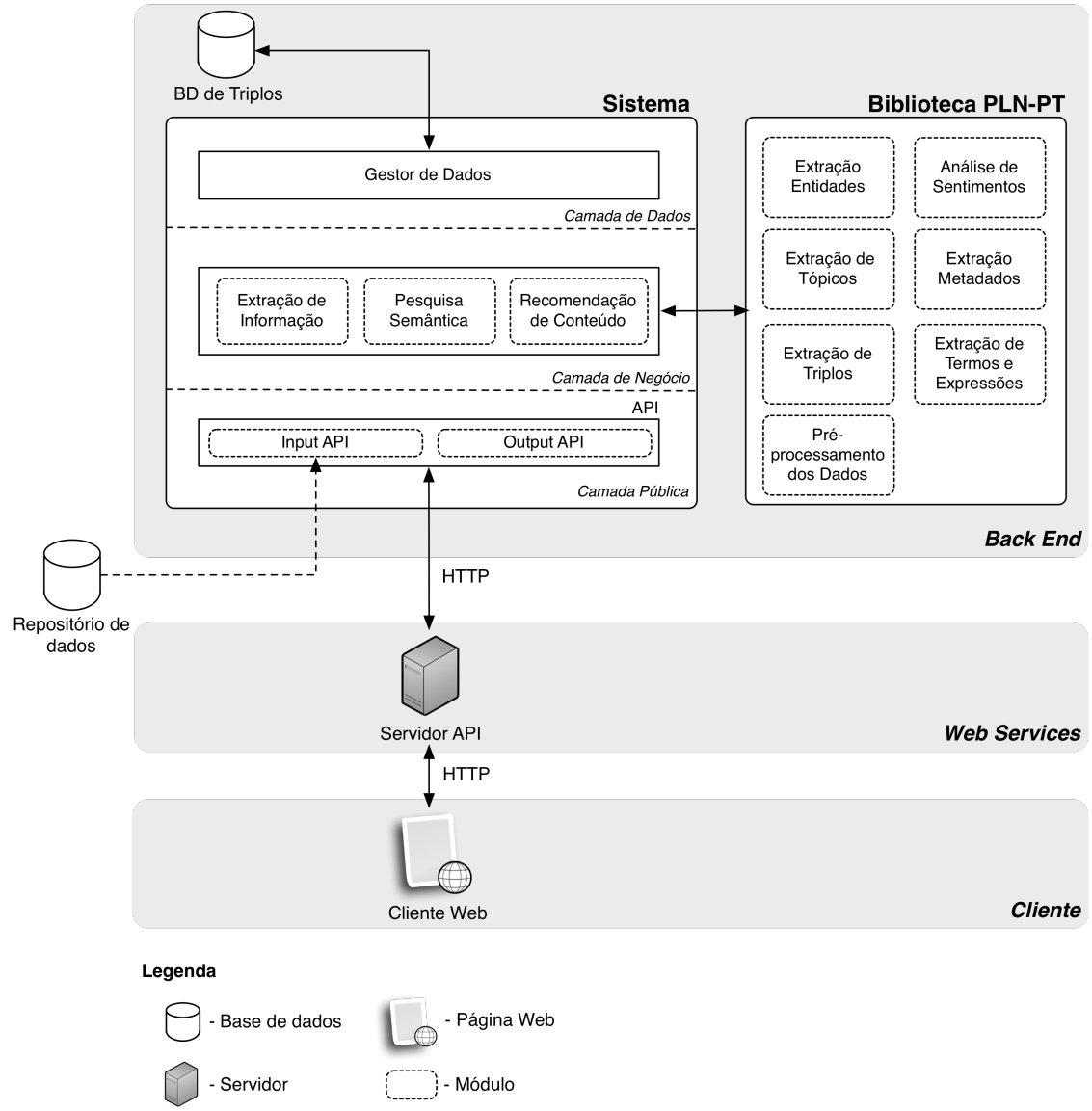

Figura 3.2: Arquitetura do sistema.

3.3.1 Cliente Web

O cliente é constituído por uma página Web, cujo principal objetivo é demonstrar as funcionalidades providenciadas pelo sistema. A página permite ao utilizador efetuar pesquisa sobre os dados, visualizar a informação extraída a partir dos vários módulos da biblioteca para cada uma das publicações (como o caso da extração de entidades, tópicos, análise de sentimentos, etc.), efetuar recomendações ao utilizador e possibilita ainda navegação no conteúdo disponibilizado, através de um conjunto de filtros.

3.3.2 Servidor API

Servidor que inclui os *Web services* responsáveis por disponibilizar as interfaces de integração com o sistema, através do módulo API. Os serviços são invocados pela página Web para obtenção da informação a apresentar ao utilizador.

3.3.3 Sistema

O sistema foi desenvolvido de acordo com um modelo de 3 camadas, de forma a conferir flexibilidade ao projeto e assim estabelecer uma separação entre os seus principais componentes:

- **Camada Pública**: Responsável pela obtenção dos dados e disponibilização das funcionalidades concedidas pelo sistema a agentes externos, através do módulo denominado API. O objetivo passa por simplificar o processo de integração com os vários módulos ao disponibilizar uma API única e simples**.** É constituída por dois submódulos:
  - ***Input* API**, detentor da interface responsável por providenciar os dados ao sistema, que serão alvo de análise e persistência sob a forma de triplos.
  - ***Output* API**, que apresenta uma interface para disponibilização das funcionalidades concedidas pelos módulos do sistema a agentes externos.
- **Camada de Negócio:** Contém todas as regras de negócio do sistema e é constituída por um conjunto de módulos que fazem uso das funcionalidades providenciadas pela biblioteca (explicada na secção 3.3.4) para cumprirem os seus objetivos:
  - **Módulo Extração de Informação**, que permite (para o conjunto de dados fornecido ao sistema) extrair informação semântica usando os vários módulos da biblioteca, para posterior persistência na BD de triplos.
  - **Módulo Pesquisa Semântica**, responsável por obter a *query* introduzida pelo utilizador e efetuar pesquisa semântica sobre os dados, retornando as publicações relevantes no contexto da pesquisa realizada.
  - **Módulo Recomendação de Conteúdo**, responsável por efetuar recomendação de eventuais publicações relevantes ao utilizador. O

motor de recomendação foca-se nas características particulares de uma determinada publicação (entidades, tópicos, termos, etc.) de forma a recomendar um conjunto de outros itens que contenham propriedades similares.

- **Camada de Dados:** Responsável pela gestão da base de dados, disponibilizando uma interface para que seja possível adicionar ou consultar dados previamente armazenados na base de dados de triplos.

3.3.4 Biblioteca PLN-PT

A biblioteca PLN-PT (*Processamento de Linguagem Natural para Português*) foi criada com o intuito de desenvolver um conjunto de módulos flexíveis e detentores de funcionalidades bem definidas, genéricos o suficiente para permitir a sua reutilização noutros projetos. De seguida são apresentados os vários módulos da biblioteca e explicado qual o objetivo pretendido no seu desenvolvimento.

Módulo Extração de Metadados

Módulo cujo objetivo é extrair informação a partir do conteúdo recolhido das várias fontes da Web 2.0. Pretende-se que realize uma filtragem dos dados recolhidos de forma a obter apenas os atributos mais relevantes, nomeadamente informações acerca do autor e da publicação, comentários e conteúdo típico da *media social* como as *hashtags*, menções e URLs.

Módulo Pré-processamento dos Dados

Módulo responsável por disponibilizar operações básicas, necessárias para realização de tarefas mais complexas, providenciadas noutros módulos. Permite também lidar com alguns dos problemas provenientes do texto presente na *media social*. Entre as tarefas disponibilizadas encontra-se a *tokenização*, remoção de *stopwords*[20], etiquetagem gramatical, tradução e normalização de pitês, extração de *emoticons*, entre outras.

Módulo Extração de Tópicos

Este módulo é responsável por efetuar extração de tópicos a partir de um conjunto de textos provenientes de publicações recolhidas. Desta forma é assim possível

[20] Lista das palavras mais frequentes do português que são ignoradas por não acrescentarem informação útil. São exemplos destas palavras "*as*", "*e*", "*os*" e "*de*".

contextualizar cada uma das publicações, através de um conjunto de palavras relacionadas, permitindo identificar qual o âmbito ou contexto mencionado.

Módulo Extração de Entidades

Este módulo permite efetuar extração de entidades (pessoas, organizações, localizações, etc.) de um determinado texto. É usado com o objetivo de, para cada publicação obtida, identificar eventuais entidades mencionadas que auxiliem também na contextualização dos dados.

Módulo Extração de Triplos

Este módulo é responsável por, a partir de uma frase, efetuar a extração de um triplo na forma de sujeito, predicado e valor, possibilitando assim a redução da dimensionalidade do texto. Neste caso, a frase deverá obrigatoriamente deter estrutura sintática de forma a permitir a correta identificação dos diversos componentes do triplo.

Módulo Análise de Sentimentos

Responsável por atribuir um sentimento ou polaridade (positivo, negativo ou neutro) a um determinado texto constituído por uma ou mais opiniões.

Módulo Extração de Termos

Módulo cujo objetivo é efetuar a extração de termos pertencentes a uma determinada categoria como nomes, verbos, adjetivos ou advérbios. Permite ainda extrair expressões multipalavra, como o caso de *Universidade de Coimbra*, *Fundação Manuel dos Santos* ou *José Saramago*, que denotam expressões cujas palavras fazem sentido que sejam agrupadas.

3.4 Protótipo da Interface

Nesta secção são apresentados os protótipos relativo à interface do sistema, utilizados para ilustrar e providenciar um melhor entendimento de como deve ser constituída a página Web disponibilizada ao utilizador.

São então de seguida apresentados os dois protótipos elaborados para o cliente Web da aplicação, que correspondem ao ecrã de listagem de resultados e de detalhes da publicação.

Protótipo de Listagem de Resultados

Na Figura 3.3 é possível visualizar a interface de listagem de resultados associados a uma pesquisa efetuada pelo utilizador.

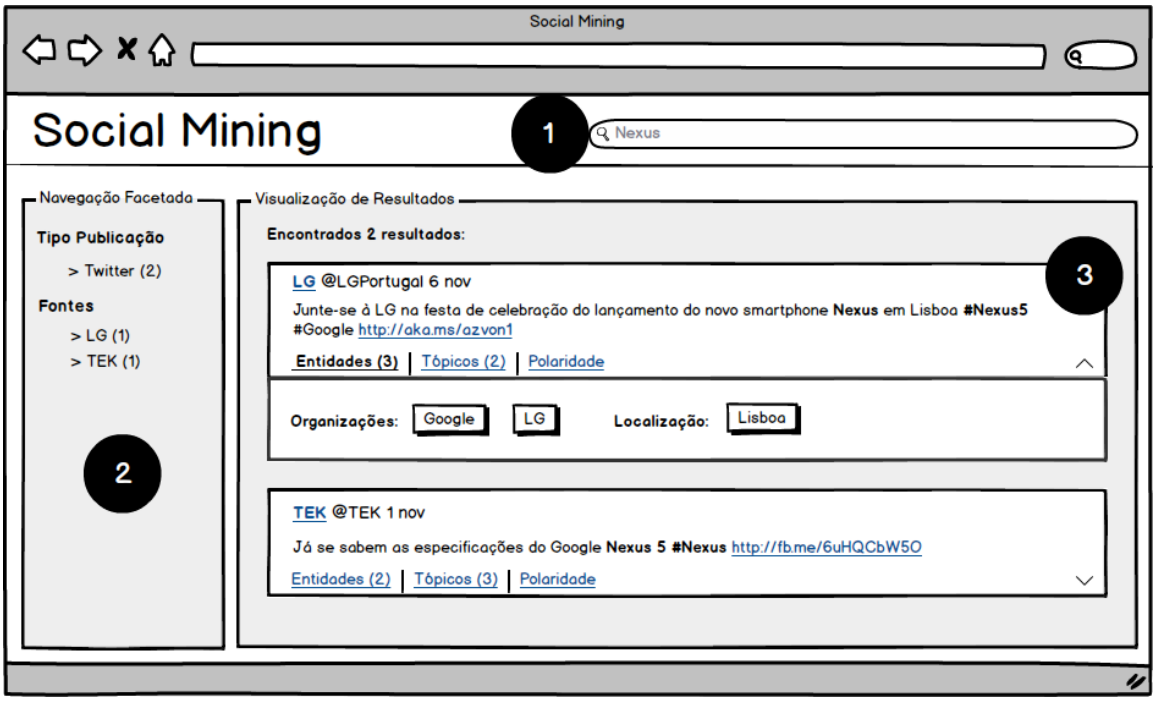

Figura 3.3: Protótipo de listagem de resultados.

O protótipo é constituído essencialmente por três áreas, representadas na imagem anterior através de algarismos, explicadas de seguida:

1. **Pesquisa:** onde o utilizador pode inserir as palavras-chave para efetuar uma pesquisa.

2. **Navegação facetada:** onde é apresentado um conjunto de facetas, isto é, um conjunto de termos/valores que são utilizados como filtros para o utilizador selecionar apenas as publicações que pretende visualizar.

3. **Visualização de resultados:** que permite visualizar os resultados da pesquisa efetuada. Cada uma das publicações possui associada uma zona expansível, usada com o intuito de rapidamente visualizar alguma da informação extraída a partir do conteúdo da publicação.

Desta forma, o utilizador começa por inserir as palavras-chave na caixa de texto de pesquisa (1) e, ao pressionar o botão de pesquisa, são apresentados os respetivos resultados em (3). Caso pretenda, pode ainda aplicar os filtros, disponibilizados em (2), sobre os resultados apresentados. O utilizador consegue ainda

selecionar uma publicação das listadas em (3), sendo depois reencaminhado para o protótipo *detalhes da publicação*.

Protótipo de Detalhes da Publicação

Na Figura 3.4 é possível visualizar a interface referente aos detalhes de uma publicação, visível sempre que o utilizador selecionar uma das publicações listadas no protótipo anterior (listagem de resultados). Esta interface foi criada com o objetivo de mostrar conteúdo cujo texto é demasiadamente longo (como o caso de notícias online e blogues) e também para mostrar informação que de outra forma ocuparia demasiado espaço no ecrã anterior.

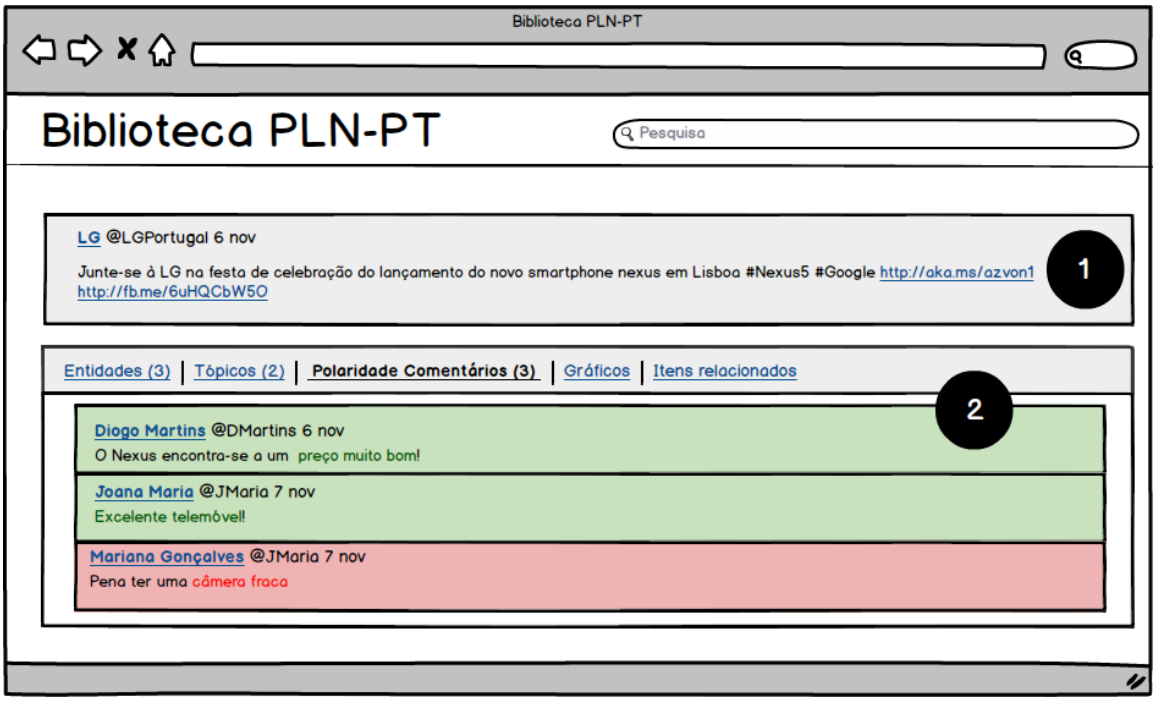

Figura 3.4: Protótipo de detalhes de publicação.

O protótipo evidencia a existência de duas áreas:

1. **Texto da publicação:** onde o utilizador pode visualizar o texto referente à publicação que selecionou, sendo útil sobretudo para textos demasiadamente longos (como os de notícias online).

2. **Visualização de informação:** onde é conferido o acesso a todas as informações extraídas a partir da publicação. Esta decisão foi tida em conta devido ao espaço que alguns elementos ocupavam, como o caso dos comentários e recomendações que podem ser bastante extensos.

O utilizador pode assim visualizar toda a informação referente à publicação apresentada em (1), acessível através dos separadores selecionáveis em (2).

Capítulo 4

Metodologia e Implementação

Neste capítulo é descrita a metodologia de trabalho usada nos desenvolvimentos efetuados (secção 4.1), sendo também apresentada a listagem de todas as tarefas realizadas no âmbito da tese. O plano de trabalho desenvolvido ao longo de cada semestre, bem como a respetiva definição das tarefas executadas em cada um dos ciclos de desenvolvimento, é visível na secção 4.2.

Detalhes relativos à implementação dos vários módulos da biblioteca são referenciados na secção 4.3 e do sistema na secção 4.4; sendo, em ambas as secções, fornecida informação acerca do processo e funcionalidades implementadas na sua construção, bem como uma breve descrição das respetivas tecnologias usadas e classes desenvolvidas. Por fim, na secção 4.5 e secção 4.6, são apresentadas, de forma bastante sucinta, as principais tecnologias adotadas na implementação dos *Web services* e cliente Web, respetivamente.

4.1 Metodologia de Desenvolvimento

O processo de desenvolvimento de software usado foi baseado na metodologia ágil Scrum (Schwaber & Beedle (2002)), que visa dar resposta às frequentes alterações de requisitos realizadas ao longo de um projeto, permitindo executar sucessivas avaliações durante o seu desenvolvimento e efetuar os ajustamentos necessários. Uma vez que apenas existe um único elemento na equipa de implementação, foi adotada uma variação da metodologia Scrum, de forma a aplicar alguns dos seus princípios, que permitiram seguir uma filosofia de desenvolvimento iterativa e incremental, mas sem cumprir de forma rígida todos os seus procedimentos.

Todo o trabalho realizado foi adicionado a uma lista denominada *product backlog* (apresentada na secção 4.1.1), que contém todas as funcionalidades desenvolvidas no âmbito do projeto. A fase de implementação foi dividida em vários ciclos de desenvolvimento (denominados *sprints*), com duração de cerca de 4 semanas, ou seja, de aproximadamente 1 mês. Foi realizada uma reunião de planeamento no início de cada ciclo, de forma a rever o trabalho desenvolvido ao longo do *sprint* e definir quais as tarefas mais prioritárias que deveriam ser implementadas no próximo. Existiram ainda reuniões semanais para acompanhamento da evolução do trabalho e de eventuais desvios ocorridos.

4.1.1 *Product Backlog*

O conjunto de tarefas que foram implementadas ao longo dos vários *sprints*, são visíveis no *product backlog* apresentado na Tabela 4.1.

ID	Descrição
1	**Criação de *dataset* com conteúdo da Web 2.0**
1.1	Selecionar fontes para obtenção de conteúdo
1.2	Obter conteúdo proveniente de redes sociais
1.3	Obter conteúdo proveniente de notícias Web
1.4	Obter conteúdo proveniente de blogues
1.5	Efetuar limpeza e tratamento inicial dos dados
1.6	Persistência dos dados numa BD.
2	**Desenvolvimento do módulo de Extração de Metadados**
2.1	Integrar *parser*[21] para extrair o conteúdo pretendido a partir dos dados recolhidos
2.2	Obter dados acerca do autor da publicação
2.3	Obter dados acerca do conteúdo da publicação (texto, data e descrição)
2.4	Obter comentários associados a cada publicação
2.5	Obter dados específicos da *media social* (*hashtags*, menções e URLs)
3	**Desenvolvimento do módulo de Pré-processamento dos Dados**
3.1	Selecionar e integrar *tokenizador*
3.2	Selecionar e integrar *tokenizador* específico para *microblogging*
3.3	Selecionar e integrar corretor ortográfico
3.4	Selecionar e integrar etiquetador gramatical
3.5	Criar dicionário de *stopwords*[22] e possibilitar a sua remoção no texto
3.6	Selecionar e integrar lematizador
3.7	Integrar analisador sintático de dependências
4	**Tratamento de dados da *media social***
4.1	Criar dicionário de pitês
4.2	Criar dicionário de *emoticons*
4.3	Criar dicionário com expressões regulares de *emoticons*
4.4	Identificar e traduzir palavras em pitês
4.5	Identificar e extrair *emoticons*
4.6	Normalizar *emoticons* e pitês presente em texto

[21] Código que efetua a leitura de um documento disponibilizado num determinado formato, analisando a sua estrutura de forma a extrair a informação pretendida.

[22] Lista das palavras mais frequentes do português que são ignoradas por não acrescentarem informação útil. São exemplos destas palavras "*as*", "*e*", "*os*" e "*de*".

5	**Desenvolvimento do módulo de Extração de Termos e Expressões Multipalavra**
5.1	Definir padrões sintáticos para obtenção de expressões multipalavra
5.2	Extrair termos de uma determinada classe gramatical (verbos, advérbios, nomes comuns e próprios, bem como adjetivos)
5.3	Extrair expressões multipalavras, com base nos padrões definidos anteriormente
6	**Desenvolvimento do módulo de Extração de Tópicos**
6.1	Integrar modelo para extração de tópicos
6.2	Gerar o modelo a partir de um conjunto de dados
6.3	Possibilitar a persistência do modelo
6.4	Realizar experimentação no módulo
7	**Desenvolvimento do módulo de Extração de Entidades**
7.1	Desenvolver *parser* para obtenção de entidades mencionadas a partir de conjunto de textos previamente etiquetados
7.2	Selecionar e extrair conjunto de atributos relevantes para representar os textos etiquetados
7.3	Construir manualmente listas de entidades para auxiliar no processo de reconhecimento de entidades
7.4	Integrar classificador para reconhecimento de entidades mencionadas
7.5	Gerar modelo a partir do conjunto de textos
7.6	Realizar experimentação no módulo
8	**Desenvolvimento de um extrator de entidades vocacionado para texto proveniente de redes sociais**
8.1	Construir manualmente listas de entidades
8.2	Identificar conjunto de regras e expressões regulares para auxiliar na identificação de entidades
8.3	Selecionar uma única entidade, quando palavra tem várias associadas.
8.4	Etiquetar manualmente um conjunto de entidades presentes em publicações de redes sociais para realização de testes
8.5	Realizar experimentação no módulo
9	**Desenvolvimento do módulo de Análise de Sentimentos**
9.1	Identificar palavras e expressões detentoras de polaridade
9.2	Identificar advérbios de negação (modificadores de polaridade)
9.3	Identificar *emoticons* e atribuir polaridade a cada um
9.4	Cálculo do valor de polaridade
9.5	Realizar experimentação no módulo
10	**Desenvolvimento do módulo de Extração de Triplos**
10.1	Analisar *output* proveniente do analisador sintático de dependências
10.2	Representar *output* através de uma estrutura de dados em árvore

10.3	Definir conjunto de regras para identificação de sujeito, predicado e objeto numa frase
10.4	Aplicar as regras para extração de triplos
10.5	Anotação manual de triplos a partir de um conjunto de publicações para realização de testes
10.6	Realizar testes no módulo
11	**Persistência da informação recolhida numa base de dados de triplos**
11.1	Criar ontologia – especificar as várias classes e propriedades
11.2	Desenvolver módulo Gestor de Dados (para integrar base de dados de triplos).
11.3	Desenvolver módulo de Extração de Informação do sistema (invocar os vários módulos da biblioteca para obter os resultados a persistir na base de dados de triplos)
11.4	Obter triplos a partir da informação recolhida
11.5	Persistir a informação na base de dados de triplos
12	**Desenvolver módulo de Pesquisa Semântica**
12.1	Interpretação da *query* introduzida pelo utilizador
12.2	Correspondência de resultados entre a *query* e os dados existentes na base de dados de triplos
12.3	*Ranking* dos resultados a apresentar
12.4	Realizar testes unitários
13	**Desenvolver sistema de recomendação baseado em conteúdo**
13.1	Definir que tipo de informação (proveniente da publicação) será usado para recomendar outro conteúdo similar
13.2	Definir relevância associada a cada tipo de informação considerada
13.3	Atribuir uma pontuação a cada recomendação, de forma a permitir efetuar um *ranking* de resultados
13.4	Implementar motor de recomendação
13.5	Realizar testes unitários
14	**Desenvolver *Web services* que disponibilizem interfaces de integração com o sistema**
14.1	Disponibilizar interface de pesquisa semântica
14.2	Disponibilizar interface de recomendação de conteúdo
15	**Desenvolver cliente Web**
15.1	Possibilitar a realização de pesquisa semântica sobre o conteúdo recolhido
15.2	Para cada publicação, visualizar a informação obtida a partir de cada um dos módulos da biblioteca, bem como recomendações de conteúdo
15.3	Suportar navegação (em inglês *browsing*) sobre os dados, através da disponibilização de filtragem dos dados

Tabela 4.1: Listagem de tarefas do *product backlog.*

4.2 Trabalho Desenvolvido

Ao longo do primeiro semestre foi elaborado o estado da arte, realizando-se uma pesquisa acerca de trabalhos relacionados e potenciais recursos e ferramentas a serem usados no âmbito do projeto. Foi depois efetuada a análise e especificação da solução, que conduziu à elaboração da lista de tarefas a desenvolver, presentes no *product backlog* disponibilizado na secção 4.1.1. Iniciou-se depois a implementação do projeto, começando pela criação de um *dataset* para auxílio no desenvolvimento e teste dos vários módulos. Seguiu-se a implementação dos módulos da biblioteca, nomeadamente o módulo de Extração de Metadados, Pré-processamento de Dados, Extração de Termos e Expressões Multipalavra, Extração de Tópicos e Extração de Entidades.

No segundo semestre foram continuados os desenvolvimentos de modo a finalizar a implementação dos restantes módulos da biblioteca, nomeadamente do módulo de Análise de Sentimentos, módulo de Extração de Entidades (vocacionado para texto não estruturado, ou seja, proveniente de redes sociais) e, por fim, o módulo de Extração de Triplos. Foram também realizados vários testes a fim de aferir o desempenho dos vários módulos desenvolvidos. Por fim, foi implementado o sistema e o cliente Web para demonstração das funcionalidades providenciadas pela biblioteca.

4.2.1 *Sprints* Realizados

De seguida são evidenciadas as funcionalidades implementadas em cada um dos ciclos de desenvolvimento, ou seja, em cada um dos *sprints* realizados:

Sprint #1 (29/10/2013 a 02/11/2013)
Criação de *dataset* com conteúdo da Web 2.0.
Desenvolvimento do módulo de Extração de Metadados.
Desenvolvimento do módulo de Pré-processamento de Dados.
Tratamento de dados da *media social*.

Sprint #2 (02/12/2013 a 30/12/2013)
Desenvolvimento do módulo de Extração de Termos e Expressões Multipalavra.
Desenvolvimento do módulo Gestor de Dados.
Desenvolvimento do módulo de Extração de Tópicos.
Desenvolvimento do módulo de Extração de Entidades (texto estruturado – blogues e notícias online).
Escrita da proposta de tese.

Sprint #3 (30/12/2013 a 03/02/2014)
Continuação do desenvolvimento do módulo de Extração de Entidades (texto estruturado – blogues e notícias online).
Escrita da proposta de tese.

Sprint #4 (03/02/2014 a 03/03/2014)
Desenvolvimento do módulo de Análise de Sentimentos.
Desenvolvimento do módulo de Extração de Entidades (texto não estruturado – redes sociais).

Sprint #5 (03/03/2014 a 31/03/2014)
Continuação do desenvolvimento do módulo de Extração de Entidades (texto não estruturado – redes sociais).
Desenvolvimento do módulo de Extração de Triplos.

Sprint #6 (01/04/2014 a 28/04/2014)
Experimentação no módulo de Extração de Tópicos.
Desenvolvimento inicial da infraestrutura do sistema (versão inicial dos *Web services* e cliente Web que permita testar o protótipo).
Persistência da informação recolhida numa base de dados de triplos.
Desenvolvimento do módulo de Recomendação.

Sprint #7 (28/04/2014 a 02/06/2014)
Desenvolvimento da pesquisa semântica.
Finalização do desenvolvimento dos *Web services*.
Finalização do desenvolvimento do cliente Web.
Realização de testes unitários e de validação de requisitos no sistema.

Sprint #8 (02/06/2014 a 30/06/2014)
Escrita da tese.

4.3 Detalhes de Implementação da Biblioteca

Esta secção visa apresentar detalhes de implementação relativos aos módulos da biblioteca desenvolvidos, nomeadamente as funcionalidades suportadas por cada um, bem como as ferramentas e bibliotecas usadas. São ainda referidas as várias classes definidas, com uma breve descrição relativa ao objetivo principal de cada uma.

4.3.1 Construção do *Dataset*

Para o desenvolvimento e teste dos vários módulos implementados, assim como para demonstração das diversas funcionalidades providenciadas pela biblioteca, foi necessário construir um *dataset* com conteúdo proveniente da Web 2.0. As fontes selecionadas para extração dos dados foram as seguintes:

- **Twitter:** Os dados extraídos a partir do Twitter foram na sua maioria retirados de fontes noticiosas, existindo a preocupação em extrair conteúdo que abrangesse diversas áreas. Desta forma, foram obtidas notícias de índole mais generalista (provenientes do jornal Público[23], Lusa[24] e RTP[25]), notícias de economia (Diário Económico[26]), tecnologia (TEK[27], Exame Informática[28] e Pplware[29]), desporto (Record[30]), entre outras. Foram também obtidos dados provenientes de personalidades portuguesas influentes como o caso de Cavaco Silva[31], Pedro Passos Coelho[32] ou Ricardo Costa[33]. No total foram recolhidos cerca de 50 mil *tweets*.

- **Facebook:** Os dados obtidos a partir desta rede social incidiram essencialmente sobre notícias generalistas (provenientes do jornal Público, Lusa e P3[34]), de cariz mais tecnológico (retiradas do TEK), notícias desportivas (provenientes do Sapo Desporto[35]), Lazer (Life&Style[36]) e também fontes mais vocacionadas para divulgação de produtos (como a Nokia[37], Microsoft[38] e Asus[39]). A escolha mais restrita deveu-se sobretudo ao facto de serem fontes com grande número de publicações diárias e estas conterem algum conteúdo textual, algo que não se verifica em algumas

[23] Disponível em *http://www.publico.pt/*.
[24] Disponível em *http://www.lusa.pt/default.aspx*.
[25] Disponível em *http://www.rtp.pt/noticias/*.
[26] Disponível em *http://economico.sapo.pt/*.
[27] Disponível em *http://tek.sapo.pt/*.
[28] Disponível em *http://exameinformatica.sapo.pt/*.
[29] Disponível em *http://pplware.sapo.pt/*.
[30] Disponível em *http://www.record.xl.pt/*.
[31] Referência em *http://www.presidencia.pt/?idc=3*.
[32] Referência em *http://www.portugal.gov.pt/pt/os-ministerios/primeiro-ministro/conheca-a-equipa/primeiro-ministro/pedro-passos-coelho.aspx*.
[33] Referência em *http://expresso.sapo.pt/ricardo-costa=s25033*.
[34] Disponível em *http://p3.publico.pt/*.
[35] Disponível em *http://desporto.sapo.pt/*.
[36] Disponível em *https://lifestyle.publico.pt/*.
[37] Disponível em *http://www.nokia.com/global/*.
[38] Disponível em *http://www.microsoft.com/pt-pt/default.aspx*.
[39] Disponível em *http://www.asus.com/pt/*.

fontes noticiosas, onde as publicações apenas contêm referências para o URL da respetiva notícia online. Foram obtidas aproximadamente 20 mil publicações.

- **Notícias Online:** As fontes escolhidas para extração foram algumas das já selecionadas no Facebook, ou seja, notícias do Público, TEK e Sapo Desporto. Esta decisão deveu-se ao facto do conteúdo extraído a partir do Facebook conter já o URL da notícia respetiva, facilitando desta forma a sua obtenção. Foram assim extraídas cerca de 2,300 notícias a partir de cada uma das fontes selecionadas, tendo sido realizada uma filtragem prévia, de forma a remover notícias que não contivessem texto (como vídeos ou fotogalerias).

- **Blogues:** No caso dos blogues foram recolhidos textos de opinião provenientes de vários autores do jornal Expresso[40]. Cada blogue foi escolhido com o intuito de abordar uma determinada área - desde desporto, política e física a tecnologia. Devido sobretudo às atualizações menos frequentes dos blogues, foram recolhidos somente cerca de 200 textos de opinião.

O facto de por vezes terem sido selecionadas as mesmas fontes para recolha de notícias (nomeadamente o jornal Público e Tek Sapo), deveu-se ao facto de poder ser interessante, para a mesma notícia, visualizar o resultado dos vários módulos da biblioteca ao lidar com os diferentes tipos de texto. Assim, apesar da possível duplicação de conteúdo, uma vez que o foco do trabalho é a biblioteca e o principal objetivo do sistema a demonstração das suas funcionalidades, optou-se por manter esta seleção de fontes.

No que diz respeito às APIs usadas para efetuar a extração de *tweets*, foi utilizada a biblioteca Twitter4J (já apresentada na subsecção A.6.3) que faz uso da Streaming API do Twitter para obtenção de *tweets* em tempo real, sendo aplicado um filtro para obter apenas os *tweets* das fontes pretendidas, através do seu identificador respetivo. O uso da Streaming API deveu-se essencialmente ao facto de permitir efetuar a recolha de comentários, essenciais para o trabalho a desenvolver. No caso do Facebook foi usada a biblioteca RestFB, igualmente apresentada na subsecção A.6.3.

Relativamente aos blogues e notícias online, visto não existir nenhuma API para obtenção do seu conteúdo, o texto e comentários foram obtidos diretamente a partir do HTML das páginas respetivas. Pelo facto de ser um processo mais moroso, o número de fontes selecionadas neste caso foi também mais restrito.

Uma vez que a API do Twitter apenas dá suporte oficial ao formato JSON (Crockford (2006)), optou-se por persistir todos os dados neste formato, numa

[40] Disponível em *http://expresso.sapo.pt/*.

base de dados MongoDB[41]. Utilizou-se esta BD apenas como repositório temporário, uma vez que este conteúdo foi utilizado para realizar as experimentações e também para demonstração do sistema criado (neste caso os dados foram persistidos numa base de dados de triplos Sesame).

Optou-se por não apresentar os vários projetos e classes desenvolvidas na obtenção do *dataset*, uma vez que não possuem muita relevância no âmbito do trabalho desenvolvido.

4.3.2 Módulo Extração de Metadados

O módulo de Extração de Metadados, tem como objetivo efetuar a extração da informação mais relevante, obtida a partir do conteúdo recolhido de cada uma das fontes de informação (nomeadamente Facebook, Twitter, notícias online e blogues).

Neste módulo é dado suporte unicamente ao formato de texto JSON, uma vez que é o único formato suportado oficialmente pela API do Twitter. Para isso foi então desenvolvido um conjunto de classes (uma para cada tipo de fonte de informação a processar), que disponibilizam métodos para que, ao receber como parâmetro uma publicação em formato JSON, seja devolvido um objeto contendo todas as propriedades da publicação (autor da publicação, conteúdo e data da publicação, comentários, etc.). Para realizar o processamento dos dados JSON foi utilizada a biblioteca JSON-java[42].

O módulo de Extração de Metadados encontra-se implementado no *package* denominado *metadataExtraction*, sendo as suas classes e interfaces descritas na Tabela 4.2.

Nome	**Tipo**	**Descrição**
MicroblogInfo	Interface	Interface que define as operações padrão para extração de metadados a partir de publicações de *microblogging* (Facebook e Twitter).
TwitterInfoExtractor	Classe	Implementa a interface "MicroblogInfo" definindo todas as operações relacionadas com a extração de metadados a partir de publicações do Twitter.

[41] Disponível em *http://www.mongodb.org/.*
[42] Disponível em *https://github.com/douglascrockford/JSON-java.*

FacebookInfoExtractor	Classe	Implementa a interface "MicroblogInfo" definindo todas as operações relacionadas com a extração de metadados a partir de publicações do Facebook.
BlogInfo	Interface	Interface que define as operações padrão para extração de metadados a partir de notícias online e blogues de opinião.
BlogInfoExtractor	Classe	Implementa a interface "BlogInfo" definindo todas as operações relacionadas com a extração de metadados a partir de publicações de notícias online e blogues.
Comment	Classe	Define um comentário associado a uma publicação.
Author	Classe	Define o autor associado a uma publicação.

Tabela 4.2: Elementos incluídos no *package* "*metadataExtraction*" pertencente à biblioteca.

4.3.3 Módulo Pré-Processamento dos Dados

Este módulo é responsável por disponibilizar um conjunto de operações consideradas básicas, visto servirem como base à realização de tarefas de maior complexidade. O conjunto de tarefas implementadas é o seguinte:

- ***Tokenizador***: efetuada a integração com o *tokenizador* para Português disponibilizado pelo OpenNLP[43].

- ***Tokenizador* vocacionado para *microblogging*:** trata-se de um *tokenizador* mais indicado para textos curtos e pouco estruturados, implementado recorrendo ao *tokenizador Sylvester UGC* [44], que contempla os requisitos necessários para lidar com este tipo de conteúdo.

- **Lematizador**: integrado usando o lematizador para português desenvolvido pelo colega Ricardo Rodrigues[45] do laboratório KIS (*Knowledge and Intelligent Systems Lab*) do CISUC[46].

- **Analisador sintático de dependências**: integrado usando o analisador sintático de dependências para português desenvolvido também pelo colega Ricardo Rodrigues. O objetivo passa por, a partir de uma frase, gerar

[43] Disponível em *https://opennlp.apache.org/*.
[44] Disponível em *http://labs.sapo.pt/2011/11/sylvester-ugc-tokenizer/*.
[45] Referência em *https://www.cisuc.uc.pt/people/show/2203*.
[46] Disponível em *https://www.cisuc.uc.pt/*.

uma árvore sintática em que a sua estrutura é descrita unicamente em termos das palavras e das relações existentes entre elas.

- **Remoção de *stopwords*[47]:** de forma a realizar esta tarefa, foi manualmente definido um conjunto de *stopwords*, de forma a ser possível efetuar a sua identificação e respetiva remoção do texto. Exemplos de *stopwords* identificadas: "a", "as", "de", "dos", "com".

- **Etiquetagem gramatical:** foi usado o etiquetador gramatical do OpenNLP, de forma a obter a categoria lexical associada a cada palavra de um texto.

- **Extração de metadados de *tweets*:** tarefa responsável por extrair *hashtags*, menções e URLs de *tweets*, através do uso de expressões regulares.

- **Tratamento de metadados de *tweets*:** disponibilização de um conjunto de funcionalidades para lidar com as *hashtags*, menções e URLS. Permite, por exemplo, remover os URLs presentes num texto, remover os símbolos que identificam as *hashtags* e menções (# e @) ou ainda identificar as entidades presentes nas *hashtags* (substituir "#CriticalSoftware" por "Critical Software"). O objetivo é remover todo o conteúdo que poderá causar ruído na realização de outras tarefas (como extração de tópicos, extração de entidades, etc.).

- **Normalização de pitês:** consiste na normalização de linguagem habitualmente usada no meio virtual, sendo um passo bastante importante para uma correta identificação deste tipo de expressões. De forma a conseguir realizar esta normalização foi criado um dicionário em que, associado a cada expressão regular, encontra-se a palavra normalizada, pelo qual deve ser substituída. Assim, acrónimos como "lolololol" ou "looollll" são substituídos pela sua forma normalizada "lol", possibilitando a sua correta identificação e tradução respetiva. De forma a tornar o processo mais abrangente, realizou-se ainda a deteção de todas as palavras que possuem mais de três caracteres consecutivos repetidos, sendo reduzidos a um único carácter.

- **Tradução de pitês:** consiste na tradução de texto que contém pitês, na sua forma "correta" sem abreviações. Para isso foi também elaborado um dicionário de pitês, contendo a tradução respetiva de cada palavra e possi-

[47] Lista das palavras mais frequentes do português que são ignoradas por não acrescentarem informação útil.

bilitando assim a substituição destes termos. A título de exemplo, "ctg" e "td" são substituídos por *contigo* e *tudo*, respetivamente.

- **Extração de *emoticons*:** tarefa responsável por extrair os *emoticons* presentes num texto, bem como o seu significado. Mais uma vez esta funcionalidade foi implementada recorrendo a um dicionário, onde para cada *emoticon*, foi definida a emoção que se pretende transmitir com o seu uso. Por exemplo, o *emoticon* denotado por ":)" representa alegria e ":(" tristeza. Trata-se de uma tarefa bastante importante sobretudo para o módulo de Análise de Sentimentos.

- **Corretor ortográfico:** pretende disponibilizar sugestões de correção para palavras que contêm erros ortográficos. Foi implementado usando o corretor ortográfico Hunspell[48], que retorna um conjunto de possíveis sugestões de correção para cada palavra com erros ortográficos. Para escolher a sugestão considerada mais "acertada", foi selecionada a que retornava a menor distância de *Levenshtein* (Gilleland (2009)), ou seja, a que possui o número mínimo de operações de inserção, remoção ou substituição de cada caractere.

O módulo de Extração de Metadados encontra-se implementado no *package* denominado *dataPreprocessing*, sendo as classes apresentadas na Tabela 4.3.

Nome	**Tipo**	**Descrição**
PosTagger	Classe	Implementa as operações responsáveis pela realização de etiquetagem gramatical a partir de um determinado texto.
StopwordRemoval	Classe	Implementa as operações responsáveis pela remoção das *stopwords* presentes num texto.
WebSlangProcessor	Classe	Implementa as operações que permitem normalizar e efetuar tradução de texto que contenha expressões pitês, *emoticons* e conteúdo específico da *media social* como menções, *hashtags* e URLs.
MetadataExtraction	Classe	Implementa as operações que permitem obter menções, *hashtags* e URLs a partir de *tweets*.
SpellChecker	Classe	Implementa as operações responsáveis pela disponibilização de um corretor ortográfico.

[48] Disponível em *http://hunspell.sourceforge.net/*.

PreProcessingTasks	Classe	Implementa operações mais complexas, que requerem o uso das funcionalidades providenciadas pelas restantes classes. Por exemplo, a operação de pré-processamento de publicações de *microblogging* envolve efetuar a *tokenização* do texto, bem como a normalização e tradução de pitês.
Tokenizer	Classe	Implementa as operações responsáveis pela disponibilização dos *tokenizadores*, quer de texto estruturado como de texto proveniente de *microblogging* (redes sociais e comentários).
Lemmatizer	Classe	Implementa as operações que disponibilizam o lematizador.
DependencyParser	Classe	Implementa as operações que disponibilizam o analisador sintático de dependências.
Token	Classe	Define cada expressão que possui associado um significado. Por exemplo, identifica cada *emoticon* e a sua emoção ou cada palavra de pitês e a sua tradução respetiva.
TranslatedText	Classe	Define o texto final resultante da aplicação de algum processo de normalização ou tradução. Detém um conjunto de pontos de decisão que correspondem às palavras normalizadas ou traduzidas (utilizando para isso a classe *DecisionPoint*).
DecisionPoint	Classe	Define cada uma das alterações realizadas num determinado texto. Isto é, regista para cada palavra que foi alvo de alteração o respetivo índice no texto, palavra inicial e a correção efetuada.

Tabela 4.3: Elementos incluídos no *package* "*dataPreprocessing*" pertencente à biblioteca.

4.3.4 Módulo Extração de Termos e Expressões Multipalavra

Este módulo tem como objetivo efetuar a extração de termos pertencentes a uma determinada classe gramatical (nomes, verbos, adjetivos ou advérbios) bem como de expressões multipalavra.

Para a extração de termos foi utilizado o etiquetador gramatical do OpenNLP (disponibilizado no módulo de Pré-processamento, mencionado anteriormente) que permite determinar a categoria lexical de cada palavra e assim filtrar apenas as categorias pretendidas.

As expressões multipalavra foram identificadas através do uso de expressões regulares que possibilitam a obtenção de um conjunto de padrões sintáticos fre-

quentemente usados, cujas palavras fazem sentido que sejam agrupadas, como os visíveis nos seguintes exemplos:

*Universidade/**nome** de/**preposição** Coimbra/**nome***
*Bonito/**adjetivo** carro/**nome***
*20/**numeral** de/**preposição** Maio/**nome***
*Mário/**nome** Soares/**nome***

Desta forma, foi aplicado o seguinte processo:

1. Normalização do texto (tradução de abreviaturas, remoção de pontuação excessiva, etc.) recorrendo ao módulo de Pré-processamento dos Dados (referido na secção 4.3.3).

2. Segmentação das várias frases do texto.

3. Para cada frase é aplicado um processo de *tokenização*, sendo obtidos cada um dos seus termos individualmente.

4. Aplicação do etiquetador gramatical (também fornecido no módulo de Pré-processamento dos Dados) aos vários termos resultantes do processo de *tokenização*.

5. Obtenção da etiqueta ou categoria gramatical associada a cada termo.

6. Caso o objetivo seja obter todos os termos de uma determinada categoria gramatical é realizada uma filtragem dos *tokens* de acordo com a classe pretendida (nome, verbo, adjetivo ou advérbio).

7. Se o propósito é obter expressões multipalavra, são usadas expressões regulares de forma a identificar um conjunto de padrões sintáticos, tendo também em conta o resultado do etiquetador gramatical. É possível ver alguns dos padrões utilizados na Tabela 4.4.

Padrão Sintático	**Tradução**
(prop|n)*	Repetição de nomes próprios ou nomes comuns consecutivos.
num prp (prop|n)	Numeral imediatamente seguido de uma preposição e de um nome próprio ou comum.
(prop|n)* prp (prop|n)*	Nomes próprios ou comuns consecutivos que contenham uma preposição entre ambos.
(prop|n)*(conj-c|prp)*(prop|n)*	Nomes próprios ou comuns consecutivos que contenham conjunções coordenativas ou preposições entre ambos.

Tabela 4.4: Exemplos de padrões sintáticos identificados para extração de expressões multipalavra.

O módulo de Extração de Termos e Expressões Multipalavra encontra-se implementado no *package* denominado *termsExtraction*, sendo identificadas as suas classes na Tabela 4.5.

Nome	**Tipo**	**Descrição**
TermsExtraction	Classe	Implementa as operações responsáveis pela extração de termos.
MultiwordExtraction	Classe	Implementa as operações responsáveis pela extração de expressões multipalavra.

Tabela 4.5: Classes referentes ao *package* "*termsExtraction*" pertencente à biblioteca.

4.3.5 Módulo Extração de Tópicos

Este módulo tem como propósito analisar um conjunto de textos que não possuem associada qualquer categoria pré-definida. O objetivo é extrair um conjunto de tópicos, sendo que, cada tópico, consiste num conjunto de palavras que ocorrem frequentemente juntas e por essa razão encontram-se relacionadas conceptualmente, possibilitando assim a contextualização de cada uma das publicações. Cada distribuição de palavras representa um conceito semântico implícito nos dados. A título de exemplo, se um tópico incluir os termos *Internet*, *Google*, *software* e *Twitter* rapidamente é associado ao domínio tecnológico, enquanto, um tópico que inclua *universidade*, *pesquisa*, *prémio* e *descoberta* encontra-se mais relacionado com a temática de ciências e educação. Optou-se por usar o MALLET[49] uma vez que disponibiliza o modelo LDA[50] (Blei et al. (2003)) para esse fim, existindo já alguma experiência no uso desta ferramenta.

O processo aplicado na extração de tópicos é o seguinte:

1. Obtenção de um conjunto de textos de treino, ao qual é aplicado um pré-processamento, de forma a remover *stopwords*, URLs, símbolos de *hashtags* e menções, entre outras operações disponibilizadas no módulo de Pré-processamento. A ideia é remover grande parte do conteúdo causador de ruído na tarefa de extração de tópicos.

2. Aquando da submissão dos textos para treino, é igualmente necessário especificar o número de tópicos que se pretende extrair para geração de um modelo. O objetivo é depois usar este modelo para classificar individualmente cada uma das publicações a partir do qual se pretendem extrair tópicos.

[49] Disponível em *http://mallet.cs.umass.edu/*.
[50] LDA significa *Latent Dirichlet Allocation.*

3. Depois do modelo estar construído, ao submeter-se um texto, é obtido um vetor com os vários tópicos associados a esse texto, sendo atribuído a cada um dos tópicos uma probabilidade representativa do seu grau de pertença ao documento em questão.

    - De facto, cada texto presente detém uma probabilidade associada de pertencer a um ou mais tópicos, uma vez que, regra geral, um documento refere-se a vários tópicos em diferentes proporções. Por exemplo, um documento pode conter 40% do seu conteúdo sobre informática e 60% acerca de gestão.

    - Por sua vez, cada tópico possui também associado um vetor, não só com os vários termos que pertencem ao tópico, mas também com um valor que indica a relevância associada de cada termo para o tópico em questão. Por exemplo, um tópico relacionado com economia pode conter os termos "Economia", "Sector económico", "Banco", "Ações", sendo que o termo "Economia" pode conter uma relevância muito superior a "Ações".

O módulo de Extração de Tópicos encontra-se implementado no *package* denominado *topicExtraction*, sendo identificadas as suas classes na Tabela 4.6.

Nome	**Tipo**	**Descrição**
TopicExtractor	Classe	Implementa as operações responsáveis pela extração de tópicos.
TrainingModel	Classe	Representa o modelo e o conjunto de tópicos extraídos a partir das publicações submetidas inicialmente.
Topic	Classe	Define um tópico constituído por um conjunto de termos, cada um com uma determinada probabilidade associada.
ClassifiedText	Classe	Representa um texto já classificado com o respetivo conjunto de tópicos que lhe foram atribuídos.

Tabela 4.6: Classe referente ao *package* "*topicExtraction*" pertencente à biblioteca.

4.3.6 Módulo Extração de Entidades

Este módulo tem como propósito permitir a identificação de entidades mencionadas, classificando-as dentro de um conjunto de categorias pré-definidas (pessoas, organizações, localizações, entre outros).

Pelo facto de se pretender lidar com diferentes tipos de textos (desde texto menos estruturado proveniente de redes sociais, a texto mais estruturado proveniente de notícias online e blogues) e cada texto deter as suas particularidades,

decidiu-se então disponibilizar, no mesmo módulo, duas implementações distintas de extratores de entidades, cada uma vocacionada para um determinado tipo de texto. Ambas as implementações são especificadas nas seguintes subsecções.

Extrator de entidades vocacionado para texto estruturado (blogues e notícias online)

Para a implementação do extrator de entidades vocacionado para texto estruturado (blogues e notícias online) foi também usada a ferramenta MALLET, sendo, neste caso, utilizado o modelo CRF (Lafferty et al. (2001)) para classificação das entidades.

Para realização desta tarefa é necessário o fornecimento prévio de um conjunto de textos já etiquetados, para o treino e criação do modelo, que depois de construído, é usado para classificar novo conteúdo. Foram assim utilizadas as coleções douradas do HAREM (referidas na secção A.5.1), uma vez que disponibilizam um conjunto de textos (provenientes de notícias online, entrevistas, artigos de opinião, etc.) já anotados e validados por humanos, onde são identificadas dez categorias distintas de entidades: Abstração, Acontecimento, Coisa, Local, Obra, Organização, Pessoa, Tempo, Valor e Outro.

Apesar de nas coleções do HAREM se encontrarem definidas as 10 categorias de entidades referidas anteriormente, optou-se por, no treino, reduzir este número para apenas 7 categorias (Organização, Tempo, Local, Valor, Pessoa, Acontecimento e Outro), pelo facto de, não só se considerarem pouco relevantes as categorias Abstração, Obra e Coisa, mas também porque introduziam ambiguidade ao classificador. A título de exemplo, as expressões "testes estatísticos de Kruskal-Wallis", "Java", "Z100" e "Cães Pastores" foram todas classificadas, na coleção dourada do primeiro HAREM, como pertencentes a "Coisa", denotando conceitos totalmente diferentes que complicam significativamente a tarefa de reconhecimento de entidades. Consequentemente, para realização do treino, todas as entidades pertencentes a qualquer uma das categorias Abstração, Obra e Coisa, foram previamente associadas à categoria Outro.

O formato de dados aceite pelo MALLET para treino e consequente geração do modelo, é o seguinte:

$$\begin{array}{l} atributo_1\ atributo_2\ atributo_3\ \ldots\ atributo_n\ etiqueta \\ atributo_1\ atributo_2\ atributo_3\ \ldots\ atributo_n\ etiqueta \\ atributo_1\ atributo_2\ atributo_3\ \ldots\ atributo_n\ etiqueta \end{array}$$

Onde cada linha tem como objetivo representar cada uma das palavras provenientes do texto etiquetado e é representada através de um conjunto de atributos (também denominados por *features*), isto é, um conjunto de variáveis consideradas relevantes na construção do modelo. A *etiqueta* será substituída pela ca-

tegoria respetiva caso esta represente uma entidade, caso contrário será representada por um O (de Outro).

O conjunto de *features* que foi definido para ser usado na construção do modelo é o seguinte:

- Cada palavra do texto e o seu lema[51] respetivo, bem como o de todas as palavras circundantes, localizadas a uma distância máxima de 2 palavras em redor da atual.

- A etiqueta gramatical associada a cada palavra do texto e das palavras circundantes (também a uma distância máxima de 2 palavras em redor da atual).

- Características relativas a cada uma das palavras, semelhantes às descritas em (Bikel et al. (1999)). Desta forma, é associada uma *feature* a cada palavra para identificação das suas particularidades, através de um conjunto de expressões regulares. As expressões permitem identificar se a palavra contém apenas letras maiúsculas, minúsculas ou dígitos, se representa uma data, percentagem ou hora, entre outro tipo de detalhes.

- Identificação das palavras que contêm mais de três letras, uma vez que, regra geral, palavras constituídas por menos de três letras não representam entidades.

- Informação presente num conjunto de listas elaboradas manualmente com o intuito de auxiliar na identificação das entidades. Foram então criadas listas contendo locais (países e suas capitais), organizações (conjunto de empresas portuguesas e internacionais), nomes próprios e sobrenomes, bem como referências temporais (meses do ano e dias da semana).

A título de exemplo, para a seguinte frase retirada do HAREM:

O professor José Perdigão Dias da Silva é o regente da cadeira.

foram extraídas, para as 5 primeiras palavras, as seguintes *features*:

O *art o Professor n professor José prop josé allCaps lessOrEqualThreeLetters O*

Professor *n professor O art o José prop josé Perdigão v-fin perder initCap moreThreeLetters PESSOA*

[51] O lema representa a forma base de uma palavra, onde é ignorado tempo verbal, o género e plural. Por exemplo, o lema de "podado" e "podadas" é "podar".

José *prop josé O art o Professor n professor Perdigão v-fin perder Dias prop dias initCap moreThreeLetters nome PESSOA*

Perdigão *v-fin perder Professor n professor José prop josé Dias prop dias da v-pcp da initCap moreThreeLetters nome PESSOA*

Dias *prop dias José prop josé Perdigão v-fin perder da v-pcp da Silva prop silva initCap moreThreeWords nome PESSOA*

O módulo de Extração de Entidades encontra-se implementado no *package* denominado *NER*, sendo identificadas as suas classes na Tabela 4.7.

Nome	**Tipo**	**Descrição**
ExtractFeatures	Classe	Implementa as operações responsáveis pela extração dos atributos mais relevantes, que permitam representar cada uma das palavras de um texto.
LoadHAREM	Classe	Implementa as operações responsáveis pela extração dos textos e entidades respetivas provenientes das coleções douradas do HAREM. Faz também uso da classe *extractFeatures* para criação dos dados de treino e teste.
CRFTagger	Classe	Implementa as operações responsáveis pela extração de entidades usando o modelo CRF.
EntitiesFinder	Classe	Disponibiliza as operações do extrator de entidades vocacionado para texto estruturado a agentes externos.
Entity	Classe	Representa uma entidade e a sua respetiva categoria.
MalletTokens	Classe	Representa o formato de dados de entrada do MALLET, usado para treino e teste.
Result	Classe	Representa o resultado do processo de extração de entidades, registando para cada *token* do texto a sua categoria respetiva (fazendo para isso uso da classe *Entity*).

Tabela 4.7: Classes referentes ao *package* "*NER*" pertencente à biblioteca.

Extrator de entidades vocacionado para texto não estruturado (redes sociais)

Pelo facto do texto proveniente de redes sociais apresentar bastantes particularidades (devido sobretudo ao facto de ser pouco estruturado, possuir tamanho bastante reduzido, pouco contexto, etc.), dificulta bastante a realização de tarefas básicas (como a etiquetagem gramatical ou a *tokenização*) necessárias para a extração de *features* e, consequentemente, essenciais para usar o modelo

CRF anteriormente apresentado. Regra geral, em texto estruturado, é preservada a letra inicial maiúscula presente em entidades como "Pedro Passos Coelho" ou "Universidade de Coimbra", sendo algo fundamental para a correta identificação das mesmas. No entanto, na grande maioria das vezes, o mesmo não acontece no texto publicado em redes sociais (sobretudo nos comentários), revelando assim a inadequação na aplicação do modelo CRF a este tipo de texto.

Por estas razões, foi necessário o desenvolvimento de um extrator de entidades vocacionado apenas para lidar com as peculiaridades inerentes a este tipo de texto. Seguiu-se uma abordagem onde, recorrendo a um conjunto de listas elaboradas manualmente (contendo os nomes mais comuns de organizações, acontecimentos, localizações, pessoas e de expressões que referenciam tempo e valores), foi possível identificar as entidades, presentes num texto, pertencentes às seguintes categorias: Organização, Tempo, Local, Valor, Pessoa e Acontecimento. À categoria Outro pertencem todas as palavras ou expressões que contenham letra inicial maiúscula (à exceção do primeiro elemento de cada frase) e que não sejam incluídas em nenhuma das entidades mencionadas anteriormente. Apesar da adição desta categoria ser discutível, optou-se por mantê-la, uma vez que representa informação adicional relevante, como nomes de produtos ("Galaxy S5", "Android Wear"), categorias da publicação ("Vídeo", "Fotogaleria") ou ainda identificação de temas mais genéricos abordados ("Tablets", "Smartphones").

De forma a melhorar o processo de identificação de entidades, foram também definidas um conjunto de expressões (algumas visíveis na Tabela 4.8), permitindo reconhecer que, por exemplo, termos que detêm o formato *dd-mm-aaaa* pertencem à categoria Tempo, ou que, expressões que representem percentagens, pertencem a categoria Valor.

Expressão	**Exemplos**	**Categoria**
Horas	20:14 ou 20h	Tempo
Dígitos separados por travessão	1-2 ou 10-12	Valor
Dígitos separados por barra	12/04	Tempo
Percentagem	20%, 12,3% ou 12.33%	Valor
Número formatado	12,43 ou 34.3	Valor
Data	12/04/2014 ou 22-02-04	Tempo

Tabela 4.8: Exemplos de expressões identificadas no extrator de entidades vocacionado para texto não estruturado.

O uso de um etiquetador gramatical possibilitou a identificação de um conjunto de entidades mais complexas como "hospital de Lisboa" ou "mais de 2000", através da definição de regras do tipo "hospital de [*nome próprio*]" e "mais de [*numeral*]", que representam Organização e Valor, respetivamente. Outros exemplos de regras consideradas no extrator são visíveis na Tabela 4.9.

Regra	**Exemplos**	**Categoria**
Seleção (de|do|da) [*nome próprio*]	Seleção de Portugal	Organização
Museu (de|do|da) [*nome próprio*]	Museu do Porto	Organização
Presidente (de|do|da) [*nome próprio*]	Presidente da República	Pessoa
(Ministra|Ministro) (da|do|das|dos) [*nome próprio ou comum*]	Ministro das Finanças	Pessoa
Presidente [*adjetivo*]	Presidente italiano	Pessoa
Cerca de [*numeral*]	Cerca de 100	Valor

Tabela 4.9: Exemplos de regras consideradas no extrator de entidades vocacionado para texto não estruturado.

No extrator teve-se também atenção ao facto de expressões como "Ruy de Carvalho", que são etiquetadas da seguinte forma:

*Ruy/**Pessoa** de Carvalho/**Pessoa***

Deverem ser anotadas na sua totalidade como Pessoa. Ou seja, quando existe um termo intermédio (como "da", "de", "dos" ou "das"), que não representa nenhuma entidade, e é rodeado por palavras pertencentes a uma entidade definida, passa também a representar essa entidade. No caso do exemplo anterior, o resultado do processo de anotação passa então a ser:

*Ruy /**Pessoa** de/**Pessoa** Carvalho/**Pessoa***

Em algumas situações, a mesma entidade é anotada como pertencendo a duas categorias distintas, como o caso de "Fátima", que encontra-se presente, tanto na lista de locais, como na lista que contém nomes de pessoas. A forma definida para atribuir uma única categoria, a partir das várias identificadas, passa por determinar qual a anotação que apresenta a maior sequência contígua possível. Ou seja, realizando ambas as anotações:

*Maria/**PESSOA** Fátima/**PESSOA** Rodrigues/**PESSOA** foi contactada ...*

*Maria/**PESSOA** Fátima/**LOCAL** Rodrigues/**PESSOA** foi contactada ...*

É possível verificar que a primeira hipótese permite obter uma sequência Pessoa composta por 3 elementos, e por isso, a entidade "Maria Fátima Rodrigues" prevalece sobre a segunda hipótese, em que Fátima é etiquetada como Local.

O módulo de Extração de Entidades encontra-se implementado no *package* denominado *NERMicroblogging*, sendo identificadas as suas classes na Tabela 4.10.

Nome	Tipo	Descrição
NERMicroblogging	Classe	Implementa as operações responsáveis pela extração de entidades mencionadas em textos e comentários provenientes de redes sociais.
Entity	Classe	Representa uma entidade e a sua respetiva categoria.
Result	Classe	Representa o resultado do processo de extração de entidades, registando para cada *token* do texto a sua categoria respetiva (fazendo para isso uso da classe *Entity*).

Tabela 4.10: Classes referentes ao *package "NERMicroblogging"* pertencente à biblioteca.

DBPedia Spotlight

Um outro recurso também adicionado a este módulo foi a ferramenta DBpedia Spotlight (referido na secção A.6), que permite efetuar a anotação automática de menções referentes a recursos da DBpedia.

Embora este recurso não tenha sido explorado, poderá eventualmente ser útil para complementar o trabalho já desenvolvido, possibilitando, não só, uma contextualização das entidades através do *link* da Wikipédia respetivo (que confere acesso a mais informação complementar), como também permite a eventual anotação de outro tipo de entidades, que não tenham sido definidas como tal pelos extratores de entidades implementados neste módulo.

Este recurso encontra-se implementado no *package* denominado *NERSpotlight*, sendo identificadas as suas classes na Tabela 4.11.

Nome	Tipo	Descrição
SpotlightExtractor	Classe	Implementa as operações responsáveis pela extração de entidades a partir da DBPedia.
EntityDBPedia	Classe	Representa uma entidade, a sua categoria e URI respetivo.

Tabela 4.11: Classes referentes ao *package "NERSpotlight"* pertencente à biblioteca.

4.3.7 Módulo Análise de Sentimentos

Este módulo tem como propósito definir qual o sentimento ou a polaridade predominante associada a um texto, permitindo assim associar a uma publicação uma polaridade positiva, negativa ou neutra.

A abordagem utilizada para implementação deste módulo consistiu no seguinte processo:

1. Efetuar normalização do texto (remoção de pontuação excessiva e caracteres repetidos, tradução de abreviaturas, etc.) disponibilizado no módulo de Pré-processamento dos Dados (referido na secção 4.3.3).

2. Segmentar as várias frases de um texto.

3. Para cada frase aplicar um processo de *tokenização*, isto é, obter cada um dos seus termos, que pode ser representado por uma palavra, um *emoticon* ou simplesmente pontuação.

4. Remoção de elementos causadores de ruído, como URLs e os símbolos de *hashtags* (#) e menções (@) que muitas vezes representam entidades, como é possível visualizar no seguinte *tweet* retirado do TeK Sapo:

   *A #**PortugalTelecom** criou um nano-data center, uma montra de tecnologias e uma sala de estar em #**Lisboa**...*

   No caso específico das *hashtags* foi ainda realizado um processamento extra para que os nomes das entidades que contêm várias palavras possam ser corretamente identificados. Desta forma o resultado do processamento do *tweet* anterior é visível de seguida:

   *A **Portugal Telecom** criou um nano-data center, uma montra de tecnologias e uma sala de estar em **Lisboa**...*

5. Identificação de todas as palavras ou expressões que contenham uma opinião associada (conhecidas em inglês por *opinion words*), isto é, de todas as palavras que são geralmente utilizadas para expressar sentimentos positivos ou negativos. Por exemplo, "*bonito*", "*maravilhoso*" e "*bom*" tratam-se de palavras que expressam uma opinião positiva, enquanto "*mau*", "*pobre*" e "*terrível*" expressam uma opinião negativa. Além de palavras, existem também expressões idiomáticas como "*acertar na mouche*" ou "*acordar com os pés de fora*". Para realizar esta identificação foi usado o léxico de sentimentos SentilexPT (referido na secção A.5.1) que define um conjunto de *opinion words* em português, sendo que cada uma possui associado um inteiro representativo da sua polaridade (+1 para uma opinião positiva, -1 para negativa e 0 para neutra).

6. Identificar todos os advérbios de negação, como "*não*" ou "*jamais*" que invertem a polaridade de todas as *opinion words* afetadas. Como é possível visualizar no seguinte exemplo, o advérbio de negação "*não*" afeta a polaridade da *opinion word* "*bonito*", que passa de +1 (opinião positiva) a -1 (opinião negativa):

   *o telemóvel não é **bonito***

Assim, todas as *opinion words* que se seguem a um determinado inversor de polaridade, vão ter a sua polaridade invertida. O inversor de polaridade irá afetar todas as palavras até ao final de uma frase, ou até encontrar algum tipo de pontuação como uma vírgula ou ponto e vírgula.

7. Identificar todos os *emoticons* ao qual também é atribuída uma polaridade. A título de exemplo o *emoticon* ":)" possui associada uma polaridade positiva, e ":(" uma polaridade negativa.

8. A polaridade final atribuída à frase é calculada com base na soma de todas as *opinion words.* Por exemplo, na seguinte frase, a polaridade é neutra uma vez que a soma da polaridade de ambas as *opinion words* possui como resultado o valor zero:

   *O hardware é bom **(+1)**, mas o telemóvel é feio **(-1)**.*

   Mas na seguinte frase a polaridade já é positiva, uma vez que o resultado final é de +3:

   *Não gostei **(-1)** da banda sonora, mas de resto o filme foi maravilhoso **(+1)**!*
   *Adorei **(+1)**! Sem dúvida vai ser recomendado **(+1)** aos meus amigos. :) **(+1)***

Este recurso encontra-se implementado no *package* denominado *sentimentAnalysis*, sendo identificadas as suas classes na Tabela 4.12.

Nome	**Tipo**	**Descrição**
LoadSentiLexicon	Classe	Implementa as operações responsáveis por efetuar a leitura do léxico de sentimentos SentiLex-PT para uma estrutura em memória.
SentiAnalysis	Classe	Implementa as operações responsáveis pela extração da polaridade associada a um texto.
OpinionWord	Classe	Representa uma palavra detentora de opinião e a sua polaridade respetiva (positiva +1, negativa -1 ou neutra 0).
TextPolarity	Classe	Representa um texto que detém uma polaridade associada e um conjunto de *opinion words* (representadas através da classe *OpinionWord*).

Tabela 4.12: Classes referentes ao *package "sentimentAnalysis"* pertencente à biblioteca.

4.3.8 Módulo Extração de Triplos

Este módulo tem como propósito realizar a extração de triplos, na forma de recurso, propriedade e valor, a partir de texto que detém estrutura sintática. O

principal objetivo passa por reduzir a dimensionalidade dos textos, obtendo-se apenas as informações mais resumidas e relevantes de cada notícia.

A abordagem seguida para realizar a identificação de cada um dos três componentes de um triplo, passou pelo uso de um analisador sintático de dependências. Este analisador gera uma árvore sintática em que a sua estrutura é descrita unicamente em função das palavras e das relações existentes entre elas. A título de exemplo, para a frase "A Maria Joana gosta de ler", é possível visualizar a árvore sintática gerada pelo analisador de dependências na Figura 4.1.

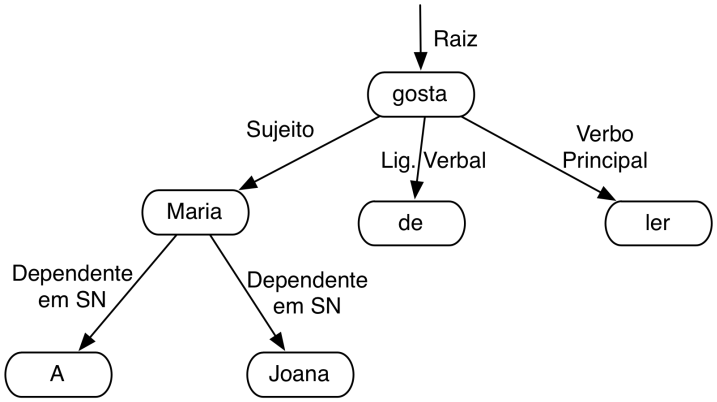

Figura 4.1: Exemplo de árvore gerada pelo analisador sintático de dependências.[52]

A partir das relações estabelecidas pelo analisador sintático de dependências e recorrendo também ao resultado proveniente do etiquetador gramatical e do módulo de Extração de Termos e Expressões Multipalavra (descrito na secção 4.3.4), é assim possível definir um conjunto de regras que permitem, para a maioria dos casos, determinar qual o sujeito, predicado e objeto numa frase. O processo aplicado foi o seguinte:

1. Pré-processamento do texto, removendo eventuais URLs, menções e *hashtags* existentes.

2. Segmentação do texto nas suas várias frases constituintes.

3. Para cada frase é aplicado um processo de *tokenização* e de extração da categoria gramatical associada a cada *token*, através do etiquetador gramatical disponibilizado no módulo de Pré-processamento.

4. Obtenção do *output* proveniente do analisador sintático de dependências, para cada uma das frases. Este *output* define uma estrutura descrita em

[52] As categorias gramaticais apresentadas no exemplo são as usadas no projeto Floresta, estando disponível a sua consulta em *http://beta.visl.sdu.dk/visl/pt/symbolset-floresta.html*.

termos das palavras e das relações existentes entre elas, que pode ser representada conceitualmente através de uma árvore.

5. Representação do *output* sob a forma de uma estrutura de dados em árvore, de forma a permitir a correta identificação dos vários elementos do triplo.

6. O predicado de uma frase é geralmente representado pela raiz da árvore, caso esta identifique um verbo (condição assegurada pelo etiquetador gramatical). No exemplo apresentado na Figura 4.1 o predicado é "gosta". No entanto, caso a raiz da árvore não contenha um verbo associado, não é realizada a extração dos outros elementos do triplo, uma vez que, se à partida o predicado estiver incorreto, consequentemente todos os outros elementos também estarão.

7. O sujeito é representado através da etiqueta Sujeito (definida pelo analisador sintático de dependências), caso esta seja constituída por nomes próprios ou comuns. No exemplo apresentado na Figura 4.1 o sujeito é "Maria Joana". Neste caso, e em todos os casos cujo sujeito seja composto por várias palavras, é necessário ter também em consideração as dependências diretas do sujeito.

8. O objeto é identificado através das dependências diretas provenientes da raiz da árvore (excluindo o sujeito). Apenas são consideradas as dependências que se iniciem por algum termo ou expressão multipalavra, ou seja, na prática, apenas são considerados os ramos que se iniciem por nomes comuns, próprios ou verbos (excluindo eventuais *stopwords*). Assim sendo, no exemplo anterior, é removida a dependência para a palavra "de" e considerado como objeto "ler". Importa ainda referir que, caso existam várias dependências diretas que representem opções igualmente válidas para o predicado, é escolhida a final com base na categoria gramatical atribuída pelo analisador sintático. Através da análise de vários exemplos, foi possível concluir que a etiqueta mais relevante e que por isso, obtém maior prioridade, é a "ACC", que representa o complemento verbal denominado objeto direto, ou seja, a palavra que completa o sentido do verbo. São igualmente relevantes etiquetas como "SC" e "OC" que representam o predicativo do sujeito e do objeto, respetivamente.

Este recurso encontra-se implementado no *package* denominado *tripleExtraction*, sendo identificadas as suas classes na Tabela 4.13.

Nome	Tipo	Descrição
DependencyTree	Classe	Implementa as operações responsáveis pela representação do resultado do analisador sintático de dependências na forma de árvore. Possui ainda um conjunto de operações que permitem a navegação e pesquisa sobre os nós da árvore.
TripleExtraction	Classe	Implementa as operações responsáveis pela extração de triplos a partir de um texto.
Node	Classe	Representa um nó da árvore.

Tabela 4.13: Classes referentes ao *package "tripleExtraction"* pertencente à biblioteca.

4.4 Detalhes de Implementação do Sistema

Esta secção visa apresentar detalhes de implementação relativos aos módulos do sistema, que se encontram presentes em cada uma das 3 camadas em que se subdivide:

- **Camada de Dados,** responsável pela gestão da base de dados de triplos do qual faz parte o módulo Gestor de Dados;
- **Camada de Negócio,** detém todas as regras de negócio do sistema e é constituída por um conjunto de módulos que fazem uso das funcionalidades providenciadas pela biblioteca para cumprirem os seus objetivos, nomeadamente o módulo Extração de Informação, o módulo de Pesquisa Semântica e o de Recomendação de Conteúdo.
- **Camada Pública,** responsável pela obtenção dos dados e disponibilização das funcionalidades concedidas pelo sistema a agentes externos, através do módulo denominado API.

Além dos detalhes acerca de cada módulo do sistema mencionado anteriormente, é ainda efetuada uma descrição da ontologia utilizada para persistência de todo o conteúdo utilizado pelo sistema.

4.4.1 Descrição da Ontologia

De forma a armazenar toda a informação relevante acerca do conteúdo das várias publicações e análises respetivas (geradas pelos módulos da biblioteca) foi criada uma ontologia. Esta ontologia é considerada de aplicação, uma vez que especializa conceitos de uma ontologia relacionada com partilha de conteúdo online (publicações de redes sociais, blogues e notícias online), de forma a possibilitar a

persistência do resultado das várias análises obtidas a partir da biblioteca de Processamento de Linguagem Natural (entidades, triplos, termos, etc.).

A ontologia é visível sob a forma de grafo na Figura 4.2 (gerado a partir do *plugin* Ontograf do Protege[53]) permitindo examinar mais facilmente a hierarquia de classes, bem como as várias ligações existentes entre elas. No Anexo D são explicadas em pormenor as várias classes e propriedades respetivas.

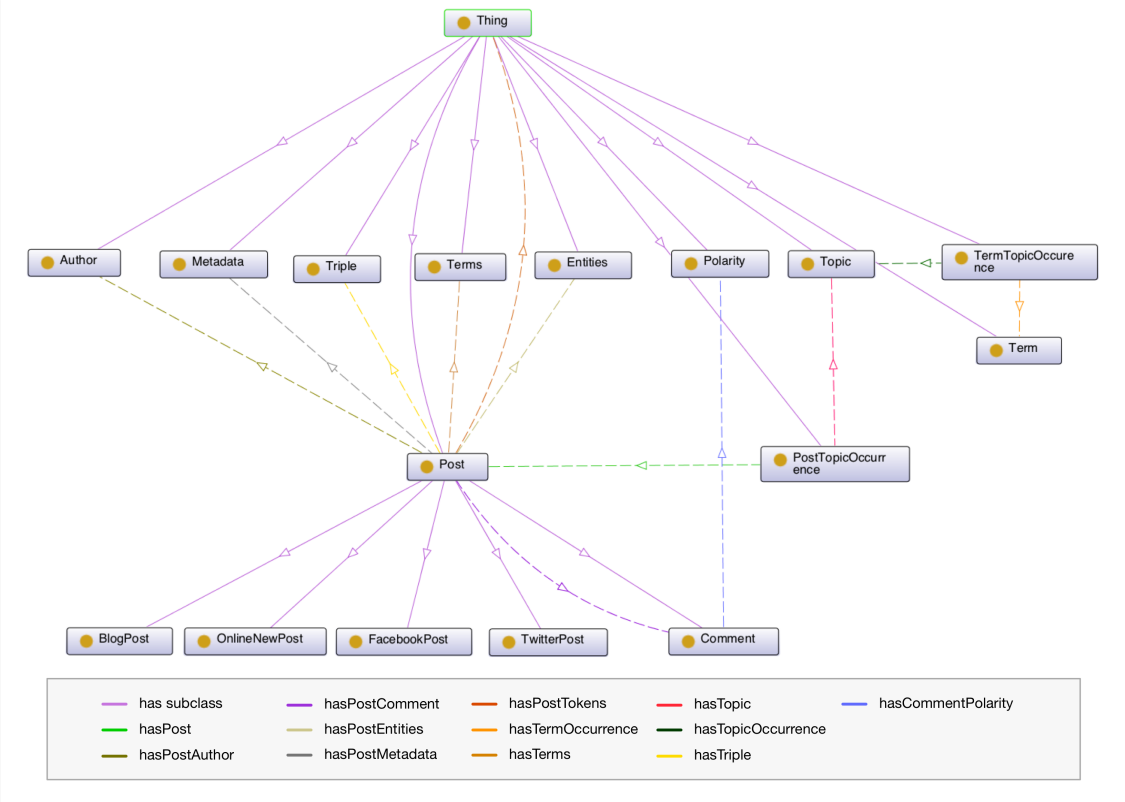

Figura 4.2: Visualização da ontologia através do *plugin Ontograf* do Protege.

4.4.2 Módulo Gestor de Dados

O módulo Gestor de Dados é o responsável pela gestão da base de dados de triplos Sesame, disponibilizando uma interface para que seja possível adicionar ou consultar os triplos existentes na BD, através da API do Sesame.

Para a adição de conteúdo à base de dados, recebe o resultado das várias análises providenciadas pela biblioteca para um conjunto pré-definido de publicações (as análises são disponibilizadas pelo módulo Extração de Informação), realizando um mapeamento da informação que se pretende persistir, nas várias classes e propriedades definidas previamente na ontologia, de forma a "automatizar" a inserção dos triplos na base de dados.

[53] Disponível em *http://protege.stanford.edu/*.

Este módulo foi desenvolvido com o intuito de, não só permitir a persistência do resultado obtido a partir dos vários módulos da biblioteca para um determinado *dataset*, como também permitir a consulta desses mesmos dados, de forma a obter os resultados a apresentar na pesquisa e recomendação semântica.

O módulo Gestor de Dados encontra-se implementado no *package* denominado *DataManager*, sendo as suas classes descritas na Tabela 4.14.

Nome	**Tipo**	**Descrição**
SesameAPI	Classe	Disponibiliza todas as operações para adição e consulta de dados na BD Sesame.
SesameManager	Classe	Faz uso das funcionalidades providenciadas pelo *SesameAPI* para possibilitar a inserção do conteúdo proveniente do *dataset*, bem como permitir a consulta de dados sobre o mesmo.

Tabela 4.14: Classes incluídas no *package* "*DataManager*" pertencente ao sistema.

4.4.3 Módulo Extração de Informação

Este módulo tem como objetivo obter toda a informação recolhida a partir das várias análises realizadas pela biblioteca para um determinado conjunto de publicações ou *dataset* definido previamente. Desta forma, pretende-se assim disponibilizar esta informação ao módulo Gestor de Dados (responsável pela comunicação com a BD), para que este conteúdo seja persistido na base de dados de triplos Sesame. Ao receber como parâmetro o texto da publicação, disponibiliza um conjunto de métodos, cada um responsável por invocar um módulo específico da biblioteca (por exemplo, *getEntities*, *getTopics*, *getPolarity*, etc.)

Neste módulo foi ainda necessário realizar uma diferenciação entre conteúdo estruturado e não estruturado, uma vez que os módulos a invocar são por vezes distintos (como o caso do extrator de entidades que disponibiliza duas implementações distintas para cada tipo de texto e o mesmo acontece com o *tokenizador*).

O módulo Extração de Informação encontra-se implementado no *package* denominado *AnalyzePosts*, sendo as suas classes descritas na Tabela 4.15.

Nome	**Tipo**	**Descrição**
ContentAnalyzer	Classe	Classe abstrata que define o conjunto de operações a serem implementadas pela classe *OnlineNewsAnalyser* e *SocialNetworkAnalyzer*.
OnlineNewsAnalyzer	Classe	Implementa todas as operações que permitem lidar com publicações que contenham texto sintaticamente estruturado (notícias online ou textos provenientes de blogues).

SocialNetworkAnalyzer	Classe	Implementa todas as operações que permitem lidar com publicações que contenham texto não estruturado sintaticamente , nomeadamente redes sociais.

Tabela 4.15: Classes incluídas no *package* "*AnalyzePosts*" pertencente ao sistema.

4.4.4 Módulo Recomendação de Conteúdo

Este módulo é responsável por disponibilizar um conjunto de recomendações com base nas características particulares de um determinado item. Isto é, para uma determinada publicação, é possível listar um conjunto de outras publicações que contenham associadas características consideradas relevantes (entidades, tópicos, termos, etc.), de forma a recomendar outros itens que possuem propriedades similares.

O módulo de Recomendação foi implementado de acordo com um processo de duas etapas:

1. Numa primeira fase, para uma determinada publicação, foram recolhidas todas as outras publicações que apresentavam algum tipo de interseção a nível de conteúdo, ou seja, que tinham em comum algum tipo de informação extraída através da biblioteca. Os tipos de informação considerados mais relevantes para a recomendação, foram os seguintes:

   - *Hashtags* e Menções;
   - Entidades (Organizações, Pessoa, Local e Outro);
   - Tópicos;
   - Termos (Nomes próprios, nomes comuns, expressões multipalavra, adjetivos e verbos).

   Nas entidades não foram consideradas as categorias Valor, Tempo e Acontecimento, uma vez que a maior parte das vezes conduziam à obtenção de dados que não eram relevantes. Por exemplo, o facto de outra publicação conter "2%" ou "20" nada garante que esteja de alguma forma relacionada. O mesmo acontece para alguns dos acontecimentos considerados, como "eleição" ou "jogo".

2. Depois de obtidas todas as publicações que possuem pelo menos uma interseção com um dos critérios definidos anteriormente, é atribuída uma pontuação a cada uma. O objetivo é permitir a realização de um *ranking* de resultados e mostrar em primeiro lugar as consideradas mais relevantes. Para isto foi utilizado o coeficiente de Jaccard (Real & Vargas (1996)), que mede a similaridade entre conjuntos de amostras finitas e é definido

como o tamanho da interseção dividido pelo tamanho da união dos conjuntos de amostras, ou seja:

$$Jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Desta maneira foi assim calculado o coeficiente de Jaccard entre o conjunto de dados da publicação original e cada uma das publicações a recomendar (ou seja, entre as entidades, termos, tópicos, *hashtags* e menções de ambos), de forma a calcular o grau de semelhança existente. Para obter o valor final é depois realizada uma média ponderada, uma vez que existem dados que são considerados mais relevantes que outros. De facto, *hashtags* e menções são as que detêm um peso superior, seguidas das entidades. Os termos e os tópicos possuem igual relevância.

O módulo de Recomendação encontra-se implementado no *package* denominado *Recommendation*, sendo as suas classes descritas na Tabela 4.16.

Nome	**Tipo**	**Descrição**
Recommendation	Classe	Disponibiliza todas as operações para, a partir de uma publicação, retornar um conjunto de recomendações.
JaccardCoefficient	Classe	Implementa todas as operações que permitem realizar o cálculo do coeficiente de Jaccard entre duas amostras.

Tabela 4.16: Classes incluídas no *package* "*Recommendation*" pertencente ao sistema.

4.4.5 Módulo Pesquisa Semântica

Este módulo tem como objetivo providenciar pesquisa de natureza semântica sobre as publicações, de forma a aperfeiçoar os resultados de pesquisa e mostrar ao utilizador os dados que lhe são mais relevantes e significativos.

O desenvolvimento deste módulo iniciou-se pela definição de qual a informação que seria mais relevante ter em conta em cada uma das publicações, para posteriormente efetuar pesquisas sob essa informação com base na *query* introduzida pelo utilizador. Foi então definido que, para a pesquisa semântica, os atributos mais relevantes são os seguintes:

- Entidades;
- Data de publicação;
- Menções e *Hashtags*;
- Autor ou fonte da publicação.

De seguida, quando o utilizador efetuar uma pesquisa, são verificadas quais as correspondências existentes entre a *query* e os dados existentes na base de dados de triplos, sendo dada maior relevância (e aparecendo em primeiro lugar) aqueles em que existe um número mais elevado de intersecções. Para que isto aconteça, a *query* introduzida pelo utilizador é convertida numa *query* SPARQL que tenta "perceber" qual o significado de cada termo presente na *query*, isto é, se referencia algum autor, entidade, data ou menção/*hashtag*, através da comparação com os dados presentes na BD de triplos.

De modo a melhorar os resultados da pesquisa, é ainda executado um processamento inicial de interpretação da *query*, que permite detetar um conjunto de padrões que influenciam os resultados apresentados, nomeadamente:

- Quando é detetada uma data é efetuada filtragem temporal, apresentando apenas as publicações pertencentes a essa data. Por exemplo: a *query* "20/05/2014 Google" apenas apresenta as publicações do dia 20 de Maio de 2014 que estejam relacionadas com a Google.

- Quando é detetado o nome de um autor ou fonte é igualmente realizada uma seleção dos dados, apresentando apenas as publicações pertencentes a esse autor ou fonte. Por exemplo: a *query* "Tek Sapo Google" apenas apresenta as publicações pertencentes à fonte Tek Sapo que referenciem a Google.

- Na presença de uma das seguintes palavras-chave: "*tweets com*", "*facebook com*", "*blogues com*" ou "*notícias online com*" é efetuada também uma escolha de resultados de acordo com a palavra imediatamente a seguir a cada uma delas, sendo nesse caso efetuada a filtragem de acordo com o tipo de publicação mencionada. Por exemplo: a *query* "Tweets com Google" filtra as publicações para apenas apresentar *tweets* que estejam relacionados com a Google.

- Quando na *query* de pesquisa é inserida uma referência explícita a uma *hashtag* usando o símbolo #, os resultados são filtrados de forma a apenas serem visíveis publicações que contenham essa mesma *hashtag*.

- É também possível realizar intersecções entre os vários padrões, como por exemplo, todas as publicações de um determinado autor que tenham uma *hashtag* específica. Por exemplo: a *query* "Tweets com Google 2013" filtra as publicações para apenas apresentar *tweets* que estejam relacionados com a Google, referentes ao ano 2013.

Além da pesquisa semântica, foi também implementada a pesquisa por palavra-chave (ou em inglês *keyword-based*), apenas com um intuito exploratório. Neste caso é realizada uma comparação entre a *query* inserida pelo utilizador e o texto proveniente de cada publicação, sendo somente apresentadas as publicações que contenham os termos especificados na pesquisa.

O módulo de Pesquisa Semântica encontra-se implementado no *package* denominado *Search*, sendo as suas classes descritas na Tabela 4.17.

Nome	**Tipo**	**Descrição**
SemanticSearch	Classe	Disponibiliza todas as operações referentes à pesquisa semântica, para que, a partir de uma determinada *query*, sejam obtidas um conjunto de publicações consideradas relevantes.
KeywordSearch	Classe	Disponibiliza todas as operações referentes à pesquisa por palavra-chave, para que, a partir de uma determinada *query*, sejam obtidas todas as publicações cujo texto contenha os seus termos.

Tabela 4.17: Classes incluídas no *package* "*Search*" pertencente ao sistema.

4.4.6 Módulo API

O módulo API é responsável, não só pela obtenção de dados que permitem alimentar o sistema e que, consequentemente, serão alvo de análise pelos vários módulos da biblioteca e persistidos na BD de triplos; mas também por disponibilizar as funcionalidades internas do sistema (nomeadamente a pesquisa e recomendação semântica) aos *Web services*.

O principal objetivo deste módulo passa por simplificar o processo de integração e por fornecer uma camada de abstração para com todos os módulos, sendo assim responsável pela instanciação e invocação das várias classes e funcionalidades, disponibilizando uma interface simples e de alto-nível a eventuais agentes externos que pretendam integrar as funcionalidades do sistema.

O módulo API encontra-se implementado no *package* denominado *SocialMiningAPI*, sendo as suas classes descritas na Tabela 4.18.

Nome	Tipo	Descrição
InputAPI	Classe	Disponibiliza todas as operações responsáveis por providenciar os dados ao sistema, que serão depois alvo de análise e persistência sob a forma de triplos.
OutputAPI	Classe	Disponibiliza todas as operações para providenciar as funcionalidades concedidas pelos módulos do sistema a agentes externos.

Tabela 4.18: Classes incluídas no *package* "*SocialMiningAPI*" pertencente ao sistema.

4.5 Detalhes de Implementação dos *Web Services*

Foram usados *Web services RESTful* (Richardson & Ruby (2008)), implementados através da *framework* RESTEasy[54] e que se encontram a ser disponibilizados por um servidor Apache Tomcat[55]. Os serviços disponibilizados retornam os resultados da pesquisa e recomendação semântica, devolvendo as várias publicações no formato JSON, para serem posteriormente visualizadas no cliente Web.

Optou-se por não apresentar os vários projetos e classes desenvolvidas na implementação dos *Web services*, uma vez que não possuem muita relevância no âmbito do trabalho desenvolvido e não apresentam um elevado grau de complexidade, realizam a disponibilização do sistema sob uma interface HTTP.

4.6 Detalhes de Implementação do Cliente Web

Foi desenvolvido um cliente Web, recorrendo a Javascript (Flanagan (2002)) e AJAX (Garrett (2005)), para obtenção e apresentação da informação proveniente da invocação dos *Web services*.

À semelhança dos *Web services* mencionados anteriormente, optou-se por não apresentar os vários projetos e classes desenvolvidas, dado o facto de também não ser relevante no âmbito da presente tese.

Na secção 4.6.1 é apresentado o resultado final da implementação do cliente Web, sendo visíveis as páginas Web referentes ao ecrã de listagem de resultados e ao de detalhes da publicação.

[54] Disponível em *http://resteasy.jboss.org/*.
[55] Disponível em *http://tomcat.apache.org/*.

4.6.1 Interface do Cliente Web

Nesta secção é apresentado o resultado final da implementação do cliente Web, sendo evidenciados os dois ecrãs com as respetivas áreas constituintes.

Página Web de Listagem de Resultados

Na Figura 4.3 é possível visualizar a interface de listagem de resultados associados a uma pesquisa efetuada pelo utilizador.

Figura 4.3: Página Web de listagem de resultados.

A página Web de listagem de resultados é constituída essencialmente por três áreas:

1. **Pesquisa Semântica ou Pesquisa por Palavra-Chave:** onde, depois de ser definido o tipo de pesquisa pretendido através do menu *dropdown*[56], o utilizador pode inserir os termos para efetuar uma pesquisa.

2. **Navegação facetada (ou filtros):** onde é apresentado um conjunto de facetas, isto é, um conjunto de termos/valores que são utilizados como filtros para que o utilizador possa selecionar apenas as publicações que pretende visualizar.

[56] Elemento da interface que permite ao utilizador escolher um valor de uma lista de opções que apenas é visível quando selecionado, sendo que, quando não é ativado, não se encontra visível para economizar espaço no ecrã.

3. **Visualização de resultados:** que permite visualizar os resultados da pesquisa efetuada.

Na área número 3 de "*Visualização de resultados*" não foi adicionada a zona expansível (visível nos protótipos) para simplificação da interface e consequente diminuição do tempo de desenvolvimento da mesma, visto que toda a informação pode ser consultada quando é selecionado o botão "*mais info*" presente em cada publicação.

Página Web de Detalhes da Publicação

Na Figura 4.4 é possível observar a interface referente aos detalhes de uma publicação, visível sempre que o utilizador seleciona o *link* "*mais info*", presente em cada uma das publicações listadas no protótipo anterior.

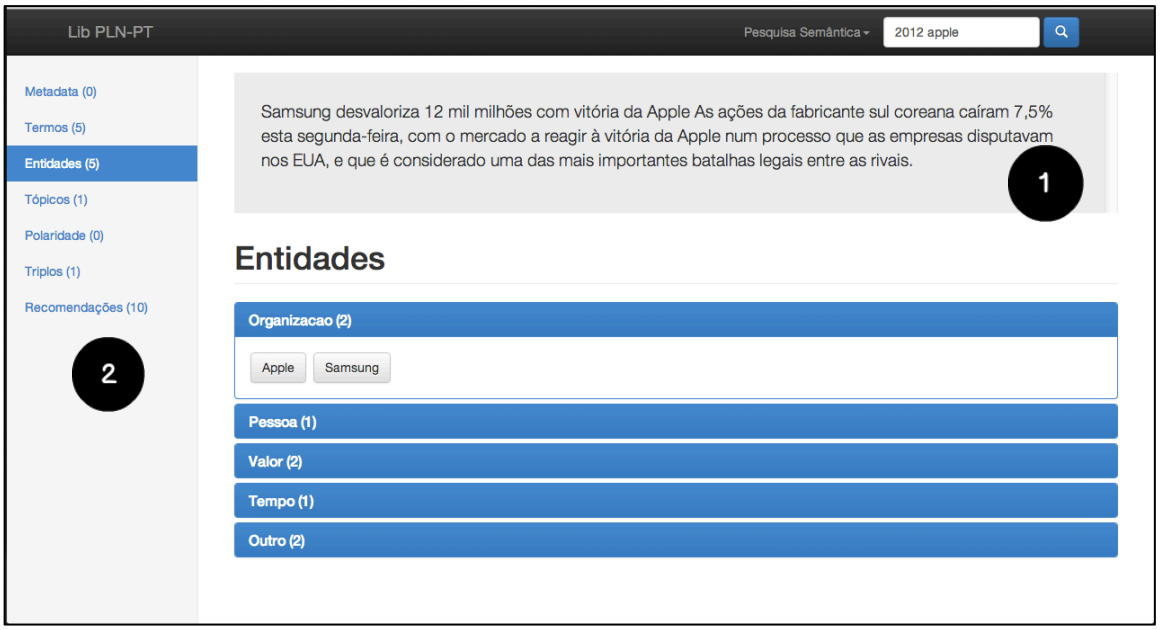

Figura 4.4: Página Web de detalhes de publicação.

A página Web evidencia a existência de duas áreas:

1. **Texto da publicação:** onde o utilizador pode visualizar o texto referente à publicação que selecionou previamente.

2. **Visualização de informação:** onde é conferido o acesso a todo o tipo de informação extraída a partir da publicação (nomeadamente entidades, tópicos, recomendações, entre outras).

Capítulo 5

Experimentação

Este capítulo tem como objetivo apresentar a experimentação realizada para avaliação do desempenho dos seguintes módulos da biblioteca:

- Extrator de Entidades vocacionado para texto estruturado (blogues e notícias online);
- Extrator de Entidades vocacionado para texto não estruturado (redes sociais);
- Analisador de Sentimentos;
- Extrator de Triplos;
- Extrator de Tópicos.

Para cada módulo é descrito o teste efetuado, apresentados os resultados e elaborada uma análise para interpretação dos mesmos.

Por fim, são ainda apresentados os resultados dos testes unitários e de validação de requisitos, de forma a aferir o correto funcionamento do sistema.

5.1 Módulo Extração de Entidades - Texto Estruturado

Esta secção apresenta os testes realizados ao extrator de entidades vocacionado para texto estruturado (blogues ou notícias online), pertencente ao módulo de Extração de Entidades descrito na secção 4.3.6.

O objetivo passa por avaliar os resultados obtidos por este extrator ao classificar sete categorias distintas de entidades: Acontecimento, Local, Organização, Pessoa, Tempo, Valor e Outro.

5.1.1 Descrição dos Testes Realizados

Os dados usados para treino e teste do modelo foram as coleções douradas do HAREM, que depois de terem sido alvo de análise, verificou-se a existência de diferenças significativas no processo de anotação das coleções pertencentes a cada

um dos eventos. Optou-se então por realizar testes complementares onde existe uma separação entre ambas, de forma a verificar se este processo diferenciado de anotação exercia influência nos resultados obtidos. A título de exemplo, expressões como "No terceiro trimestre" e "em 1935" foram definidas como pertencendo à categoria Tempo na segunda coleção dourada, mas o mesmo não se verifica na coleção anterior onde, das expressões referidas anteriormente, apenas "1935" é considerado como Tempo. Desta forma, foram assim realizados 3 testes distintos – o primeiro usando ambas as coleções para treino e teste e os dois restantes usando cada coleção em separado.

Para avaliação do desempenho do modelo treinado em cada um dos testes, foi usado um dos métodos mais popularmente utilizados – a validação cruzada (Refaeilzadeh, Tang & Liu (2009)) com 10 repetições (ou seja, validação cruzada 10-Fold) sendo, a cada repetição, realizada a divisão dos dados em 10 subconjuntos iguais, usando 1 para teste e os restantes para treino.

Teste 1 (Todas as Coleções HAREM)

No primeiro teste foram usadas todas as coleções do HAREM (pertencentes ao primeiro e segundo evento) para treinar e testar o modelo CRF (Lafferty et al. (2001)). As coleções reúnem um total de 416 textos, compostos por 11,378 instâncias (sendo cada instância representada por cada uma das frases constituintes de um texto). Na Tabela 5.1 é possível visualizar, para cada categoria, o número total de exemplos disponibilizados em ambas as coleções consideradas.

Categorias	**Coleções HAREM 1º Evento**	**Coleções HAREM 2º Evento**	**Total**
Organização	1,498	1,199	2,697
Outro	1,358	1,259	2,617
Tempo	798	1,423	2,221
Local	2,049	1,493	3,542
Valor	803	463	1,266
Pessoa	1,820	2,396	4,216
Acontecimento	183	339	522

Tabela 5.1: Teste 1 – Número total de exemplos pertencentes a cada categoria em ambas as coleções douradas HAREM.

Teste 2 (Coleções 1º Evento HAREM)

Neste teste foram utilizadas apenas as duas coleções referentes ao primeiro evento do HAREM, ou seja, a coleção dourada do Primeiro HAREM e a do miniHAREM. O total de textos considerados foi 257, compostos por 7,063 instâncias, sendo o número de exemplos de cada categoria apresentado na Tabela 5.2.

Categorias	1º HAREM	Mini HAREM	Total
Organização	875	623	1,498
Outro	769	589	1,358
Tempo	440	358	798
Local	1,232	817	2,049
Valor	468	335	803
Pessoa	1,018	802	1,820
Acontecimento	127	56	183

Tabela 5.2: Teste 2 – Número total de exemplos pertencentes a cada categoria nas coleções douradas HAREM do 1º evento.

Teste 3 (Coleções 2º Evento HAREM)

No terceiro teste foram utilizadas apenas as duas coleções referentes ao segundo evento do HAREM - coleção dourada do Segundo HAREM e a coleção TEMPO. Neste caso, o total de textos considerados foi 159, compostos por 4,315 instâncias. O número de exemplos considerados para cada categoria é apresentado na Tabela 5.3.

Categorias	2º HAREM	CD TEMPO	Total
Organização	953	246	1,199
Outro	1,094	165	1,259
Tempo	1,189	234	1,423
Local	1,240	253	1,493
Valor	352	111	463
Pessoa	2,048	348	2,396
Acontecimento	302	37	339

Tabela 5.3: Teste 3 - Número total de exemplos pertencentes a cada categoria nas coleções douradas HAREM do 2º evento.

5.1.2 Resultados

As medidas de avaliação aplicadas consistem na verificação dos resultados obtidos pelo classificador, face a resultados que já estejam previamente confirmados como corretos (neste caso os textos etiquetados do HAREM). Esta verificação é feita recorrendo à medida precisão[57] (número de resultados corretos dividido pelo número de resultados selecionados) e à abrangência[58] (número de

[57] O termo utilizado em inglês é *precision.*

[58] O termo utilizado em inglês é *recall.*

resultados corretos dividido pelo número de resultados já confirmados como corretos, ou seja, os resultados que deveriam ter sido retornados).

Associada a estes conceitos existe também a medida-F[59], uma medida que combina a precisão e a abrangência para calcular o seu valor, podendo ser interpretada como uma média ponderada de ambas. O valor da medida-F varia também entre 0 e 1, sendo calculada através da seguinte fórmula:

$$medida - F = 2 \times \frac{precisão \times abrangência}{precisão + abrangência}$$

Na Tabela 5.4, Tabela 5.5 e Tabela 5.6 são apresentadas várias medidas recolhidas referentes à média das 10 repetições executadas para o Teste 1, Teste 2 e Teste 3, respetivamente.

Categorias	**Precisão**	**Abrangência**	**Medida-F**
O (não entidade)	0.9723	0.9829	0.9776
Organização	0.6788	0.6330	0.6530
Outro	0.5431	0.5264	0.5337
Tempo	0.7733	0.5963	0.6727
Local	0.7007	0.6764	0.6876
Valor	0.7807	0.6787	0.7246
Pessoa	0.6889	0.7315	0.7087
Acontecimento	0.6338	0.4092	0.4898
Média	**0.7215**	**0.6543**	**0.6810**
Desvio Padrão	**0.1183**	**0.1562**	**0.1368**

Tabela 5.4: Teste 1 - Medidas de avaliação do classificador CRF, usando todas as coleções HAREM.

[59] O termo utilizado em inglês é *f-measure*.

Categorias	Precisão	Abrangência	Medida-F
O (não entidade)	0.9804	0.9868	0.9836
Organização	0.6806	0.6531	0.6652
Outro	0.4980	0.4511	0.4711
Tempo	0.8431	0.8348	0.8384
Local	0.7425	0.7292	0.7343
Valor	0.7944	0.6739	0.7252
Pessoa	0.6680	0.6916	0.6788
Acontecimento	0.4962	0.2815	0.3442
Média	**0.7129**	**0.6628**	**0.6801**
Desvio Padrão	**0.1550**	**0.2032**	**0.1867**

Tabela 5.5: Teste 2 - Medidas de avaliação do classificador CRF, usando as coleções do 1º evento HAREM.

Categorias	Precisão	Abrangência	Medida-F
O (não entidade)	0.9738	0.9835	0.9786
Organização	0.6529	0.6556	0.6526
Outro	0.7333	0.6780	0.7023
Tempo	0.7397	0.5430	0.6189
Local	0.8745	0.7795	0.8228
Valor	0.7433	0.8140	0.7761
Pessoa	0.7449	0.6640	0.7009
Acontecimento	0.8069	0.6791	0.7326
Média	**0.7837**	**0.7246**	**0.7481**
Desvio Padrão	**0.0933**	**0.1243**	**0.1060**

Tabela 5.6: Teste 3 – Medidas de avaliação do classificador CRF, usando as coleções do 2º evento HAREM.

5.1.3 Análise de Resultados

Com base numa análise prévia das coleções do HAREM e dos ficheiros usados para treino do modelo, foi possível verificar que algumas das entidades não se encontravam corretamente identificadas. Isto aconteceu devido a erros de formatação existentes em alguns textos provenientes das coleções douradas, como o caso de "C o m p r a s ." ou "D.Teresa:-", que tiveram influência no processo de *tokenização*, consequentemente conduzindo a uma incorreta etiquetagem gramatical no ficheiro de treino. Também o processo de divisão dos textos nas suas frases constituintes não foi muitas vezes realizado corretamente, uma vez que, por exemplo, neste excerto de uma entrevista retirada do HAREM:

A. H.: Há bons exemplos de empresas...

Foi realizada uma divisão em duas frases, a primeira contendo apenas "A." e a segunda "H.: Há bons exemplos de empresas...". Na prática, uma vez que cada frase representa uma instância de treino, existirá então uma instância que apenas possui "A." (representando a categoria Pessoa), enquanto deveria ter sido considerado como Pessoa o termo "A. H." na sua totalidade.

Das várias medidas recolhidas nos testes realizados foi possível verificar que, de facto, os dados usados para treino possuem influência no resultado final, existindo uma diferenciação clara nos valores do teste 2 para o teste 3, mostrando desta forma o impacto existente no processo diferenciado de anotação das duas coleções do HAREM. Assim, um dos possíveis melhoramentos a realizar, seria precisamente uniformizar a anotação utilizada em ambas as versões utilizadas, de forma a conseguir melhorar os valores apresentados no teste 1. O facto de, tanto no teste 2 como no teste 3, apenas ser usada uma das duas coleções para gerar o modelo, tem algumas consequências negativas, devido sobretudo ao número de instâncias de treino diminuir e o número de exemplos pertencentes a algumas categorias ser consideravelmente inferior a outros. A título de exemplo, no teste 2, o número de exemplos pertencentes à categoria Acontecimento é apenas de 183, enquanto no teste 3 são 339, ajudando assim a perceber o porquê dos resultados tão baixos alcançados pelo teste 2 nesta categoria.

Uma análise mais aprofundada ao conteúdo das várias coleções revelou também a existência de palavras que mediante o contexto são anotadas de forma distinta, como o caso da palavra "Portugal" que é etiquetada tanto como Pessoa, Organização ou Local, contradizendo assim as listas elaboradas (usadas como *feature*) onde os vários países e capitais são definidos como pertencendo unicamente à categoria Local. Este é claramente outro ponto que necessita de melhoramento, ou pela uniformização destes casos nas várias coleções, definindo-se a categoria mais provável, ou então usando o contexto presente na frase para determinar que, por exemplo, caso se refira a um evento desportivo, Portugal deva ser considerado como uma Organização.

Um outro ponto que também pode ser melhorado diz respeito ao número de entidades presentes nas listas criadas, que são ainda consideravelmente baixas (a lista de maior dimensão contém cerca de 1,000 entidades) e que auxiliam bastante no processo de categorização.

5.2 Módulo Extração de Entidades - Texto Não Estruturado (Redes Sociais)

Esta secção apresenta o teste realizado ao extrator de entidades vocacionado para texto que não apresenta estrutura sintática, ou seja, para o texto proveniente sobretudo de redes sociais e que faz parte do módulo de Extração de Entidades descrito na secção 4.3.6.

O objetivo passa por avaliar os resultados obtidos por este extrator ao classificar sete categorias distintas de entidades: Acontecimento, Local, Organização, Pessoa, Tempo, Valor e Outro.

5.2.1 Descrição do Teste Realizado

Uma vez que o foco deste extrator de entidades incide sobre publicações provenientes de redes sociais (texto não estruturado) e não tendo sido possível obter um conjunto de exemplos de teste já etiquetados, revelou-se necessário efetuar uma coleta e anotação manual de dados para avaliação do desempenho deste módulo.

Desta forma, de uma população de 10 mil publicações provenientes de redes sociais (constituída por 5 mil publicações provenientes do Twitter e 5 mil do Facebook), foram assim etiquetadas manualmente um conjunto de 1,000 publicações pertencentes às mais variadas áreas, de forma a abranger um largo conjunto de domínios - desde notícias mais generalistas provenientes do Público, Lusa, Expresso, Jornal Sol, etc. a notícias de índole tecnológica, desportiva, económica, de entretenimento, entre outras.

O método utilizado para calcular a amostra baseou-se na tabela para estatísticas de (Arkin & Colton (1950)). De acordo com esta tabela[60], a seleção aleatória de uma amostra composta por mil indivíduos de um universo de 10 mil, permite alcançar uma margem de erro na casa dos 3%, com um coeficiente de confiança de 95.5%.

A etiquetagem foi realizada por dois anotadores, tendo sido previamente acordado um guião entre ambos, com o objetivo de uniformizar o processo de anotação realizado, através, por exemplo, da definição de quais as expressões temporais válidas ou do domínio dos acontecimentos considerados. As principais discordâncias detetadas dizem respeito às seguintes categorias:

[60] É possível visualizar a tabela em *http://www.mbi.com.br/MBI/biblioteca/tutoriais/amostragem/.*

- **Pessoa:** Expressões como "CEO" ou "Vice-Presidente" foram considerados por um anotador como pertencendo à classe Pessoa, enquanto o outro anotador não as considerou como entidades.

- **Organização e Local:** Por vezes não existiu consenso na atribuição destas categorias, uma vez um dos anotadores considerou expressões como "escola de Mirandela" ou "prisão de Lousiana" como sendo Local e o outro como Organização.

- **Acontecimento:** Algumas discrepâncias foram detetadas ao nível da categoria Acontecimento, sendo que um dos anotadores definiu termos como "fusão" ou "liga" como sendo acontecimento, enquanto o outro não.

- **Outro:** As divergências encontradas nesta categoria têm que ver sobretudo com a diferenciação gerada entre Organização e Outro (por exemplo, Skype[61] e MSN[62] foram considerados como Organização por um elemento e pelo outro como produtos, sendo por isso associados à entidade Outro).

O cálculo da concordância, isto é, do número de classificações iguais dadas por ambos os anotadores no universo total de elementos pertencentes a cada categoria, é apresentado na Tabela 5.7. Na Tabela 5.8 é ainda possível visualizar o número total de anotações contabilizadas, por cada anotador, para cada uma das categorias.

Categorias	**Concordância**
Organização	0.955
Outro	0.980
Tempo	1
Local	0.961
Valor	1
Pessoa	0.972
Acontecimento	0.968

Tabela 5.7: Cálculo da concordância existente entre as duas anotações.

[61] Disponível em *http://www.skype.com/pt/*.
[62] Disponível em *http://pt.msn.com/*.

Categorias	Número de Anotações Anotador 1	Número de Anotações Anotador 2
Organização	587	593
Outro	331	333
Tempo	197	197
Local	428	421
Valor	292	292
Pessoa	347	352
Acontecimento	247	248

Tabela 5.8: Número total de anotações realizadas por cada anotador em cada categoria.

5.2.2 Resultados

Na Tabela 5.9 é possível visualizar as várias métricas recolhidas referentes ao teste realizado, sendo que cada valor é dado pela média do resultado obtido usando cada uma das anotações.

Categorias	Precisão	Abrangência	Medida-F
O (não entidade)	0.9770	0.9684	0.9726
Organização	0.8954	0.5453	0.6778
Outro	0.3397	0.8549	0.4862
Tempo	0.7267	0.9225	0.8130
Local	0.7993	0.6500	0.7170
Valor	0.8276	0.9190	0.8709
Pessoa	0.8987	0.3810	0.5351
Acontecimento	0.7617	0.5566	0.6443
Média	**0.7782**	**0.7247**	**0.7145**
Desvio Padrão	**0.1823**	**0.2054**	**0.1547**

Tabela 5.9: Medidas de avaliação do extrator de entidades vocacionado para texto não estruturado.

5.2.3 Análise de Resultados

Uma das categorias que possui os valores mais baixos é a categoria Outro, sendo algo previsível, dado o facto do extrator de entidades ter definido como regra que todas as palavras que não possuem nenhuma entidade associada e que contêm letra inicial maiúscula são consideradas como pertencentes a esta categoria (à exceção da primeira palavra de cada frase). Pretende-se desta forma considerar classes genéricas como “Smartphones”, “Tablets” ou produtos como “Samsung Galaxy”. Na prática, todas as entidades que se iniciem por letra maiúscula

(como nomes de pessoas e países) mas que não se encontrem definidas nas várias listas elaboradas, são reconhecidas como pertencentes à classe Outro, o que provoca valores tão baixos na métrica precisão (dado o número elevado de entidades que são incorretamente classificadas como Outro por não se encontrarem presentes nas listas). Podia ter-se removido esta categoria para evitar este tipo de situações, no entanto, pelo facto de na grande parte das vezes acrescentar informação útil às publicações recolhidas (que de outra forma não seria detetada), optou-se por manter esta regra.

A categoria Pessoa apresenta também valores relativamente baixos na métrica abrangência e isso deve-se, em grande parte, ao facto da lista elaborada conter um número ainda limitado de exemplos. Apenas foram reunidos os nomes mais populares utilizados em diversas regiões, nomeadamente África, América, Europa, Ásia, etc. pelo que, apesar das poucas expressões que foram identificados como pertencentes à categoria Pessoa estarem de facto corretas (daí a elevada precisão), o número total é ainda relativamente escasso, gerando uma abrangência tão baixa.

Um dos principais problemas detetados no uso deste extrator tem que ver com o facto de não existir nenhum tipo de desambiguação no sentido das palavras, fazendo com que, quando surjam palavras como "Fátima", o sistema não seja capaz de perceber que se refere a um local (por exemplo, Santuário de Fátima) ou a uma pessoa (por exemplo, Maria de Fátima) e o mesmo acontece com Barcelona que tanto pode representar um local (cidade), como uma organização (equipa de futebol). Este tipo de situações ocorre com bastante frequência e apesar de ser possível notificar o utilizador das duas possíveis interpretações distintas (que acontece quando a mesma palavra encontra-se definida em vários dicionários), caso se pretenda automatizar o sistema, de forma a obter apenas uma interpretação possível, a mesma terá de ser escolhida com base na que apresenta a maior sequência possível, ou então de forma aleatória. Seria igualmente interessante (e um possível melhoramento a realizar) possibilitar a escolha do sentido mais provável tendo em conta o contexto, isto é, sendo uma publicação acerca de futebol então Barcelona seria anotado como Organização, ou seja, permitir a desambiguação do sentido das palavras.

O desempenho deste módulo está também bastante relacionado com a capacidade de reconhecimento de possíveis variações presentes em nomes de entidades. A título de exemplo, é muito frequente a utilização da palavra "face" como referência a "facebook" ou "scp" para representar "sporting", pelo que a existência de um dicionário com um conjunto de abreviaturas de entidades e respetivo significado, possibilitaria também melhorar resultados.

De uma forma geral, apesar dos problemas detetados e eventuais melhoramentos a realizar, os valores revelaram-se satisfatórios dado o facto do número de entidades definidas nas várias listas elaboradas ser relativamente reduzido, contando a maior lista com cerca de 1,200 elementos. Tendo em conta o variado

domínio das publicações, apenas foram consideradas as entidades mais populares e genéricas, bem como as expressões regulares e regras definidas são ainda igualmente escassas.

5.3 Módulo Análise de Sentimentos

Esta secção apresenta o teste realizado ao analisador de polaridade, que faz parte do módulo de Análise de Sentimentos descrito na secção 4.3.7.

O objetivo passa por avaliar os resultados obtidos por este analisador ao atribuir a cada publicação uma polaridade associada que pode ser positiva, negativa ou neutra.

5.3.1 Descrição do Teste Realizado

Para a realização de testes neste módulo foi usado um conjunto de publicações já etiquetadas com uma polaridade associada (positiva, negativa e neutra) disponibilizadas pelo Laboratório KIS (Knowledge and Intelligent Systems Lab), pertencente ao CISUC. Foi assim possível obter um número considerável de dados para teste através deste *dataset* constituído por 7,892 textos, provenientes sobretudo de comentários do Facebook, Twitter e de vários fóruns online, como o fórum de esclarecimento de dúvidas da Vodafone[63].

O número total de publicações anotadas, para cada uma das três polaridades definidas, pode ser visualizado na Tabela 5.10.

Polaridade	**Número Publicações**
Positiva	2,058
Negativa	1,278
Neutra	4,556
Total	**7,892**

Tabela 5.10: Número total de exemplos considerados no teste do analisador de polaridade.

5.3.2 Resultados

Na Tabela 5.11 é possível visualizar as várias medidas recolhidas para o teste realizado.

[63] Disponível em *http://forum.vodafone.pt/*.

Polaridade	Precisão	Abrangência	Medida-F
Positiva	0.5232	0.6201	0.5675
Negativa	0.3950	0.4343	0.4137
Neutra	0.7171	0.6372	0.6748
Média	**0.5451**	**0.5639**	**0.5520**
Desvio Padrão	**0.1324**	**0.0919**	**0.1072**

Tabela 5.11: Medidas de avaliação do analisador de polaridade.

5.3.3 Análise de Resultados

O módulo de Análise de Sentimentos foi consideravelmente complexo de testar porque o próprio processo de atribuição de uma polaridade a uma publicação não é de todo trivial e por vezes não é fácil obter um consenso. A título de exemplo, a seguinte frase:

"AS MELHORAS EUSÉBIO TU ÉS SÍMBOLO DE PORTUGAL!"

É difícil de anotar visto, por um lado, poder ser considerada como tendo polaridade positiva porque o autor demonstra grande apreço ao Eusébio, mas por outro lado, o facto de desejar as melhoras também pode ser considerado como algo negativo, pelo facto do Eusébio se encontrar doente. O mesmo é visível em frases como:

O atleta abalroou a concorrência.

Que, se por um lado detém polaridade positiva para o atleta, representa algo negativo para a concorrência. Estes exemplos demonstram assim que, mediante o contexto, a frase pode adquirir diferentes polaridades, tornando o próprio processo de atribuição de uma polaridade muitas vezes difícil e não havendo propriamente um consenso, estando apenas ao critério de quem está a realizar a anotação.

A identificação da polaridade negativa foi a que obteve piores resultados, isto porque, muitas vezes é representado um sentimento de desaprovação sem recorrer a palavras que contenham obrigatoriamente polaridade negativa associada, sendo até usada a ironia ou sarcasmo para este fim. Desta forma, em algumas situações, torna-se bastante difícil conseguir determinar a polaridade negativa associada a algumas publicações, como as visíveis de seguida:

grande serviço público... ou não!!!

Estamos na era do vale tudo sem dúvida.

Enfim...

Tu precisas de um cirurgião plástico e não de um médico.

Um outro ponto negativo detetado nos testes realizados ao módulo, diz respeito à forma como é realizado o cálculo para atribuir um valor de polaridade a uma frase que contém várias opiniões. Este cálculo baseia-se na soma de todas as polaridades associadas às *opinion words* identificadas, isto é, a todas as palavras que expressam algum tipo de sentimento. Assim, na seguinte frase:

Pode ser verdade mas não acredito !! A Susana é muito FALSA !!O Marco é muito novinho e com um coração do tamanho do mundo !! Ela que não o faça SOFRER !!!

São identificadas duas expressões com conotação positiva (*não o faça sofrer* e *coração do tamanho do mundo*) e duas expressões com conotação negativa (*não acredito* e *falsa*), acabando assim por a frase adquirir polaridade neutra. Esta metodologia acaba por nem sempre fazer muito sentido, uma vez que, por exemplo, a frase referida anteriormente, tem claramente associada uma conotação negativa relativamente à Susana e o facto de o somatório ter como resultado 0, não significa necessariamente que a polaridade seja neutra (ou seja, que não se encontra a ser expressa nenhuma opinião). Uma forma de melhorar este aspeto e consequentemente também enriquecer o módulo seria, em vez de apenas atribuir uma polaridade final, selecionar na frase as várias expressões que denotam algum tipo de sentimento e assinalá-las com a polaridade respetiva, deixando assim de existir as frases com polaridade neutra (ou então considerando com polaridade neutra aquelas que não possuem qualquer tipo de expressão anotada).

Apesar dos valores obtidos na experimentação não serem elevados, o mesmo acaba por ser expectável devido ao grande domínio de textos considerados no teste. Quando este módulo é aplicado a um domínio mais restrito, é possível, à partida, ter em conta um conjunto de outros fatores como expressões ou vocabulário habitual que auxilia bastante neste processo. A título de exemplo, no domínio político, é habitual visualizar nos comentários o uso de termos pejorativos como "pinócrates" ou calão, pelo que a criação de listas contendo este tipo de vocabulário permitiria "intensificar" o valor da polaridade negativa, sendo o cálculo da polaridade final dado através de uma média ponderada. Desta forma, o presente módulo pretende constituir uma base que possa depois ser melhorada em função do contexto aplicado, para obtenção de resultados mais satisfatórios.

5.4 Módulo Extração de Triplos

Esta secção apresenta o teste realizado ao extrator de triplos, pertencente ao módulo de Extração de Triplos, descrito na secção 4.3.8.

O objetivo passa por avaliar os resultados obtidos por este extrator ao definir, para cada frase proveniente de uma publicação, um triplo no formato de sujeito, predicado e objeto. Este triplo será então comparado com outro anotado

manualmente, de forma a conseguir validar o desempenho deste módulo na identificação de cada um dos três componentes da frase (sujeito, predicado e objeto).

5.4.1 Descrição do Teste Realizado

Para realização do teste foi necessário anotar manualmente um conjunto de triplos, visto não existir nenhuma coleção já disponibilizada para o efeito. Assim, foram recolhidos um conjunto de textos provenientes de várias fontes noticiosas online, bem como excertos de blogs, tendo sido extraídas frases para anotação manual do sujeito, predicado e objeto respetivo. Este módulo apenas faz sentido que seja aplicado a texto estruturado, detentor de uma sintaxe a partir do qual seja possível extrair os vários componentes do triplo, pelo que, comentários e publicações menos estruturadas, pertencentes sobretudo a redes sociais como o Twitter, não foram tidos em conta.

Desta forma, de uma população de 10 mil frases provenientes de publicações de redes sociais, notícias online e blogues (constituída por 5 mil frases provenientes de publicações do Facebook e 5 mil de notícias online e blogues) foram assim etiquetadas manualmente um conjunto de mil frases. O método utilizado para calcular a amostra baseou-se na tabela para estatísticas de (Arkin & Colton (1950)). De acordo com esta tabela[64], a seleção aleatória de uma amostra composta por mil indivíduos de um universo de 10 mil, permite alcançar uma margem de erro na casa dos 3%, com um coeficiente de confiança de 95.5%.

Tal como já foi explicado, devido à não existência de estrutura sintática na frase, nem sempre é possível realizar a extração de um triplo com sucesso. Assim, sempre que este cenário sucedia, era selecionada aleatoriamente outra frase da população inicial, para garantir que o extrator retorna resultados válidos e se consegue avaliar com sucesso o desempenho deste, face ao triplo anotado manualmente.

A etiquetagem foi realizada por dois anotadores, tendo sido previamente definido um conjunto de diretivas entre ambos os elementos com o objetivo de uniformizar todo o processo de anotação realizado, através, por exemplo, da definição de quais os predicados compostos considerados válidos (que contêm verbo principal e auxiliar ou que contêm advérbio de negação). As principais discordâncias existentes entre ambas as anotações têm que ver essencialmente com a especificidade atribuída a cada um dos elementos. A título de exemplo, na frase "*Papa pretende discutir reforma na Igreja*", é discutível a inclusão da expressão "*reforma na igreja*" como sendo o objeto ou considerar somente

[64] É possível visualizar a tabela em *http://www.mbi.com.br/MBI/biblioteca/tutoriais/amostragem/.*

"*reforma*". O predicado também causou algumas discrepâncias em frases como "... *escreve Manuel Carvalho sobre o despedimento*" em que existiam dúvidas se o predicado considerado devesse ser só "*escreve*" ou "*escreve sobre*".

O cálculo da concordância, isto é, do número de classificações iguais dadas por ambos os anotadores no universo total de elementos pertencentes a cada componente do triplo, é apresentado na Tabela 5.12.

Componente do triplo	Concordância
Sujeito	0.980
Predicado	0.969
Objeto	0.945

Tabela 5.12: Cálculo da concordância existente entre os dois anotadores.

5.4.2 Resultados

Na Tabela 5.13 é possível visualizar as várias métricas recolhidas para o teste realizado, sendo que cada valor é dado pela média do resultado obtido usando cada uma das anotações.

Componente do Triplo	Precisão	Abrangência	Medida-F
Sujeito	0.8565	0.9118	0.8832
Predicado	0.8866	0.7719	0.8252
Objeto	0.6567	0.6341	0.6451
Média	**0.7999**	**0.7726**	**0.7845**
Desvio Padrão	**0.1020**	**0.1134**	**0.1014**

Tabela 5.13: Medidas de avaliação do extrator de triplos.

5.4.3 Análise de Resultados

O processo de anotação manual de triplos para realizar o teste não foi simples, isto porque muitas vezes foi difícil conseguir definir o grau de especificidade desejado para cada componente do triplo. Por exemplo, na seguinte publicação:

Comunidade ucraniana em Lisboa acusa russos de perseguições na Crimeia.

Em que o sujeito definido foi "Comunidade ucraniana em Lisboa", a adição de "em Lisboa" acaba por ser discutível, estando somente ao critério da pessoa que faz a anotação decidir se acha que é relevante ou não a sua adição.

De uma forma geral, as métricas recolhidas para a avaliação do sujeito e do predicado revelaram-se bastante satisfatórias. No caso específico do sujeito, os

erros ocorridos no teste têm muito que ver com a questão da especificidade referida anteriormente, uma vez que o extrator, regra geral, considera como sujeito expressões mais longas do que as que foram consideradas nos triplos anotados manualmente para teste. Já no que diz respeito ao predicado, a maioria das discrepâncias ocorridas devem-se ao facto do analisador sintático de dependências considerar como raiz da árvore um único verbo. Isto faz com que predicados compostos por mais do que uma palavra não sejam anotados na sua totalidade. Nas frases abaixo é possível visualizar alguns exemplos de predicados compostos:

*Passageiros das ligações com Bissau **vão ser** reembolsados.*

*Governo **não autoriza** manifestação na ponte 25 de Abril.*

Na primeira frase o predicado definido é "vão ser" e no segunda "não autoriza", sendo que, o extrator de triplos considera apenas "vão" e "autoriza" como sendo os predicados respetivos. Este seria um dos aspetos passíveis de melhoramento, através da definição de um conjunto de regras para obtenção destes casos particulares, como, por exemplo, do predicado composto "vão ser", sendo necessário neste caso extrair o verbo principal e o auxiliar no infinitivo.

O componente que apresentou piores resultados no teste foi o objeto, devido também à questão da especificidade desejada (falada para o sujeito) pois frases como:

Petição contra venda de 85 obras do artista Miró é discutida no parlamento.

Onde o predicado anotado para teste foi "discutida no parlamento", o extrator considerou unicamente "discutida", devido ao facto de apenas esta palavra representar um termo (obtido através do módulo Extrator de Termos e Expressões Multipalavra).

Um outro ponto de falha frequente são as frases que contêm mais que um triplo associado, como o seguinte exemplo:

Benfica vence Marítimo e lidera a liga.

A partir do qual é possível identificar os dois triplos seguintes:

Benfica vence Marítimo.

Benfica lidera a liga.

Embora, na prática, o extrator apenas esteja a reconhecer um deles, não realizando nenhuma separação pela conjunção coordenativa "e". Esta situação faz com que, em frases significativamente mais complexas, visto não ser realizado nenhum tipo de separação pelas conjunções existentes, leva muitas vezes à incorreta classificação do objeto na frase. Um dos possíveis melhoramentos a realizar no extrator de triplos seria precisamente conseguir "delimitar" a frase em

várias subsecções que permitissem a definição de vários triplos a partir de uma única frase.

5.5 Módulo Extração de Tópicos

Esta secção apresenta os testes realizados ao extrator de tópicos, pertencente ao módulo de Extração de Tópicos, descrito na secção 4.3.5.

Com estes testes pretende-se avaliar os resultados obtidos por este extrator ao obter tópicos para um conjunto de notícias que detêm já uma categoria previamente definida. A ideia passa então por avaliar se o tópico atribuído está de acordo com o respetivo contexto, tendo em conta a categoria inicialmente atribuída à notícia em questão, permitindo assim avaliar a capacidade de classificação de documentos.

5.5.1 Descrição do Teste Realizado

De forma a avaliar o extrator de tópicos, foram recolhidas 2,640 notícias em português, provenientes de sete categorias distintas da WikiNews[65] - "Ciência e tecnologia", "Cultura e entretenimento", "Desastres e acidentes", "Economia e negócios", "Política e conflitos", "Religião" e "Justiça". O número de notícias provenientes de cada categoria é apresentado na Tabela 5.14, tendo sido para cada uma recolhidas 400 notícias, à exceção da categoria "Religião" que continha somente 240. Na WikiNews, regra geral, o número de notícias pertencentes a cada categoria não excede o valor 480, daí que, dessa população inicial, se tenha selecionado aleatoriamente 400, uma vez que se trata de um número que permite obter uma representação significativa de cada categoria.

Para realização dos testes foram gerados tópicos a partir das 2,640 notícias obtidas, sendo no primeiro teste extraídos 7 tópicos e no segundo 20 tópicos. Tal como referido na secção referente ao módulo Extrator de Tópicos (secção 4.3.5), cada tópico é definido através de um conjunto de termos e, de forma a possibilitar a classificação das notícias, teve que ser previamente gerado um modelo para cada uma das experimentações, de forma a ser possível inferir o tópico associado a cada notícia. Após este passo, com base nos termos provenientes de cada um dos tópicos obtidos, foi manualmente atribuída uma "etiqueta" que permite representar o tópico em questão numa só expressão. A título de exemplo, o tópico constituído por palavras como *católica*, *vaticano*, *igreja*, *papa*, *bispo* foi definido

[65] Disponível em *https://pt.wikinews.org/wiki/Página_principal*.

como pertencente a "Religião", enquanto *internet*, *Google*, *tecnologia*, *universidade*, *cientistas* foi definido como pertencente a "Ciência e tecnologias".

A seleção de 7 tópicos para o primeiro teste teve que ver com o facto de terem sido extraídas 7 categorias da WikiNews, de forma a realizar-se uma comparação direta entre ambas. Já a escolha de 20 tópicos para o segundo teste, deveu-se, por um lado, à necessidade de aumentar o número relativamente ao primeiro teste (de forma a verificar se os resultados continuavam a ser consistentes), mas, por outro lado, também foi necessário criar um compromisso, para que o número não fosse excessivo e dificultasse demasiado a tarefa de análise e atribuição manual de uma etiqueta a cada tópico.

O objetivo desta experimentação é que todos os textos sejam classificados pelos modelos criados anteriormente (de 7 e 20 tópicos respetivamente), de forma a validar se essa classificação faz sentido, tendo em conta a categoria atribuída originalmente a cada texto pela WikiNews.

Categoria	**Número de textos**
Ciência e tecnologia	400
Cultura e entretenimento	400
Desastres e acidentes	400
Economia e negócios	400
Política e conflitos	400
Religião	240
Justiça	400
Total	**2,640**

Tabela 5.14: Número total de textos considerados para teste do extrator de tópicos.

5.5.2 Resultados

Os resultados são apresentados em duas tabelas, sendo visível na Tabela 5.15 os resultados da avaliação realizada para as 2,640 notícias classificadas em 7 tópicos e na Tabela 5.16 para a classificação com 20 tópicos.

Tópico/Categoria	Ciência e tecnologia	Cultura e entretenimento	Desastres e acidentes	Economia e negócios	Política e conflitos	Religião	Justiça
Religião	10	188	18	3	17	124	34
Política	45	44	12	113	145	16	20
Justiça	10	22	9	28	65	8	171
Economia	87	42	9	202	39	8	17
Conflitos	13	22	38	14	102	73	83
Ciência e tecnologia	206	66	8	22	9	1	19
Acidentes	29	16	306	18	23	10	56

Tabela 5.15: Resultados da classificação do extrator de tópicos para 7 tópicos gerados.

Tópico/Categoria	Ciência e tecnologia	Cultura e entretenimento	Desastres e acidentes	Economia e negócios	Política e conflitos	Religião	Justiça
Conflitos e Justiça	1	10	2	5	23	12	79
Religião	1	6	2	0	9	93	8
Ciência e Saúde	25	15	24	9	8	3	10
Economia	28	12	4	149	11	2	4
Tecnologia e Meio Ambiente	67	8	7	1	4	4	2
Entretenimento	10	39	8	4	2	4	8
Acidentes	10	0	119	14	5	0	7
Cultura e Ensino	24	34	2	17	16	21	26
Política e Conflitos	12	10	5	13	43	11	24
Ciências e Educação	68	5	1	5	2	4	8
Media Social	6	27	5	16	8	2	5
Lei e Política	6	9	3	20	34	3	91
Desastres Naturais	19	5	85	13	14	1	2
Desastres e Crimes	0	8	85	2	11	10	69
Tecnologia	73	20	1	15	2	0	21
Campanhas Políticas	28	10	3	65	51	1	4
Eleições Políticas	5	13	0	12	48	6	10
Arte e Cultura	7	153	6	1	2	9	1
Relações Políticas	10	15	6	36	62	5	14
Atentados	0	1	32	3	45	49	7

Tabela 5.16: Resultados da classificação do extrator de tópicos para 20 tópicos gerados.

5.5.3 Análise de Resultados

Na Tabela 5.15 é possível visualizar os resultados da classificação realizada para as 2,640 notícias com geração de 7 tópicos. O resultado da análise para cada uma das categorias atribuídas pela WikiNews é apresentado de seguida:

- **Ciência e tecnologia:** a maioria das notícias foi classificada corretamente como pertencendo ao tópico "Ciência e tecnologia" (cerca de 206 notícias em 400). Apesar do segundo tópico a reunir mais classificações ser "Economia" (87 notícias) e não estar muito relacionado com a categoria considerada, a diferença de valor entre ambos ainda é considerável.

- **Cultura e entretenimento:** grande parte dos documentos foi classificado como "Religião" (188 notícias em 400) o que, apesar de representar conceitos distintos, é aceitável visto muitas notícias existentes na categoria "Cultura e entretenimento" mencionarem festas e acontecimentos religiosos e isso acaba por refletir-se nos resultados. O mesmo acontece para o tópico "Ciência e tecnologia" (classificado com 66 notícias), uma vez que muitas das notícias acerca de descobertas e eventos referentes a este tema encontram-se igualmente presentes na categoria "Cultura e entretenimento".

- **Desastres e acidentes:** a grande maioria das notícias (306 em 400) foi classificada como pertencente ao tópico "Acidentes" o que comprova que foram corretamente classificadas.

- **Economia e negócios:** a generalidade dos documentos foi classificada como pertencendo a "Economia" e a "Política" (202 e 113 notícias, respetivamente). O tópico "Política" pode ser explicado pela forte influência que o sector político detém sobre a economia e sobre os negócios das empresas públicas, existindo assim várias notícias que evidenciam a relação entre estes dois temas e assim comprovando a correta classificação.

- **Política e conflitos:** a maioria das notícias foi classificada como pertencendo a "Política" e "Conflitos" (145 e 102, respetivamente).

- **Religião:** grande parte das notícias foi considerada como pertencente aos tópicos "Religião" e "Conflitos" (124 e 73 notícias, respetivamente). A inclusão de "Conflitos" também é aceitável pelo facto de serem incluídos inúmeros textos relacionados com conflitos provocados por questões religiosas, como os atentados suicidas reivindicados por grupos islâmicos.

- **Justiça:** a maior parte das notícias foi classificada como pertencendo aos tópicos "Justiça" e "Conflitos" (171 e 83 notícias, respetivamente). Uma vez que esta categoria contém um elevado número de notícias acerca de terrorismo e guerra, faz sentido que sejam também classificadas como pertencentes à categoria "Conflitos".

Na Tabela 5.16 é possível visualizar os resultados da classificação realizada para as 2,640 notícias ao serem gerados 20 tópicos. O resultado da análise para cada uma das categorias atribuídas pela WikiNews é o seguinte:

- **Ciência e tecnologia:** a grande maioria dos documentos foi classificada como pertencente aos tópicos "Ciências e Educação", "Tecnologia" e "Tecnologia e Meio Ambiente" (68, 67 e 73 notícias, respetivamente) pelo que a classificação realizada foi correta.

- **Cultura e entretenimento:** grande parte das notícias foi definida como pertencente aos tópicos "Arte e Cultura", "Cultura e Ensino" e "Entretenimento" (153, 34 e 39 notícias, respetivamente) sendo que também neste caso a classificação foi realizada acertadamente.

- **Desastres e acidentes:** a generalidade dos documentos foi definida como fazendo parte de "Acidentes", "Desastres Naturais" e "Desastres e Crimes" (119, 85 e 85 notícias, respetivamente) o que também faz sentido.

- **Economia e negócios:** a larga maioria dos resultados diz respeito aos tópicos "Economia" e "Campanhas Políticas" (149 e 65 notícias respetivamente), relevando também resultados bastante satisfatórios nesta categoria.

- **Política e conflitos:** encontra-se associada sobretudo a notícias que foram classificadas como "Conflitos e Justiça", "Lei e política", "Política e Conflitos", "Campanhas Políticas", "Eleições Políticas", "Relações Políticas" e "Atentados", apresentando por isso resultados bastante satisfatórios, uma vez que todos os tópicos referenciados se relacionam com política ou com conflitos.

- **Religião:** esta categoria está relacionada com os tópicos referentes a "Religião" e "Atentados" (93 e 49 notícias, respetivamente), visto muitos dos atentados relatados nas notícias deterem motivações religiosas.

- **Justiça:** a grande maioria dos documentos está associada aos tópicos "Conflitos e Justiça", "Lei e Política" e "Desastres e Crimes" (79, 91 e 69 notícias, respetivamente). De facto, muitas das notícias que detêm esta categoria dizem respeito a conflitos, guerra e terrorismo, bem como a disputas entre empresas por patentes e leis, daí que englobe também outros tópicos. A classificação foi também considerada correta neste caso.

De uma forma geral, os resultados da classificação realizada foram bastante satisfatórios, tanto no modelo gerado com 7 como 20 tópicos. O exemplo mais díspar encontrado ocorreu no teste realizado ao modelo de 7 tópicos, onde 87 das

400 notícias pertencentes à categoria da WikiNews "Ciências e Tecnologias", foram associadas ao tópico relacionado com "Economia", apesar de não ter sido um número muito significativo.

Em muitos casos existiu também associação dos tópicos a categorias aparentemente distintas, como o caso do tópico "Conflitos", que foi muitas vezes associado à categoria "Religião". No entanto, depois de uma análise ao conteúdo presente nas notícias, verificou-se que, apesar de representarem categorias distintas, existia interseção a nível de conteúdo. De facto, muitas notícias pertencentes à categoria "Religião" referiam conflitos religiosos e isso refletiu-se na atribuição dos tópicos, acontecendo o mesmo para outras categorias de notícias identificadas.

5.6 Módulo Pesquisa e Recomendação Semântica

Pelo facto do foco da tese ser o desenvolvimento da biblioteca, não foram realizados testes no sistema com o intuito de averiguar a qualidade dos módulos desenvolvidos. No entanto, ao longo da implementação dos módulos de Pesquisa e Recomendação Semântica, foram especificados um conjunto de testes unitários, isto é, unidades de código que executam uma funcionalidade específica a ser testada no código desenvolvido. Assim, pretende-se essencialmente testar um método ou sequência de métodos de uma classe de forma isolada, com o intuito de validar as várias unidades básicas constituintes de um módulo. O principal objetivo destes testes é assegurar que o código executa tal como pretendido e são também bastante úteis para garantir que este continua funcional após eventuais modificações realizadas, possibilitando o desenvolvimento de novos recursos sem o esforço acrescido de realização manual de testes.

Os testes unitários foram desenvolvidos recorrendo à *framework* Java JUnit[66], que possibilita a criação de um modelo padrão de testes e consequente automatização de todo o processo para realização de testes unitários, com uma apresentação de resultados bastante simples e intuitiva.

De seguida é apresentada a especificação dos vários testes unitários desenvolvidos para o módulo de Pesquisa e de Recomendação Semântica.

5.6.1 Testes Unitários - Recomendação Semântica

No módulo de Recomendação Semântica foram realizados testes unitários com o intuito de verificar se os dois pressupostos base estavam a ser cumpridos:

[66] Disponível em *http://junit.org/*.

- Obtenção com sucesso das publicações (e da respetiva informação extraída com o auxílio da biblioteca) a partir da base de dados de triplos. Este passo é necessário para que seja possível realizar a recomendação de outras publicações, uma vez que se trata de recomendação baseada em conteúdo. Assim sendo, foram definidos métodos cujo objetivo é verificar se, para um conjunto de publicações existentes na BD, é obtido com sucesso todo o seu conteúdo. Foi então realizada uma divisão entre os vários tipos de fontes consideradas neste trabalho, ou seja, publicações pertencentes à rede social Twitter, Facebook, blogues e notícias online:

  - *testGetTweetPostById*
  - *testGetFacebookPostById*
  - *testGetBlogPostById*
  - *testGetOnlineNewsById*

- Obtenção com sucesso de um conjunto de recomendações associadas a cada uma das publicações extraídas anteriormente.

  - *testGetRecommendationForTweets*
  - *testGetRecommendationForFacebook*
  - *testGetRecommendationForBlog*
  - *testGetRecommendationForOnlineNews*

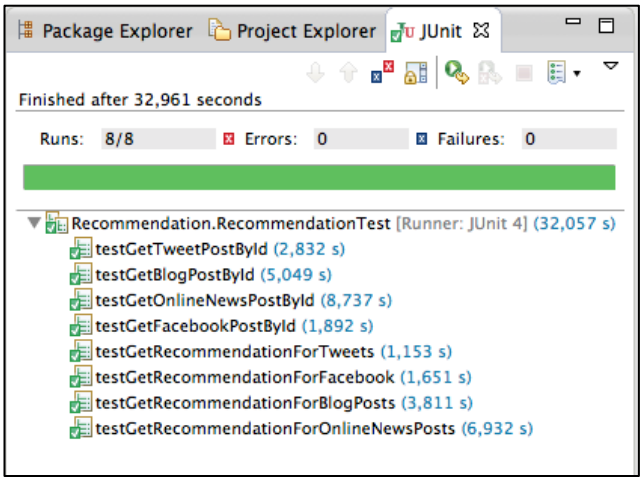

Figura 5.1: Resultado dos testes unitários realizados ao módulo da Recomendação Semântica usando o Junit.

5.6.2 Testes Unitários - Pesquisa Semântica

No módulo de Pesquisa Semântica foram realizados testes unitários com o objetivo de assegurar que todo o processo para obtenção das publicações associadas a uma determinada *query* foi realizado com sucesso. Desta forma, foi assim possível ir continuamente testando os vários tipos de *queries* suportadas pelo sistema, ao

mesmo tempo que os desenvolvimentos estavam a ser realizados, garantindo que o sistema se encontrava funcional. Uma vez que foi também desenvolvida a pesquisa por palavra-chave (em inglês conhecida por *keyword search*), foi adicionado um método para garantir o correto funcionamento da mesma.

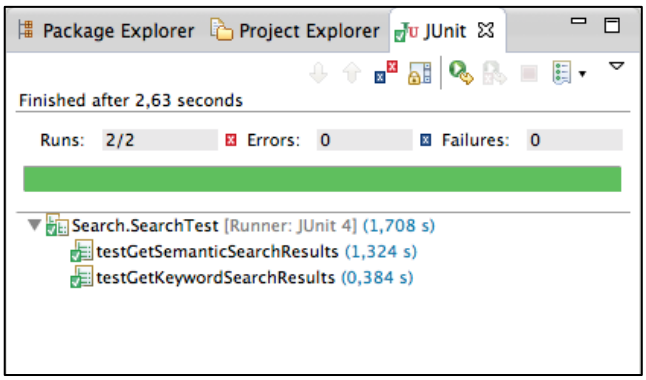

Figura 5.2: Resultado dos testes unitários realizados ao módulo da Pesquisa Semântica usando o Junit.

5.7 Testes de Validação de Requisitos Alto-Nível

A modelação de requisitos alto-nível foi conseguida através da definição dos vários casos de uso, sendo que cada caso de uso representa uma funcionalidade disponibilizada pelo cliente Web aos utilizadores. A validação dos requisitos alto-nível é realizada através de um processo de 2 fases: primeiro verifica-se que a funcionalidade se encontra acessível para o utilizador (único ator no sistema desenvolvido) e em segundo confirma-se que o comportamento desencadeado pela página Web é o esperado.

Na Tabela 5.17 são analisados os diferentes casos de uso, tendo em atenção os dois aspetos mencionados anteriormente: existência da funcionalidade e se o comportamento é, de facto, o esperado. Não se entra em detalhe relativamente à execução destes testes, uma vez que estes podem ser encontrados no anexo B, onde é efetuada a especificação textual de cada caso de uso.

Requisito Alto-Nível	**Funcionalidade Disponível**	**Comportamento Esperado**
Efetuar pesquisa	✓	✓
Visualizar informação extraída	✓	✓
Recomendação de conteúdo relacionado	✓	✓
Permitir navegação no conteúdo	✓	✓

Tabela 5.17: Validação dos requisitos alto-nível apresentados.

A partir da tabela pode assim concluir-se que todos os requisitos alto-nível foram implementados e se comportam de acordo com o esperado. Este processo permite assim avaliar de forma objetiva o cumprimento de um conjunto de requisitos fundamentais para o correto funcionamento do cliente Web.

5.8 Conclusões

O principal objetivo desta tese foi o desenvolvimento de uma biblioteca capaz de extrair vários tipos de informação semântica a partir de um conjunto de textos em linguagem natural.

A elaboração deste tipo de ferramenta é por si só bastante complexa, devido não só a questões relacionadas com a complexidade da própria linguagem (ter que lidar com a ironia ou a ambiguidade existente), mas também porque muitas vezes o conteúdo textual não detém qualquer tipo de regras ao nível da sintaxe (sobretudo em textos provenientes de redes sociais), dificultando a tarefa de análise e extração de informação. O número de recursos e ferramentas disponíveis para a língua portuguesa é também bastante escasso e limitado, tornando ainda mais desafiante todo o processo.

Uma vez que alguns módulos apresentam um nível de complexidade consideravelmente superior a outros, optou-se por não realizar testes em todos eles. Esta decisão deveu-se também ao tempo limitado para realização da fase de experimentação e ao facto de, para alguns módulos, ter sido necessário definir manualmente os testes através da anotação de publicações ou da geração de *datasets* (visto para a língua portuguesa existirem poucos recursos disponibilizados para este efeito), tornando-se um processo bastante moroso.

Assim, não foi efetuada experimentação no módulo de Pré-processamento uma vez que foi desenvolvido maioritariamente recorrendo a ferramentas externas (como o caso do *tokenizador*, analisador sintático de dependências, etiquetador gramatical, etc.) e em que a qualidade depende diretamente dos recursos utilizados e dos vários dicionários de *emoticons*, pitês e *stopwords*. O módulo de Extração de Termos e Expressões Multipalavra também não foi sujeito a testes, uma vez que o seu objetivo prende-se essencialmente com a extração de padrões obtidos a partir de um conjunto de expressões regulares, pelo que os resultados obtidos dependem também diretamente do conjunto de expressões pré-definidas. Visto o módulo de Extração de Metadados deter maioritariamente funções de suporte, ao funcionar como *parser* que permite filtrar a informação mais relevante (como o autor, texto da publicação, comentários, etc.) extraída a partir das publicações provenientes de redes sociais, notícias online e blogues, também não foi alvo de experimentação.

De uma forma geral, todos os testes executados revelaram resultados bastante satisfatórios tendo em conta o tempo limitado para desenvolvimento de cada um dos módulos e com base nos resultados da experimentação foi ainda possível definir possíveis melhoramentos a realizar.

A conclusão mais importante a retirar é que a biblioteca possui agora um importante conjunto de módulos detentores de uma qualidade razoável e para o qual foram identificados pontos fortes e fracos, que permitiram desta forma definir um conjunto de diretrizes para realização de possíveis melhoramentos futuros.

Capítulo 6

Conclusões e Trabalho Futuro

O principal objetivo desta tese é a análise e extração de diversos tipos de informação a partir de fontes da Web 2.0. Com esse intuito, foi desenvolvida uma biblioteca que disponibiliza um conjunto de módulos para extração de conhecimento, sendo a mesma integrada num sistema que permite usar as suas funcionalidades para realizar pesquisa, recomendação e navegação semântica sobre o conteúdo extraído. O foco do projeto é unicamente a língua portuguesa, o que por si só também o torna mais desafiante, devido ao número reduzido de recursos disponibilizados nesta língua.

De uma forma geral, os resultados experimentais alcançados no desenvolvimento dos vários módulos da biblioteca revelaram-se bastante razoáveis, tendo em conta as limitações existentes a nível temporal para implementação e teste dos mesmos, bem como o facto do número de recursos disponibilizados ser bastante escasso. O desempenho dos vários módulos foi avaliado através do cálculo da medida-F, sendo os resultados médios alcançados os seguintes: 74.81% e 71.45% para o Extrator de Entidades vocacionado para texto estruturado e não estruturado, respetivamente; 55.20% para o Analisador de Sentimentos e 78.45% para o Extrator de Triplos.

A biblioteca contém agora um conjunto base de módulos que possuem uma qualidade bastante aceitável e permitem, não só a realização de melhoramentos futuros (referidos nas várias experimentações realizadas), como possibilitam, com relativa facilidade, a extensão dos mesmos para aplicação a domínios mais específicos (por exemplo, análise de sentimentos no contexto político, tecnológico, etc.).

De facto, a participação neste projeto foi extremamente gratificante e recompensadora por englobar diversos domínios da área de Processamento de Linguagem Natural o que, por um lado, possibilita a aplicação da biblioteca a projetos de domínios totalmente distintos e, por outro, permitiu a aquisição de conhecimentos bastante relevantes acerca de um vasto conjunto de áreas de PLN. Devido à escassez de recursos para o português, mesmo ainda durante a fase de desenvolvimentos, foi disponibilizada a biblioteca aos alunos João Marques e António Marques. Ambos usaram o módulo de Pré-processamento e de Extração de Entidades, no âmbito da sua tese de mestrado, para desenvolvimento de um sistema de deteção de eventos por análise semântica de conteúdo da Web 2.0 e utilização do contexto do utilizador (tópicos de interesse) para melhorar a performance da pesquisa em ambientes móveis, respetivamente. O aluno Ricardo

Rodrigues fez igualmente uso do módulo de Extração de Entidades no contexto da sua tese de doutoramento de resposta automática a perguntas, baseada em tecnologias da WS e PLN para o português.

Por outro lado, o contacto com algumas das tecnologias da Web Semântica permitiu aferir a adaptabilidade que estas detêm, quer pela disponibilização de um esquema conceptual bastante flexível (pela facilidade na adição e remoção de propriedades e classes na ontologia) como pelo poder de expressividade do SPARQL, possibilitando a realização de *queries* complexas, que envolvem o cruzamento de diversos tipos de informação, de uma forma simples e intuitiva.

Concluindo, os principais contributos deste trabalho são os seguintes:

- Elaboração do estado da arte referente ao Processamento de Linguagem Natural e Web Semântica, bem como análise de diversas ferramentas e recursos disponíveis para a língua portuguesa;

- *Crawlers* que possibilitam a obtenção de publicações e comentários provenientes das redes sociais Twitter e Facebook, bem como de diversas fontes de notícias online (Jornal Público, Sapo Desporto, TEK e blogs do jornal Expresso);

- Desenvolvimento de uma biblioteca modular para extração de vários tipos de conhecimento semântico a partir de recursos textuais em português, com o intuito de ser integrável noutros projetos e que oferece suporte a vários sistemas operativos;

- Desenvolvimento de um sistema que, faz uso da informação extraída a partir da biblioteca para, a partir de um conjunto de dados de teste, possibilitar a realização de pesquisa semântica e navegação sobre a informação extraída, bem como obtenção de recomendações de conteúdo;

- Cliente Web que demonstra as funcionalidades do sistema e consequentemente da biblioteca, a partir de um *dataset* de teste;

- Anotação manual de um conjunto de publicações de forma a possibilitar a realização de testes nos módulos de Extração de Entidades e Extração de Triplos. Estes recursos podem ser utilizados para treino e teste noutros projetos, fornecendo assim um conjunto de *datasets* essenciais para estas operações e que não se encontram disponíveis para a língua portuguesa;

- Criação de um conjunto de recursos léxicos para o português, reutilizáveis noutros projetos (dicionários de pitês e *emoticons*, bem como as listas de entidades ou os inversores de polaridade);

- Alguns módulos da biblioteca foram usados para participação na avaliação SemEval-2014 (mais concretamente na tarefa de análise de sentimentos de publicações do Twitter), do qual resultou a aceitação de um artigo científico (Leal et al. (in press)).

Apesar da metodologia ágil adotada nos desenvolvimentos realizados não requerer explicitamente a criação de um diagrama de Gantt, a sua elaboração ajudou a identificar todo o trabalho realizado ao longo do ano letivo, pelo que também foi incluído. Nos diagramas de Gantt presentes na Figura 6.1 e Figura 6.2 é possível visualizar as várias tarefas planeadas e desenvolvidas ao longo do primeiro e segundo semestre, respetivamente.

6.1 Publicação SemEval-2014

Embora a biblioteca tenha sido desenvolvida para dar suporte à língua portuguesa (mais concretamente português de Portugal), grande parte da metodologia desenvolvida é facilmente aplicável a outras línguas, desde que os recursos utilizados (dicionários, listas, etc.) sejam também criados para a língua respetiva. Desta forma, foi assim possível usar algumas das funcionalidades providenciadas pela biblioteca para participação na avaliação SemEval-2014[67] com a equipa CISUC_KIS, na tarefa de análise de sentimentos em publicações do Twitter.

Assim foram usados alguns dos módulos da biblioteca (nomeadamente o módulo Extração de Tópicos, módulo de Pré-processamento e Extrator de Termos e Expressões Multipalavra) para geração de um conjunto de *features* usadas depois para treinar um classificador SVM (Joachims (1998)) que permite atribuir a um determinado *tweet* uma polaridade positiva, negativa ou neutra. O desempenho do modelo foi avaliado através do cálculo da medida-F referente às classes positiva e negativa, usando a validação cruzada 10-fold. Na Tabela 6.1 é possível visualizar os resultados da avaliação oficial[68], tendo a equipa CISUC_KIS obtido valores bastante satisfatórios, uma vez que a média dos resultados obtidos a partir dos vários *datasets*, permitiu alcançar o 2º lugar. Desta participação resultou ainda a aceitação de um artigo científico, onde é explicada a abordagem seguida, bem como todo o conjunto de *features* consideradas no treino do classificador (Leal et al. (in press)).

[67] Referência em *http://alt.qcri.org/semeval2014/*.

[68] Os resultados encontram-se disponíveis em *https://docs.google.com/spreadsheets/d/1CmDicfEIxRgyoAix9BsVcC3qoRFEq0XDTSnBLVdavu8/edit#gid=2040081749*.

Sistema	Live Journal 2014	SMS2013	Twitter2013	Twitter2014	Twitter 2014 Sarcasm
NRC-Canada	74.84	70.28	70.75	69.85	58.16
SAIL	69.34	56.98	66.80	67.77	57.26
TeamX	69.44	57.36	72.12	70.96	56.50
AUEB	70.75	64.32	63.92	66.38	56.16
CISUC_KIS	**74.46**	**65.90**	**67.56**	**67.95**	**55.49**
senti.ue	71.39	59.34	67.34	63.81	55.31
UKPDIPF	71.92	60.56	60.65	63.77	54.59
GPLSI	57.32	46.63	57.49	56.06	53.90
UMCC_DLSI_Graph	47.81	36.66	43.24	45.49	53.15
TUGAS	69.79	62.77	65.64	69.00	52.87
BUAP	53.94	44.27	56.85	55.76	51.52
ECNUd	69.44	59.75	62.31	63.17	51.43
Synalp-Empathic	71.75	62.54	63.65	67.43	51.06
SWISS-CHOCOLATE	73.25	66.43	64.81	67.54	49.46
SAP-RI-B	57.86	49.00	50.18	55.47	48.64
SU-FMI	68.24	61.67	60.96	63.62	48.34
AMI_ERIC	65.32	60.29	70.09	66.55	48.19
Lt_3	68.56	64.78	65.56	65.47	47.76
RTRGO	72.20	67.51	69.10	69.95	47.09
coooolll	72.90	67.68	70.40	70.14	46.66
Rapanakis	59.71	54.02	58.52	63.01	44.69
KUNLPLab	63.77	55.89	58.12	61.72	44.60
USP_Biocom	67.80	53.57	58.05	59.21	43.56
SentiKLUE	73.99	67.40	69.06	67.02	43.36
UMCC_DLSI_Sem	53.12	50.01	51.96	55.40	42.76
DejaVu	64.69	55.57	57.43	57.02	42.46
LyS	69.79	60.45	66.92	64.92	42.40
NILC_USP	69.02	61.35	65.39	63.94	42.06
Indian_InstTech_Patna	60.39	51.96	52.58	57.25	41.33
Citius-B	62.40	57.69	62.53	61.92	41.00
CMUQ-Hybrid	65.14	61.75	63.22	62.71	40.95
CMU-Qatar	65.63	62.95	65.11	65.53	40.52
Alberta	52.38	49.05	53.85	52.06	40.40
columbia_nlp	68.79	59.84	64.60	65.42	40.02
University-of-Warwick	39.60	29.50	39.17	45.56	39.77
UPV-ELiRF	64.11	55.36	63.97	59.33	37.46
IITPatna	54.68	40.56	50.32	48.22	36.73
lsis_lif	61.09	38.56	46.38	52.02	34.64
IBM_EG	59.24	46.62	54.51	52.26	34.14
SU-sentilab	55.11	49.60	50.17	49.52	31.49
SINAI	58.33	57.34	50.59	49.50	31.15

DAEDALUS	40.83	40.86	36.57	33.03	28.96

Tabela 6.1: Resultados da avaliação oficial do SemEval-2014 para a tarefa de análise de sentimentos em publicações do Twitter.

6.2 Principais Desafios

Esta secção tem como objetivo explicitar quais os principais desafios encontrados durante o desenvolvimento do projeto e a forma como os mesmos foram ultrapassados.

Assim, as principais dificuldades com o qual se teve que lidar no decurso da realização da tese foram os seguintes:

Qualidade dos recursos utilizados para treino

A qualidade dos recursos utilizados (nomeadamente da corpora linguística) detém um impacto significativo no treino e consequentemente influencia também a qualidade do modelo gerado. Um exemplo claro desta situação diz respeito ao extrator de entidades, que faz uso das coleções do HAREM para geração do modelo, e que, devido à existência de diferenças significativas no processo de anotação entre ambas as coleções, influenciou a qualidade do modelo gerado (tal como foi possível visualizar na experimentação apresentada na secção 5.1). Por vezes a melhor solução passa por rever a anotação realizada ou então obter um novo conjunto de dados para treino, sendo que, no âmbito da tese, o mesmo não foi possível realizar devido às restrições temporais existentes. Assim, optou-se por gerar um modelo em separado para cada uma das coleções.

Número muito reduzido de ferramentas e bibliotecas disponíveis

Pelo facto do número de ferramentas e bibliotecas que fornecem suporte à língua portuguesa ser muito reduzido, não existiram escolhas alternativas, tendo sido usado o que se encontrava disponível de forma a conseguir cumprir os prazos e não prolongar em demasia o tempo de implementação. A título de exemplo, recursos como o analisador sintático de dependências ou o *tokenizador* vocacionado para redes sociais, não foram alvo de comparação com outras ferramentas, devido a inexistência de alternativas que fossem gratuitas ou que fossem abrangidas por uma licença de software permissiva. Neste caso, a possibilidade de realizar melhoramentos foi também colocada de parte devido ao tempo limitado para implementação, estando por isso a escolha bastante restringida.

Implementação demorada dos algoritmos

A escassez ou inexistência de recursos (referida no ponto anterior), levou a que por vezes a implementação acabasse por ser mais morosa e complexa do que inicialmente estava previsto. Um destes casos ocorreu na implementação do módulo de Extração de Entidades, onde foram usadas as coleções do HAREM para geração de um modelo que apenas faz sentido que seja aplicado a texto estruturado (uma vez que as coleções são constituídas sobretudo por textos com estrutura sintática). No entanto, um dos requisitos do projeto é que fosse também dado suporte a texto proveniente de redes sociais. Pelo facto de não existir nenhuma coleção já etiquetada para esse efeito e o texto dessas fontes ser, regra geral, pouco estruturado, teve necessariamente de ser seguida uma abordagem distinta e naturalmente consumido um tempo superior ao que estava inicialmente previsto para implementação deste módulo.

Realização demorada de testes

Uma vez que para a realização dos testes no módulo de Extração de Entidades e Extração de Triplos foi necessária a anotação manual dos dados (visto não existirem coleções já etiquetadas para o efeito) e esse processo ser bastante moroso, revelou-se assim necessário conseguir encontrar um meio-termo entre não despender demasiado tempo a etiquetar, mas ao mesmo tempo assegurar que a amostra detém um valor adequado que seja suficientemente representativo da população em questão.

Desta forma, com base nas tabelas estatísticas de (Arkin & Colton (1950)), foi possível comprovar que a seleção aleatória de uma amostra composta por mil indivíduos de um universo de 10 mil permite, por um lado, alcançar uma margem de erro bastante aceitável (na casa dos 3%) e por outro, um tamanho base da população representativo o suficiente para obter margens de erro na ordem dos 1%.

6.3 Trabalho Futuro

Nesta secção são identificadas algumas direções para trabalho passível de ser realizado futuramente. Algumas destas indicações têm como objetivo melhorar o trabalho executado, enquanto outras exploram novas ideias e conceitos que seriam igualmente interessantes de desenvolver.

Desambiguação do sentido das palavras

Depois de analisar os resultados da experimentação realizada no módulo de Extração de Entidades, concluiu-se que, um dos aspetos mais relevantes que melhoraria consideravelmente o desempenho do mesmo, seria a inclusão de um processo para desambiguação do sentido das palavras. Isto porque, muitas vezes, a mesma palavra adquire um significado diferente mediante o contexto em que se encontra inserida e isso é algo bastante relevante no extrator de entidades. Por exemplo, um caso relativamente frequente é o uso da palavra "Portugal", que tanto se encontra associada à categoria Local, como no contexto desportivo a Organização (à Seleção Portuguesa de Futebol). Neste caso seria interessante ter em conta o contexto onde se encontra inserida a palavra para atribuição da respetiva categoria.

Análise de sentimentos

No módulo de Análise de Sentimentos seria interessante seguir uma abordagem não tão focada na extração de uma polaridade final associada ao texto na sua totalidade, mas antes conseguir atribuir uma polaridade a determinados elementos da frase que expressem algum tipo de opinião. Isto poderia eventualmente ser implementado usando o analisador sintático de dependências, de forma a perceber quais são as outras palavras que, na frase, estão a ser afetadas por uma determinada *opinion word* (ou seja, uma palavra detentora de opinião) permitindo assim, por exemplo, na seguinte frase:

Ela é muito falsa mas ele tem bom coração.

reconhecer que a expressão "Ela é muito falsa" tem associada uma polaridade negativa e "ele tem bom coração" uma polaridade positiva. O uso do analisador sintático de dependências possibilitaria também melhorar alguns aspetos na atual abordagem implementada, como o reconhecimento de quais as *opinion words* afetadas por um determinado inversor de polaridade, sendo que, na atual implementação, cada inversor afeta todas as *opinion words* até ao final de uma frase ou até encontrar pontuação como uma vírgula ou ponto e vírgula.

Seria igualmente interessante explorar uma abordagem mais vocacionada para a extração de aspetos, dado que apresenta mais informação útil para os utilizadores (abordagem mencionada na secção A.1.6). A principal razão para não adotar este tipo de metodologia nos desenvolvimentos realizados teve que ver com a complexidade associada ao lidar com texto proveniente de diversos domínios, que dificulta bastante a tarefa de extração de aspetos. Quando aplicada a um domínio mais particular, como, por exemplo, comentários às

aplicações Android[69] na PlayStore[70], onde o uso de determinados termos como "usabilidade" e "gráficos" constituem aspetos bastante utilizados no que se refere à avaliação das aplicações, este tipo de abordagem faz bastante sentido, uma vez que permite ao utilizador distinguir claramente quais os pontos positivos e negativos de cada aplicação.

Distinção de texto estruturado e não estruturado

Um outro ponto que seria interessante explorar diz respeito ao desenvolvimento de um módulo que permitisse efetuar a distinção entre texto não estruturado (isto é, texto sem estrutura sintática, como o caso da maioria das publicações e comentários provenientes de redes sociais) de texto estruturado. Isto permitiria automatizar a escolha das ferramentas mediante a perceção do tipo de texto sem ter que obrigar explicitamente os utilizadores da biblioteca a definir esse comportamento.

Melhoramento dos recursos utilizados

Um ponto que traria melhorias consideráveis nos resultados alcançados pelos módulos da biblioteca seria a adição de mais conteúdo aos recursos criados – desde as várias listas de entidades consideradas pelos extratores de entidades, aos dicionários de pitês, *stopwords*, *emoticons*, etc. que contam ainda com um número relativamente reduzido de elementos. Uma possível solução a considerar seria o desenvolvimento de uma ferramenta que permitisse, por exemplo, recorrer a bases de conhecimento como a Wikipédia para rapidamente expandir o número de elementos das mais variadas categorias consideradas. Seria igualmente útil a revisão de alguns recursos, como o caso da uniformização das várias versões do HAREM, que possibilitaria a obtenção de resultados mais satisfatórios no módulo de Extração de Entidades.

Explorar o uso das tecnologias da Web Semântica

Uma vez que o foco desta tese incidiu essencialmente sobre o desenvolvimento de uma biblioteca de Processamento de Linguagem Natural e devido também às restrições temporais impostas para realização da tese, a Web Semântica não teve um impacto tão relevante e consequentemente esta área acabou por não ser muito explorada.

Assim, um dos possíveis melhoramentos consiste em tirar proveito das potencialidades oferecidas pelas várias tecnologias da Web Semântica de forma a aper-

[69] Referência em *http://www.android.com/*.
[70] Disponível em *https://play.google.com/store*.

feiçoar os módulos. Por exemplo, permitir a realização de cruzamento de informação com outras bases de conhecimento (como o caso da DBpedia), para que, se numa determinada publicação fosse identificada unicamente a entidade "iPhone", recorrendo à informação disponibilizada pela DBpedia, fosse possível "perceber" que iPhone se trata de um produto da organização Apple e consequentemente enriquecer os módulos com este tipo adicional de informação.

Enriquecimento do contexto das publicações

Apesar de ter sido realizado um esforço no sentido de usar conteúdo específico da *media social* para melhoramento dos resultados dos vários módulos (por exemplo, uso dos *emoticons* na análise de sentimentos ou identificação de entidades a partir das *hashtags*), seria relevante conseguir obter mais informação útil para contextualização das publicações, sobretudo as de *microblogging* (como o caso dos *tweets*) que contêm menos informação.

De facto, verificou-se que grande parte destas publicações contêm um *link* para uma página Web e por isso, seria importante explorar o conteúdo presente nessas páginas como forma de complementar o pouco texto disponível. A título de exemplo, permitiria melhorar o desempenho do módulo de Extração de Tópicos quando aplicado somente a textos muito curtos, pela adição do conteúdo total da notícia original ou então de parte dele (como do seu sumário) no processo de obtenção de tópicos.

Uso de abordagens alternativas

Seria interessante utilizar abordagens alternativas às implementadas de forma a poder realizar comparações e por conseguinte escolher qual seria a mais adequada ou a que traria melhorias em função dos resultados obtidos. Em grande parte dos módulos foi excluído à partida o uso de abordagens de aprendizagem supervisionada, não porque se achasse que não seriam igualmente interessantes, mas devido essencialmente a restrições temporais, uma vez que requerem consideravelmente mais tempo para anotação dos dados usados no treino e teste.

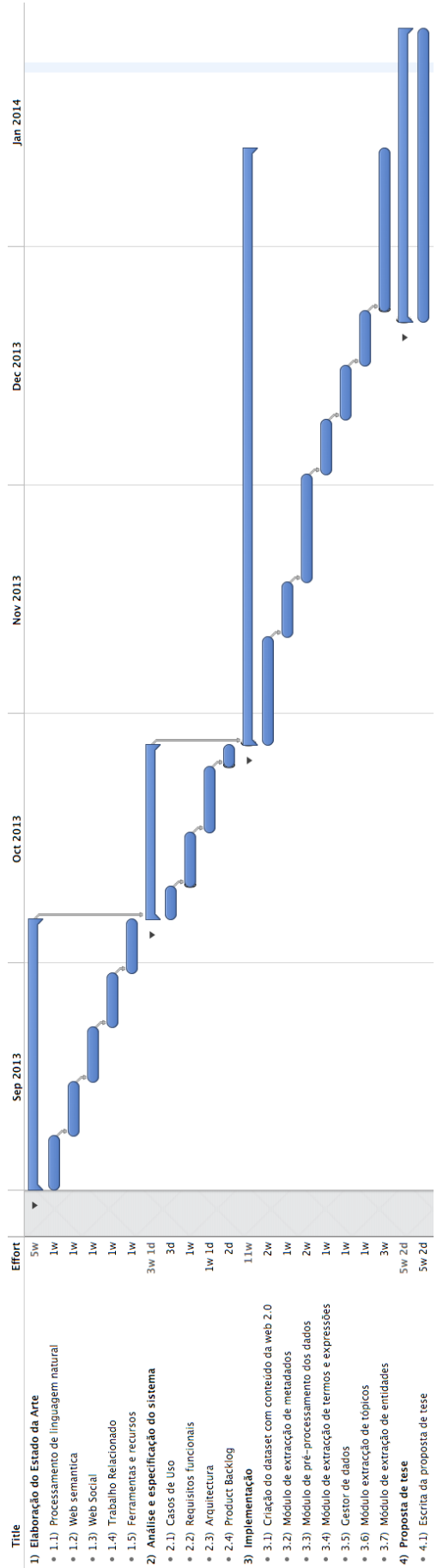

Figura 6.1: Diagrama Gantt com tarefas desenvolvidas no 1º Semestre.

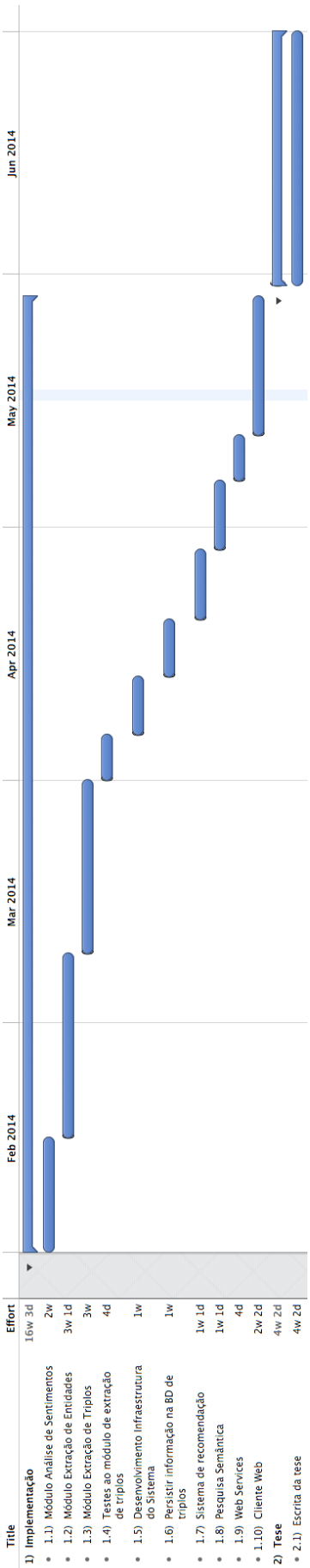

Figura 6.2: Diagrama Gantt com tarefas desenvolvidas no 2º Semestre.

Referências

Aggarwal, C. C., & Zhai, C. (2012). A survey of text clustering algorithms. In*Mining Text Data* (pp. 77-128). Springer US.

Agichtein, E., Gravano, L., Pavel, J., Sokolova, V., & Voskoboynik, A. (2001, May). Snowball: A prototype system for extracting relations from large text collections. In *ACM SIGMOD Record* (Vol. 30, No. 2, p. 612). ACM.

Allen, James F. "Time and time again: The many ways to represent time."*International Journal of Intelligent Systems* 6.4 (1991): 341-355.

Appelquist, D., Brickley, D., Carvahlo, M., Iannella, R., Passant, A., Perey, C., & Story, H. (2010). A standards-based, open and privacy-aware social web. *W3C Incubator Group Report*, *6*.

Arkin, H., & Colton, R. R. (1950). Tables For Statisticians.

Artz, D., & Gil, Y. (2007). A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, *5*(2), 58-71.

Berners-Lee, T., Fischetti, M., & Dertouzos, M. L. (1999). Weaving the Web. 1999. *Orion Business, New York*.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific american*, *284*(5), 28-37.

Berners-Lee, T. (2005). *Semantic Web Concepts.* Obtido em 7 de 9 de 2013, de http://www.w3.org/2005/Talks/0517-boit-tbl/

Bick, E. (2000). *The Parsing System" Palavras": Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework* (Vol. 202, pp. 589-637). Aarhus: Aarhus University Press.

Bikel, D. M., Schwartz, R., & Weischedel, R. M. (1999). An algorithm that learns what's in a name. *Machine learning*, *34*(1-3), 211-231.

Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked data-the story so far.*International Journal on Semantic Web and Information Systems (IJSWIS)*,*5*(3), 1-22.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research*, *3*, 993-1022.

Boley, H., Tabet, S., & Wagner, G. (2001). Design Rationale for RuleML: A Markup Language for Semantic Web Rules. In *SWWS* (Vol. 1, pp. 381-401).

Branco, A., & Silva, J. (2004). Evaluating Solutions for the Rapid Development of State-of-the-Art POS Taggers for Portuguese. In *LREC*.

Branco, A., Castro, S., Silva, J., & Costa, F. (2011). CINTIL DepBank Handbook: Design options for the representation of grammatical dependencies.

Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., & Yergeau, F. (1997). Extensible markup language (XML). *World Wide Web Journal*, *2*(4), 27-66.

Brickley, D., & Guha, R. V. (2000). Resource Description Framework (RDF) Schema Specification 1.0: W3C Candidate Recommendation 27 March 2000.

Brickley, D., Guha, R. V., & McBride, B. (2004). RDF vocabulary description language 1.0: RDF Schema. W3C Recommendation (2004).

Broekstra, J., & Kampman, A. (2004). Serql: An rdf query and transformation language. In *Submitted to the International Semantic Web Conference, ISWC* (Vol. 2004).

Cardoso, N. (2008). Rembrandt-reconhecimento de entidades mencionadas baseado em relaçoes e análise detalhada do texto. *Mota and Santos (Mota and Santos, 2008)*.

Cha, M., Haddadi, H., Benevenuto, F., & Gummadi, P. K. (2010). Measuring User Influence in Twitter: The Million Follower Fallacy. *ICWSM*, *10*, 10-17.

Chinchor, N., & Robinson, P. (1997, September). MUC-7 named entity task definition. In *Proceedings of the 7th Conference on Message Understanding*.

Chomsky, N. (1956). Three models for the description of language. *Information Theory, IRE Transactions on*, *2*(3), 113-124.

Codina, V., & Ceccaroni, L. (2010). Taking advantage of semantics in recommendation systems. In *Artificial Intelligence Research and Development: Proceedings of the 13th International Conference of the Catalan Association for Artificial Intelligence* (Vol. 220, p. 163). IOS Press, Incorporated.

Cormode, G., & Krishnamurthy, B. (2008). Key differences between Web 1.0 and Web 2.0. *First Monday*, *13*(6).

Crockford, D. (2006). The application/json media type for javascript object notation (json).

Daconta, M. C., Obrst, L. J., & Smith, K. T. (2003). *The Semantic Web: a guide to the future of XML, Web services, and knowledge management*. Wiley. com.

De Saussure, F. (2011). *Course in general linguistics*. Columbia University Press.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. Numerische mathematik, 1(1), 269-271.

Fillmore, C. (1982). Frame semantics. *Linguistics in the morning calm*, 111-137.

Flanagan, D. (2002). JavaScript: the definitive guide. " O'Reilly Media, Inc.".

Gabrilovich, E., & Markovitch, S. (2007). Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *IJCAI* (Vol. 7, pp. 1606-1611).

Garrett, J. J. (2005). Ajax: A new approach to web applications. *Adaprive Path.*

Gilleland, M. (2009). Levenshtein distance, in three flavors. *Merriam Park Software: http://www. merriampark.com/ld.htm.*

Golbreich, C., Wallace, E. K., & Patel-Schneider, P. F. (2009). OWL 2 web ontology language: New features and rationale. *W3C working draft, W3C.*

Gorin, R. E., Willisson, P., Buehring, W., & Kuenning, G. (1971). Ispell, a free software package for spell checking files. *The UNIX community.*

Grady, R. B. (1992). *Practical software metrics for project management and process improvement*. Prentice-Hall, Inc..

Grishman, R. (1997). Information extraction: Techniques and challenges. In*Information Extraction A Multidisciplinary Approach to an Emerging Information Technology* (pp. 10-27). Springer Berlin Heidelberg.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International journal of human-computer studies*, *43*(5), 907-928.

Guarino, N. (1998). Formal Ontology and Information Systems. In Proc. *1st International Conference on Formal Ontologies in Information Systems (FOIS'98)*, pages 3–15. IOS Press.

Guha, R., McCool, R., & Miller, E. (2003). Semantic search. In *Proceedings of the 12th international conference on World Wide Web* (pp. 700-709). ACM.

Harris, S., & Seaborne, A. (2010). SPARQL 1.1 query language. W3C working draft, 14.

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence* (pp. 289-296). Morgan Kaufmann Publishers Inc..

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). SWRL: A semantic web rule language combining OWL and RuleML. *W3C Member submission*, *21*, 79.

Hu, X., & Liu, H. (2012). Text analytics in social media. In *Mining Text Data* (pp. 385-414). Springer US.

Hutchins, W. J., & Somers, H. L. (1992). An introduction to machine translation.

Ide, N., & Véronis, J. (1998). Introduction to the special issue on word sense disambiguation: the state of the art. *Computational linguistics*, *24*(1), 2-40.

Jacobson, I., Booch, G., Rumbaugh, J., Rumbaugh, J., & Booch, G. (1999). *The unified software development process* (Vol. 1). Reading: Addison-Wesley.

Joachims, T. (1998). *Text categorization with support vector machines: Learning with many relevant features* (pp. 137-142). Springer Berlin Heidelberg.

Jurafsky, D. & Martin, J. H. (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (2nd Edition ed.). Prentice Hall Series in Artificial Intelligence.

Kifer, M. (2008). Rule interchange format: The framework. In *Web reasoning and rule systems* (pp. 1-11). Springer Berlin Heidelberg.

Koivunen, M. R., & Miller, E. (2001). W3c semantic web activity. *Semantic Web Kick-Off in Finland*, 27-44.

Laboreiro, G., Sarmento, L., Teixeira, J., & Oliveira, E. (2010). Tokenizing micro-blogging messages using a text classification approach. In*Proceedings of the fourth workshop on Analytics for noisy unstructured text data* (pp. 81-88). ACM.

Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lassila, O., & Swick, R. R. (1999). Resource description framework (RDF) model and syntax specification.

Leal, J., Pinto, S., Bento, A., Oliveira, H., & Gomes, P. (in press). Semeval-2014 task 9: Sentiment Analysis in Twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation* (SemEval-2014).

Li, W., & McCallum, A. (2006). Pachinko allocation: DAG-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning* (pp. 577-584). ACM.

Liu, B. (2010). Sentiment analysis and subjectivity. *Handbook of natural language processing*, *2*, 568.

Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In *Mining Text Data* (pp. 415-463). Springer US.

Mani, I., & Maybury, M. T. (1999). *Advances in automatic text summarization.* the MIT Press.

Manning, C. D., & Schütze, H. (1999). Foundations of statistical natural language processing (Vol. 999). Cambridge: MIT press.

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*,*19*(2), 313-330.

Marrafa, P., Amaro, R., Chaves, R. P., Lourosa, S., Martins, C., & Mendes, S. (2005). WordNet.Pt-Uma Rede Léxico-conceptual do português on-line. *XXI Encontro da Associação Portuguesa de Linguística, Porto, Portugal*.

Maziero, E. G., Pardo, T. A., Di Felippo, A., & Dias-da-Silva, B. C. (2008). A base de dados lexical e a interface web do TeP 2.0: thesaurus eletrônico para o Português do Brasil. In *Companion Proceedings of the XIV Brazilian Symposium on Multimedia and the Web* (pp. 390-392). ACM.

McCallum, A., & Nigam, K. (1998). A comparison of event models for naive bayes text classification. In AAAI-98 workshop on learning for text categorization (Vol. 752, pp. 41-48).

McCallum, A., & Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4* (pp. 188-191). Association for Computational Linguistics.

McGuinness, D. L., & Van Harmelen, F. (2004). OWL web ontology language overview. *W3C recommendation*, *10*(2004-03), 10.

Mendes, P. N., Passant, A., & Kapanipathi, P. (2010). Twarql: tapping into the wisdom of the crowd. In *Proceedings of the 6th International Conference on Semantic Systems* (p. 45). ACM.

Mendoza, M., Poblete, B., & Castillo, C. (2010). Twitter Under Crisis: Can we trust what we RT?. In *Proceedings of the first workshop on social media analytics* (pp. 71-79). ACM.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database*. *International journal of lexicography*, *3*(4), 235-244.

Mitchell, T. M. (1997). *Machine Learning*. New York: McGraw-Hill.

Mitkov, R. (2002). *Anaphora resolution* (Vol. 134). London: Longman.

Motik, B., Grau, B. C., Horrocks, I., Wu, Z., Fokoue, A., & Lutz, C. (2009). Owl 2 web ontology language: Profiles. W3C recommendation, 27, 61.

Moturu, S. (2009). *Quantifying the trustworthiness of user-generated social media content*. Arizona State University.

Naber, D. (2004). Openthesaurus: Building a thesaurus with a web community. *Retrieved January*, *3*, 2005.

Nivre, J. (2005). Dependency grammar and dependency parsing. *MSI report*, *5133*(1959), 1-32.

Oliveira, H. G., Santos, D., Gomes, P., & Seco, N. (2008). PAPEL: a dictionary-based lexical ontology for Portuguese. In *Computational Processing of the Portuguese Language* (pp. 31-40). Springer Berlin Heidelberg.

O'Reilly, T. (2007). What is Web 2.0: Design patterns and business models for the next generation of software. *Communications & strategies*, (1), 17.

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, *31*(1), 71-106.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis.*Foundations and trends in information retrieval*, *2*(1-2), 1-135.

Pierre, N. (2012). Entity extraction: From unstructured text to DBpedia RDF triples.

Pollock, J. T. (2009). *Semantic web for dummies*. John Wiley & Sons.

Prud'Hommeaux, E., & Seaborne, A. (2008). SPARQL query language for RDF. *W3C recommendation*, *15*.

Rajaraman, A., & Ullman, J. D. (2012). *Mining of massive datasets* (pp. 287-320). Cambridge University Press.

Ranchhod, E., Mota, C., & Baptista, J. (1999). A computational lexicon of Portuguese for automatic text parsing. In *Proceedings of SIGLEX99: Standardizing Lexical Resources, 37th Annual Meeting of the ACL* (pp. 74-80).

Real, R., & Vargas, J. M. (1996). The probabilistic basis of Jaccard's index of similarity. Systematic biology, 45(3), 380-385.

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. *In Encyclopedia of Database Systems* (pp. 532-538). Springer US.

Richardson, L., & Ruby, S. (2008). RESTful web services. O'Reilly Media, Inc..

Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., & Edwards, D. D. (1995). *Artificial intelligence: a modern approach* (Vol. 74). Englewood Cliffs: Prentice hall.

Salton, G., & McGill, M. J. (1986). Introduction to modern information retrieval.

Santos, D., & Sarmento, L. (2002). O projecto AC/DC: acesso a corpora/disponibilização de corpora. *Actas do XVIII Encontro da Associação Portuguesa de Linguística*, 705-717.

Sarmento, L. & Nunes, S. (2009). Automatic extraction of quotes and topics from news feeds. In *4th Doctoral Symposium on Informatics Engineering*.

Sarmento, L., Nunes, S., Teixeira, J., & Oliveira, E. (2009). Propagating fine-grained topic labels in news snippets. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 03* (pp. 515-518). IEEE Computer Society.

Santos, D., & Cardoso, N. (2007). Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área.

Schwaber, K., & Beedle, M. (2002). *Agile software development with Scrum* (Vol. 1). Upper Saddle River: Prentice Hall.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, *34*(1), 1-47.

Silva, M. J., Carvalho, P., & Sarmento, L. (2012). Building a sentiment lexicon for social judgement mining. In *Computational Processing of the Portuguese Language* (pp. 218-228). Springer Berlin Heidelberg.

Silva, M. J., Carvalho, P., Costa, C., & Sarmento, L. (2010). Automatic expansion of a social judgment lexicon for sentiment analysis.

Silva, M. J. (2011). Notas sobre a Realização e Qualidade do Twitómetro. University of Lisbon, Faculty of Sciences, LASIGE.

Silva, A. M. R., & Videira, C. A. E. (2001). *UML, metodologias e ferramentas CASE: liguagem de modelação UML, metodologias e ferramentas CASE na concepção e desenvolvimento de software*.

Simões, A. M., & Almeida, J. J. (2001). Jspell. pm–a morphological analysis module for natural language processing. *Actas do XVII Encontro da Associação Portuguesa de Linguística, Lisbon*, 485-495.

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

Sontag, D., & Roy, D. (2011). Complexity of inference in latent dirichlet allocation. In Advances in Neural Information Processing Systems (pp. 1008-1016).

Steyvers, M., & Griffiths, T. (2007). Probabilistic topic models. *Handbook of latent semantic analysis*, *427*(7), 424-440.

Strzalkowski, T., & Harabagiu, S. M. (Eds.). (2006). *Advances in open domain question answering* (Vol. 32). Springer.

Unicode Consortium. (2006). *The Unicode standard, version 5.0*. Addison-Wesley Professional.

Urbani, J., Kotoulas, S., Maassen, J., Van Harmelen, F., & Bal, H. (2010). OWL reasoning with WebPIE: calculating the closure of 100 billion triples. In *The Semantic Web: Research and Applications* (pp. 213-227). Springer Berlin Heidelberg.

Uschold, M., & Gruninger, M. (1996). Ontologies: Principles, methods and applications. *Knowledge engineering review*, *11*(2), 93-136.

Vickery, G., & Wunsch-Vincent, S. (2007). *Participative web and user-created content: Web 2.0 wikis and social networking*. Organization for Economic Cooperation and Development (OECD).

Anexos

Anexo A

Estado da Arte

No presente anexo são introduzidos os conceitos fundamentais para a compreensão da presente da tese. Começa com uma breve introdução ao Processamento de Linguagem Natural (Jurafsky & Martin (2008)) na secção A.1, sendo apresentados alguns dos níveis mais relevantes no âmbito da investigação, seguidos por uma descrição concisa das principais tarefas de PLN e terminando com a exploração dos principais desafios associados à realização destas tarefas no contexto da *media social* (Appelquist et al. (2010)).

Noções relacionadas com a Web Semântica (Berners-Lee, Hendler & Lassila (2001)) são apresentadas na secção A.2, onde é possível visualizar a sua hierarquia de camadas, sendo depois apresentadas as tecnologias mais relevantes presentes em cada uma delas e introduzidos os conceitos de pesquisa e recomendação semântica. Este capítulo é importante uma vez que apresenta vários recursos que permitem representar e inferir novo conhecimento. De seguida, na secção A.3, é efetuada uma apresentação da Web Social, também conhecida como Web 2.0, sendo conferido particular ênfase às redes sociais e apresentada a terminologia habitualmente usada pelos seus utilizadores. São ainda apresentados outros trabalhos de investigação relacionados com a presente tese na secção A.4. Recursos linguísticos e semânticos, bem como ferramentas e bibliotecas relevantes no âmbito desta tese são apresentados na secção A.5 e A.6 respetivamente.

A.1 Processamento de Linguagem Natural

O tópico referente ao Processamento de Linguagem Natural (Jurafsky & Martin (2008)) é frequentemente associado a algumas visões futuristas presentes em filmes como *2001: Odisseia no Espaço*[71], onde o computador HAL 9000 consegue manter uma conversação com humanos usando linguagem natural.

O PLN trata-se de uma área da Inteligência Artificial (Russell et al. (1995)) e da Linguística (De Saussure (2011)) que consiste na aplicação de métodos e técnicas que possibilitam ao computador extrair a semântica presente na linguagem humana e assim possibilitar uma interação homem-máquina.

[71] Referência em *http://www.imdb.com/title/tt0062622/.*

Os computadores encontram-se aptos para compreender e executar todas as instruções presentes em linguagens de programação como Java[72], C[73] ou Python[74], pelo facto de serem linguagens que contêm estruturas lógicas e um conjunto de regras bem definidas, permitindo assim ao computador saber exatamente como proceder a cada instrução. No entanto, o mesmo já não acontece com a linguagem natural, uma vez que uma simples frase poderá conter ambiguidades e interpretações que dependem do contexto, de regras gramaticais ou culturais.

Consequentemente, o objetivo do PLN passa por fornecer aos computadores a capacidade de "entender" um texto, sendo que a maior dificuldade reside na ambiguidade que pode ocorrer a vários níveis: fonético, morfológico, sintático, semântico, pragmático e do discurso. Nas seguintes subsecções será introduzido cada um destes níveis, sendo conferido particular enfâse à análise morfológica, sintática e semântica que são as áreas mais relevantes para a presente tese.

A.1.1 Análise Fonética

A fonologia envolve a análise dos sons presentes no discurso e a sua conversão em símbolos. Este nível não será aprofundado, uma vez que o trabalho desenvolvido lidará unicamente com texto escrito.

A ambiguidade existente a nível fonológico deve-se à pronúncia semelhante de determinadas palavras como *cem* ou *conselho*, uma vez que podem igualmente ser interpretadas como *sem* ou *concelho*.

A.1.2 Análise Morfológica

A análise morfológica trata-se do ramo da Linguística que estuda a estrutura interna das diversas palavras com o intuito de identificar, analisar e descrever cada palavra por si só. Esta análise permite determinar, para cada palavra, as classes gramaticais correspondentes (substantivos, artigos, pronomes, verbos, adjetivos, etc.), identificando a sua forma base (conhecida por lema) e o seu radical[75]. Permite ainda determinar outras características dependentes da sua categoria como o género, número, pessoa e tempo verbal.

No caso dos substantivos e adjetivos, o lema é geralmente representado por uma palavra no género masculino e número singular, enquanto, no caso dos verbos, é representado através do seu infinitivo. Já o radical representa o

[72] Disponível em *https://www.java.com.*
[73] Disponível em *http://www.cprogramming.com/.*
[74] Disponível em *http://www.python.org/.*
[75] O termo utilizado em inglês é *stem.*

fragmento mínimo comum a um conjunto de palavras, também conhecido como sendo o núcleo indivisível da palavra.

Dois exemplos de análise morfológica são ilustrados de seguida:

*As palavras: **professor** (singular, masculino), **professores** (plural, masculino) e **professoras** (plural, feminino), possuem o lema **professor** e o radical **profes**.*

*As palavras **fui** (1ª pessoa do singular no Pretérito Perfeito) e **serei** (1ª pessoa do singular no Futuro) possuem ambas como lema a palavra **ser** e como radical **fui** e **ser**, respetivamente.*

Na análise morfológica é considerada cada palavra em si, independentemente de todas as outras presentes na frase, não sendo por isso tido em conta qualquer tipo de relação com outras palavras presentes na oração. Consequentemente, ao realizar uma análise morfológica, poderá existir ambiguidade pelo facto de algumas palavras representarem mais do que uma categoria gramatical. Por exemplo, a palavra *caminho* representa um substantivo na frase "*O caminho para a escola é longo*", mas na frase "*Eu caminho todos os dias*" representa o verbo caminhar.

De forma a identificar a categoria gramatical num determinado contexto, é necessário efetuar uma análise sintática para que sejam tidas em consideração não só as palavras adjacentes, como também a estrutura no qual a palavra se encontra inserida.

A.1.3 Análise Sintática

A análise sintática tem como objetivo analisar a relação existente entre as várias palavras presentes numa frase, de forma a determinar a sua estrutura gramatical.

As palavras, dependendo da posição e contexto numa frase, podem deter classes gramaticais diferentes, sendo o processo de atribuição da categoria respetiva denominado por Etiquetagem Gramatical (ou em inglês *Part-of-Speech Tagging*) (Jurafsky & Martin (2008)). Através da análise das relações existentes entre palavras vizinhas numa frase é assim possível efetuar a desambiguação, uma vez que, por exemplo, os pronomes possessivos são suscetíveis de ser seguidos por substantivos, enquanto os pronomes pessoais são geralmente seguidos por verbos. Desta forma, recorrendo a classificadores, quer por definição manual das regras ou métodos probabilísticos, é assim possível determinar qual a classe gramatical mais apropriada em função do contexto, o qual não poderia ter sido efetuado somente a nível morfológico. Para além disso a etiquetagem gramatical é ainda bastante útil em outras tarefas de PLN como Extração de Informação (Grishman (1997)) ou Desambiguação do Sentido das Palavras (Ide & Véronis (1998)). Um exemplo de etiquetagem gramatical é o seguinte:

*O/**artigo** livro/**nome** é/**verbo** meu/**pronome***

Dada uma gramática livre de contexto (Chomsky (1956)) é possível derivar uma frase que é representada na forma de uma árvore de constituência (ou em inglês *constituency-based parse tree*) onde é possível identificar as relações entre os constituintes de uma frase (sintagma nominal, sintagma verbal, etc.). A Figura A.1 evidencia a gramática e a árvore sintática gerada (processo designado em inglês por *parsing*) para a seguinte frase: "*O David trouxe o livro*".

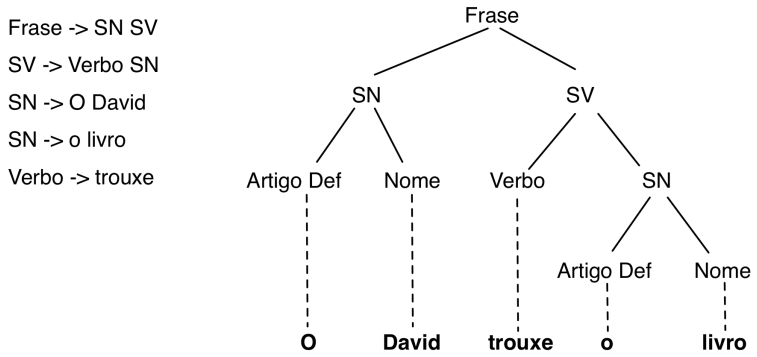

Figura A.1: Exemplo de gramática e da árvore de constituência gerada.

É também possível gerar árvores sintáticas a partir de gramáticas de dependência (Nivre (2005)) sendo, neste caso, a estrutura sintática descrita unicamente em termos das palavras e das relações existentes entre elas. A Figura A.2 apresenta a árvore de dependência (ou em inglês *dependency-based parse tree*) para a frase "*O David trouxe o livro*".

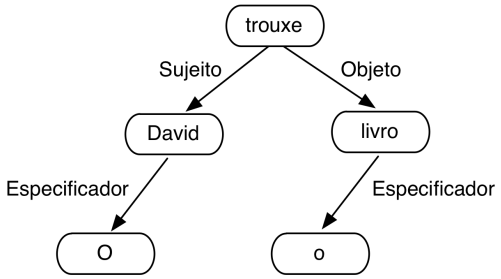

Figura A.2: Exemplo de árvore de dependência[76].

No entanto, também na análise sintática pode existir ambiguidade em situações distintas. Por exemplo, a seguinte frase "*Eu vi o homem com o telescópio*" levanta as seguintes questões: Quem tem o telescópio? Eu ou o homem? Sendo difícil conseguir responder a esta questão sem ter acesso a outros elementos que ajudem a compreender melhor o contexto.

[76] Anotação do *CINTIL-Treebank* (Branco et al.(2011)), estando o guia de anotação disponível em *http://nlxserv.di.fc.ul.pt/buscador/conteudo.html#guidelines*.

A.1.4 Análise Semântica

A análise semântica diz respeito ao estudo do significado da linguagem (sintaticamente bem formada). Para o conseguir é necessário efetuar o mapeamento da linguagem natural numa linguagem formal sem ambiguidade, possibilitando assim a interpretação das palavras, frases e textos, por máquinas.

Existem várias formas de construir representações deste significado entre as quais lógica de predicados (Smullyan (1995)), grafos direcionais ou *frames* semânticas (Filmore (1982)).

A Figura A.3, Figura A.4 e Figura A.5 evidenciam as diferentes representações do significado presente na seguinte frase "*O computador é uma máquina que tem um processador*".

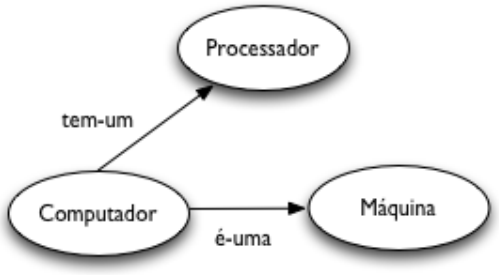

Figura A.3: Representação de conhecimento 1: grafos direcionais.

é-uma(computador, máquina)
tem-um(computador, processador)

Figura A.4: Representação de conhecimento 2: lógica de predicados.

computador
é-uma : máquina
tem-um : processador

Figura A.5: Representação de conhecimento 3: *frame* computador.

Nas representações anteriores existe um relacionamento entre o significado das palavras, frequentemente designado por relacionamento semântico entre unidades lexicais. A entidade *máquina* é detentora de um sentido mais abrangente que *computador*, enquanto *processador* é considerado uma parte relativamente a um todo, sendo neste caso o todo representado pela entidade *computador*. Desta forma, *máquina* é um hiperónimo de *computador* e por sua vez, *computador* é um holónimo de *processador*.

De seguida são formalmente apresentadas algumas das relações existentes, com uma breve explicação das mesmas.

Relações de Hiponímia e Hiperonímia

Uma entidade X é um hipónimo de uma entidade Y se X for um subtipo ou instância de Y. Em suma, um hipónimo é uma palavra cujo significado é hierarquicamente mais específico que o de outra palavra.

Por exemplo, *maçã* e *laranja* estão em relação de hiponímia relativamente a *fruto*, já *animal* é um hiperónimo de *cão*, *gato* e *burro*.

Relações de Holonímia e Meronímia

Relações que envolvem uma parte relativamente a um todo. Holonímia ocorre quando uma entidade é considerada o todo, contendo ou incluindo outra entidade. Por sua vez, o merónimo designa a parte em relação ao todo.

Por exemplo, o holónimo *corpo* é formado pelos merónimos *mão*, *perna*, *cabeça*, etc.

Relações de Sinonímia

Trata-se de uma relação semântica estabelecida entre palavras que possuem um significado equivalente. Esta propriedade implica que os termos sejam substituíveis sem alterar o seu significado.

Por exemplo, *carro* é sinónimo de *automóvel* e *casa* de *lar*.

A.1.5 Análise Pragmática e do Discurso

A pragmática estuda essencialmente os objetivos da comunicação, isto é, a forma como a linguagem é usada para atingir determinados propósitos num determinado contexto. De forma a adquirir informação acerca do contexto é necessário obter conhecimento de todo o discurso e não apenas de frases ou partes específicas do mesmo.

A análise ao nível do discurso estuda as relações entre frases, identificando relações entre unidades maiores que uma frase e procurando desta forma a compreensão do contexto que falta à frase em estudo. Um exemplo deste tipo de análise é a resolução de anáforas, que lida com a identificação de expressões que se referem a uma outra que ocorre na mesma frase ou texto. Por exemplo, na seguinte frase, a palavra *ele* encontra-se associada a *David*.

O David não foi trabalhar. ***Ele*** *está doente.*

A.1.6 Tarefas de PLN

Combinando os vários níveis anteriormente referidos é possível desempenhar um conjunto de tarefas mais complexas, como é o caso das seguintes:

- **Resposta a questões (RQ)** (Strzalkowski & Harabagiu (2006)): Resposta automática a questões colocadas em linguagem natural.

- **Tradução automática[77] (TA)** (Hutchins & Somers (1992))**:** Processo automático de tradução de um texto escrito em linguagem natural noutro idioma.

- **Análise Sintática Superficial[78]** (Manning & Schütze (1999))**:** Divisão de um texto em partes sintaticamente relacionadas, como o caso dos sintagmas nominais e verbais, mas sem especificar a sua estrutura interna ou a sua função principal na frase.

- **Obtenção de informação[79]** (Salton & McGill (1986))**:** Tarefa relacionada com a obtenção de documentos, outro tipo de recursos existentes em linguagem natural ou da informação contida neles, de acordo com uma *query* introduzida pelo utilizador.

- **Reconhecimento de Entidades Mencionadas[80] (REM)** (Chinchor & Robinson (1997))**:** Tarefa de identificação de entidades mencionadas, na sua maioria nome próprios, classificando-as dentro de um conjunto de categorias pré-definidas (pessoas, organizações, localizações, entre outros). Em (McCallum & Li (2003)) são aplicados os modelos matemáticos probabilísticos Conditional Random Fields ou CRF (Lafferty et al. (2001)), com o objetivo de segmentar e etiquetar dados sequenciais em tarefas de REM, obtendo-se resultados bastante satisfatórios.

- **Desambiguação do Sentido das Palavras[81] (DSP)** (Ide & Véronis (1998))**:** Selecionar qual o sentido mais adequado para uma palavra num determinado contexto.

- **Sumarização automática de texto (SAT)** (Mani & Maybury (1999))**:** Criação automática de uma versão curta e sucinta de um texto.

---

[77] O termo utilizado em inglês é *Machine Translation.*
[78] O termo utilizado em inglês é Chunking ou Shallow Parsing.
[79] O termo utilizado em inglês é *Information Retrieval.*
[80] O termo utilizado em inglês é *Named Entity Recognition.*
[81] O termo utilizado em inglês é *Word Sense Disambiguation.*

- **Resolução de Anáforas** (Mitkov (2002)): Identificação de anáforas e determinação das entidades ou expressões ao qual se referem.

- **Classificação de texto** (Sebastiani (2002)): Tarefa de atribuição de uma categoria (a partir de um conjunto pré-definido de categorias) a um texto. Por exemplo, classificar um texto como sendo "spam" ou "não-spam".

- **Extração de Tópicos**[82] (Steyvers & Griffiths (2007)): Obtenção de um conjunto de tópicos que ocorrem numa coleção de documentos, sendo cada tópico composto por um conjunto de palavras frequentemente referenciadas no mesmo contexto. Os pressupostos básicos de qualquer abordagem para extração de tópicos são os seguintes (Aggarwal & Zhai (2012)):

  - Cada documento presente na coleção tem uma probabilidade associada de pertencer a um ou mais tópicos, uma vez que, regra geral um documento refere-se a vários tópicos em diferentes proporções. Por exemplo, um documento pode conter 30% do seu conteúdo sobre economia e 70% acerca de informática.

  - Cada tópico tem associado um vetor de probabilidades, que quantifica a probabilidade associada de cada termo pertencer a esse tópico. Por exemplo, o tópico economia poderá conter os termos "Economia", "Sector económico", "Banco", "Ações", etc.

  O modelo Latent Dirichlet Allocation (LDA) (Blei et al. (2003)) é um dos mais populares e difundidos na tarefa de extração de tópicos (Sontag & Roy (2011)), não requerendo categorização prévia de documentos, sendo os tópicos obtidos unicamente a partir dos textos originais.

- **Análise de sentimentos (**Liu & Zhang (2012)): Visa identificar qual a opinião, ou seja, qual o sentimento (positivo ou negativo), atitude, emoção ou avaliação nutrida relativamente a uma entidade (produto, organização, indivíduo, evento, etc.) ou um aspeto dessa entidade, sob o ponto de vista de um individuo. Pode ser definida através de um tuplo $(e_i, a_{ij}, oo_{ijk}, h_k, t_l)$, onde e_i é o nome da entidade; a_{ij} representa um aspeto da entidade e_i (caso a opinião seja referente à entidade como um todo é usado o aspeto GERAL); oo_{ijk} trata-se da orientação da opinião (positiva, negativa ou neutra, podendo também ser expressa com diferentes níveis de intensidade) acerca do aspeto a_{ij} da entidade e_i; h_k identifica o autor da opinião e t_l identifica a data em que a opinião foi expressa por h_k (Liu

[82] O termo utilizado em inglês é *Topic Modeling.*

(2010)). Para que, numa coleção de documentos que contenham uma opinião se obtenham todos os tuplos, é necessária a realização dos seguintes tarefas (Liu & Zhang (2012)):

- **Extração e agrupamento de entidades:** Extração de todas as expressões referentes a entidades e agrupamento de expressões sinónimas. Por exemplo, *Facebook* e *Face* devem ser agrupados uma vez que representam a mesma entidade.

- **Extração e agrupamento de aspetos:** Extração de todas as expressões referentes a aspetos de entidades e agrupamento de expressões sinónimas. Cada agrupamento representa um aspeto único da entidade, como o aspeto *bateria* da entidade *iPhone*.

- **Extração do autor e data:** Extrair a identificação do autor e a data de publicação do texto.

- **Classificação dos aspetos:** Determinar o sentimento associado a cada aspeto de uma entidade, isto é, se é positivo, negativo ou neutro.

- **Geração de tuplos:** Produzir todos os tuplos na forma $(e_i, a_{ij}, oo_{ijk}, h_k, t_l)$, com base nos resultados obtidos a partir das tarefas enunciadas anteriormente. Por exemplo, caso fosse uma opinião expressa acerca da câmara do *iPhone*, um possível tuplo seria (iPhone, câmara, positivo, João Rodrigues, 16-Setembro-2013).

- **Extração de informação[83] (EI)** (Grishman (1997)): Tarefa que consiste em extrair uma representação estruturada de informação relevante a partir de um texto em linguagem natural (não estruturada). Esta tarefa engloba outras subáreas do PLN como o REM, deteção de relacionamentos entre entidades (Agichtein et al. (2001)), análise de expressões temporais (Allen (1991)), entre outras. Existem várias formas para efetuar esta extração de informação, sendo apresentados de seguida alguns métodos presentes em (Jurafsky & Martin (2008)):

  - **Identificação de frases**: consiste na segmentação de um texto nas várias frases constituintes, através do reconhecimento do início e fim de cada uma das frases. Regra geral este reconhecimento é conseguido através da análise da pontuação utilizada, como o ponto final, exclamação ou interrogação.

[83] O termo utilizado em inglês é *Information Extraction.*

- ***Tokenização***: consiste na divisão de um texto numa sequência de palavras, frases, símbolos ou outro tipo de unidades básicas a serem processadas (denominadas por *tokens* em inglês). Regra geral, a abordagem utilizada para delimitação dos *tokens* passa por segmentar os vários elementos presentes numa frase através dos espaços em branco ou da pontuação existente, como o ponto final.

- **Etiquetagem Gramatical**[84]: consiste em atribuir a cada palavra do texto a sua categoria gramatical respetiva (nome, verbo, adjetivo, advérbio, etc.) tendo em conta a definição ou o contexto (relação existente entre palavras ou frases adjacentes).

- **Lematização:** consiste na obtenção do lema de uma palavra. Desta forma é ignorado o tempo verbal (caso seja um verbo), o género e o plural da palavra. Por exemplo, as palavras *computador* e *computorização*, quando aplicado o processo de lematização, passam à sua forma base (infinitivo do verbo) *computar*.

- ***Stemming*:** consiste na extração do radical da palavra, que corresponde ao seu núcleo indivisível ou raiz. Por exemplo, as palavras *editor*, *editando* ou *editado*, quando aplicado o processo de *stemming*, passam à sua forma raiz *edit*.

A.1.7 O PLN na *Media Social*

A *media social* é definida como "*o uso da Internet e ferramentas eletrónicas com o propósito de compartilhar e discutir informações e experiências com outras pessoas de forma mais eficiente*" (Moturu (2009)) sendo o Twitter[85] e o Facebook[86] alguns dos seus exemplo de maior popularidade. Entre os vários formatos de dados que circulam na *media social*, o texto desempenha um importante papel, uma vez que grande parte da informação presente nestes *sites* é armazenada em formato de texto. No entanto, o Processamento de Linguagem Natural apresenta neste contexto vários desafios, sobretudo devido às características diferenciadas deste tipo de texto (Hu & Liu (2012)):

- **Tamanho reduzido dos textos:** Alguns *sites* limitam o tamanho das mensagens introduzidas pelo utilizador, conduzindo a textos demasiadamente pequenos e consequentemente não providenciando

[84] O termo utilizado em inglês é *Part-of-Speech Tagging*.
[85] Disponível em *https://twitter.com/*.
[86] Disponível em *https://facebook.com/*.

informações suficientes acerca do contexto. Uma das abordagem propostas para tentar resolver o problema associado à escassez de dados consiste em enriquecer o contexto dos segmentos de texto através da exploração de recursos externos a partir de fontes como a Wikipédia[87] (Gabrilovich & Markovitch (2007)).

- **Frases não estruturadas:** A qualidade do texto presente na *media social* pode variar bastante, daí que o processamento deste tipo de mensagens possa ser problemático. É comum, sobretudo em mensagens mais curtas, o uso excessivo de pontuação ("?!?!", "... .."); inclusão de palavras específicas da media ("lol", "ftw", "awww"); abreviações de palavras ("td bem ctg?", "sou de cbr"); uso de *emoticons* (":-)",":-P"); existência de erros de ortografia e por vezes até inclusão de dois idiomas distintos na mesma mensagem, dificultando operações básicas como a *tokenização*.

- **Informação abundante:** Na *media social* existe também muita informação passível de ser extraída além do conteúdo textual. Por exemplo, a rede social Twitter permite que os utilizadores associem à sua mensagem referências para outros utilizadores usando o símbolo arroba (@) ou que façam partilha de mensagens (o chamado *retweet*); à semelhança do Facebook possibilita a conexão a outros utilizadores de forma a receber as suas atualizações. Investigação realizada recentemente tem em conta este tipo de ligações para, por exemplo, avaliar a influência que um utilizador detém no Twitter (Cha et al. (2010)) ou efetuar a distinção entre notícias credíveis e rumores (Mendoza et al. (2010)). Este tipo de informações externas apresenta uma oportunidade para exploração de novas relações.

A.2 Web Semântica

A Web Semântica (WS) é definida como "*uma extensão da Web atual no qual a informação possui um significado bem definido, permitindo a cooperação entre computadores e pessoas*" (Berners-Lee, Hendler & Lassila (2001)).

A WS resulta de um esforço colaborativo liderado pelo *World Wide Web Consortium* (W3C)[88], cujo principal objetivo é melhorar as potencialidades da Web através da criação de padrões e ferramentas que permitam atribuir significado ao conteúdo dos dados presentes na Web (Berners-Lee, Fischetti & Dertouzos (1999)) e assim permitir aos utilizadores encontrar, compartilhar e

[87] Disponível em *http://pt.wikipedia.org/*.
[88] Disponível em *http://www.w3.org/*.

combinar informações mais facilmente. A WS pretende assim promover o compartilhamento de dados estruturados na Web e a interligação desses mesmos dados estruturados - a denominada *Linked Data* (Bizer, Heath & Berners-Lee (2009)), sendo que, quanto maior for o grau de interligação de um dado com outros, maior será a sua relevância e utilidade. Desta forma, visa resolver questões associadas à sobrecarga de informação, uso de sistemas fechados ou proprietários e as limitações associadas aos agregadores de conteúdo (Daconta, Obrst & Smith (2003)).

Para atingir este objetivo é necessário que os computadores sejam capazes de "entender" e dar resposta a pedidos humanos complexos baseando-se no seu significado, sendo para isso necessário que as fontes de informação relevantes se encontrem semanticamente estruturadas (ou seja, é necessário atribuir semântica ao conteúdo das páginas através da descrição de metadados). Desta forma, as máquinas passariam a interpretar questões como "Quantas companhias áreas possuem voos de Lisboa a Barcelona?" ou "Quais as características do MacBook Air mais barato?" que implicam um processo moroso de procura de informação, combinação de resultados e por vezes até de dedução de novas informações e relações entre os conceitos.

De forma a tornar todo o processo praticável e exequível, foram desenvolvidos um conjunto de padrões tecnológicos que possibilitam a identificação de recursos na Web, assim como a representação de informações referentes a esses recursos, levando à criação de informação legível para as máquinas e consequentemente permitindo a partilha global de conhecimento e reutilização de dados. Esse conjunto de padrões forma as diversas camadas em que se divide a arquitetura da WS, visível na Figura A.6 (também conhecida como *Semantic Web Stack*), onde cada camada explora e usa as capacidades das camadas inferiores.

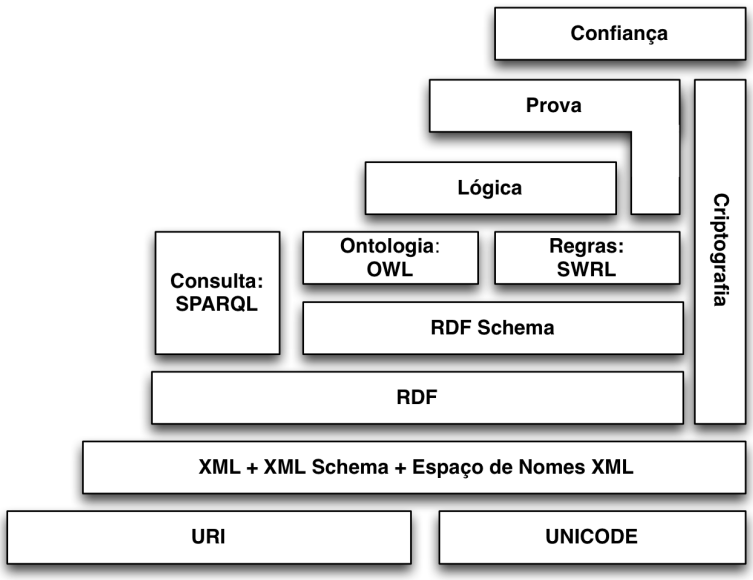

Figura A.6: Hierarquia da Web Semântica, adaptado de (*Berners-Lee* (2005)).

Nas próximas secções serão descritas e exploradas as várias camadas da WS e respetivas tecnologias mais relevantes no contexto desta tese.

A.2.1 Camada URI/Unicode

Trata-se da camada inferior responsável por garantir a interoperabilidade em relação à codificação de caracteres e ao endereçamento e designação de recursos na WS.

Unicode

O Unicode trata-se de um padrão de codificação de caracteres internacionais. Publicado no livro (Unicode Consortium (2006)), cada caractere recebe um identificador único de forma a fornecer uma representação numérica universal, independente da plataforma de software e do idioma utilizado.

URI

O URI (Identificador Uniforme de Recursos) permite identificar univocamente um recurso físico ou abstrato de forma não ambígua, através de um endereço. Um possível exemplo é:

http://www.w3.org/1999/02/22-rdf-syntax-ns#resource

sendo o URI constituído pelos seguintes componentes:

$$\frac{\textbf{\textit{Esquema URI}}}{http} + \frac{\textbf{\textit{Nome Domínio}}}{www.w3.org/1999/02/22 - rdf - syntax - ns} + \frac{\textbf{\textit{Nome Recurso}}}{resource}$$

Por sua vez, o URL (Localizador Uniforme de Recursos) trata-se de um caso específico de URI que representa o endereço de um recurso, como o caso de um documento ou impressora. É formado pela concatenação de caracteres, que permitem identificar o protocolo de acesso ao recurso, o endereço da máquina que disponibiliza o recurso designado e o próprio recurso em questão.

A.2.2 Camada XML

Camada responsável por fornecer a interoperabilidade relativamente à sintaxe de descrição de recursos da WS.

XML

O XML (*Extensible Markup Language*) (Bray et al. (1997)) trata-se de um formato para representação de recursos, composto por dados estruturados e organizados de forma hierárquica. Uma vez que não depende do hardware ou software utilizado, garante também a sua independência relativamente à plataforma usada.

As principais características de um documento XML prendem-se com a existência de elementos (as denominadas *tags* em inglês) que delimitam os dados, atribuindo-lhes um significado específico. A grande vantagem do XML é o facto de ser extensível, podendo cada autor definir os seus próprios elementos, bem como a estrutura do documento. Na Figura A.7 é possível visualizar uma possível representação de um CD de música no formato XML:

```
<CD>
   <TITLE>Wish you were here</TITLE>
   <ARTIST>Pink Floyd</ARTIST>
   <COMPANY>Columbia Records</COMPANY>
   <PRICE>7.80</PRICE>
   <YEAR>1975</YEAR>
</CD>
```

Figura A.7: Exemplo de representação de um CD em XML.

Os elementos XML podem ainda conter atributos que providenciam informação adicional sobre os elementos, como se pode verificar no exemplo abaixo:

<imagem tipo="gif">arvore.gif</imagem>

XML Schema

O *XML Schema* representa uma descrição de um documento XML, normalmente expressa em termos de restrições relativamente ao conteúdo do documento e à sua estrutura hierárquica específica. Por exemplo, através de um *XML Schema* pode definir-se que um determinado elemento (neste caso denominado *autor*) apenas pode ser do tipo *String*:

<xsd:element name="autor" type="xsd:string" />

Espaços de nomes XML

Os espaços de nomes XML[89] são usados para criar identificadores únicos de elementos e atributos num documento XML, através da sua associação aos espaços

[89] O termo utilizado em inglês é *XML Namespaces*.

de nomes identificados por referências do URI. O propósito é evitar eventuais conflitos nas designações, podendo agora existir no mesmo XML outros elementos com o mesmo nome, uma vez que pertencem a espaços de nomes diferentes. De seguida é possível visualizar a declaração de dois espaços de nomes distintos:

xmlns:pessoaxml="http://www.exemplo.pt/xml/pessoa"
xmlns:cidadexml="http://www. exemplo.pt/xml/cidade"

O exemplo anterior define que o prefixo *pessoaxml* se encontra associado ao espaço de nomes *http://www.exemplo.pt/xml/pessoa* e o mesmo acontece com o prefixo *cidadexml,* funcionando assim como uma abreviação de ambos. Desta forma será possível a existência dos seguintes elementos no mesmo documento XML, sem existir qualquer tipo de ambiguidade relativamente ao que o elemento "nome" significa:

< pessoaxml:nome>David</ pessoaxml:nome >
< cidadexml:nome>Coimbra</ cidadexml:nome >

Pode ainda ser declarado um espaço de nomes XML por omissão, através da eliminação do prefixo na sua declaração, de forma a não ser necessário adicionar prefixos a cada subelemento declarado.

A.2.3 Camada RDF

Camada que providencia um modelo de descrição lógica dos dados para representação de informações sobre recursos, de forma a serem facilmente processados por computadores.

A RDF (*Resource Description Framework*) (Lassila & Swick (1999)) trata-se de uma forma de representação de metadados na Internet, baseado em triplos compostos por recurso, propriedade e valor:

- **Recurso**: Entidade que se encontra a ser descrita pela propriedade e valor. A nível gramatical, o recurso é frequentemente representado pelo substantivo que executa a ação. A título de exemplo, na frase "A empresa vende carros" o recurso é a *empresa.*

- **Propriedade**: Estabelece a relação entre entidades e/ou valores**.** Regra geral, numa frase, corresponde ao verbo que modifica o sujeito. Na frase "A empresa vende carros" a propriedade seria o verbo *vende.*

- **Valor**: Refere-se ao recurso (ou valor atómico) referenciado pela propriedade. Habitualmente equivale ao substantivo que é afetado pelo verbo. Na frase "A empresa vende carros" o valor é representado por *carros.*

Em suma, o relacionamento pode ser expresso através da seguinte frase cuja representação gráfica é visível na Figura A.8:

*<**Recurso**> tem <**Propriedade**> com <**Valor**>*

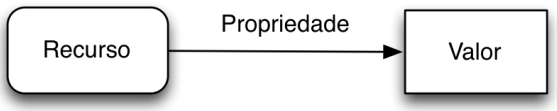

Figura A.8: Representação de Triplo RDF.

É ainda necessário garantir a atribuição de identificadores únicos (designados URIs) para que, recurso, propriedade ou valor possa ser identificado de forma consistente entre vários outros triplos. Pretende-se assim evitar possíveis ambiguidades existentes a nível sintático, como é possível verificar na palavra "banco" que tanto pode referir-se a uma instituição financeira como a mobília, sendo fundamental conseguir distinguir os domínios. É também prática comum na RDF encurtar os URIs, atribuindo um espaço de nomes ao URI base.

Notações RDF

Pelo facto da recomendação original da W3C usar a RDF como modelo de dados e o XML para expressar esses modelos RDF, por vezes associa-se implicitamente ao termo RDF o formato RDF/XML, sendo essa a notação mais popular. No entanto, importa referir que a RDF pode ser representada segundo diferentes tipos de notações, sendo de seguida expresso o mesmo conceito de quatro formas distintas: em linguagem natural, sob a forma de grafo RDF, notação N-Triple e ainda notação RDF/XML, de forma a representar a seguinte frase:

F(1): "O Documento Y (*http://lab.uc.pt/jrod/Documento-Y*) foi criado por João Rodrigues (*http://lab.umb.pt/~jrod*)"

A frase F(1) detém a seguinte estrutura:

- **Recurso**: Documento Y
- **Propriedade**: http://purl.oclc.org/DC/creator[90]
- **Valor**: João Rodrigues

[90] A propriedade *creator* pertence ao espaço de nomes Dublin Core, composto por um conjunto de propriedades pré-definidas para descrição de documentos, que podem ser consultados em *http://dublincore.org/documents/dcmi-terms/*.

Representação da frase F(1) na forma de grafo:

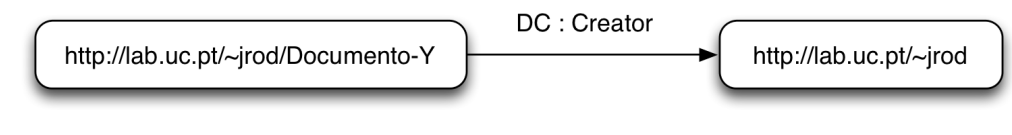

Figura A.9: Representação de frase na forma de grafo RDF.

Representação da frase F(1) segundo o formato N-Triple[91]:

```
< http://lab.uc.pt/jrod/Documento-Y>
< http://purl.oclc.org/DC/creator>
< http://lab.uc.pt/~jrod/>
```

Figura A.10: Representação de frase no formato N-Triple.

Representação da frase F(1) na notação RDF/XML:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.oclc.org/DC/#">
        <rdf:Description rdf:about = "http://lab.uc.pt/jrod/Document-Y">
            <dc:creator rdf:resource="http://lab.uc.pt/~jrod/"/>
        </rdf:Description>
</rdf:RDF>
```

Figura A.11: Representação de frase no formato RDF/XML.

A.2.4 Camada RDF Schema

O RDF Schema (ou RDFS) (Brickley & Guha (2000)) permite expandir a especificação básica da RDF (o qual representa unicamente dados) incluindo o suporte necessário para definir classes e restrições de valores das propriedades, consequentemente estabelecendo uma hierarquia que permite inferir algum conhecimento.

De seguida é descrito parte dos termos mais relevantes no RDF Schema (Brickley et al. (2004)):

- **rdfs:Class** Elemento que contém várias propriedades, representativas de um conjunto de características presentes em elementos similares. Detém uma noção semelhante à classe existente em linguagens de programação orientada a objetos.

[91] O formato N-Triple encontra-se documentado em *http://www.w3.org/TR/rdf-testcases/#ntriples.*

- **rdf:Property** Representa uma propriedade que pertence a uma classe e que possui definidos os valores que toma.
- **rdfs:subclassOf** Especifica uma relação hierárquica entre classes, definindo que a classe A é uma subclasse da classe B. Segue o modelo da herança em que a classe filha herda as propriedades da classe pai.
- **rdfs:subPropertyOf** Especifica a relação hierárquica entre propriedades, definindo que uma propriedade A é uma subpropriedade da propriedade B.
- **rdfs:domain** Define a classe ao qual a propriedade pertence, ou seja, apenas as instâncias da classe definida podem ter essa propriedade.
- **rdfs:range** Define que valores uma propriedade pode tomar, ou seja, de que tipo são os valores que pode assumir.
- **rdf:type** Define o tipo de recurso, relacionando-o com a classe ao qual pertence.

Na Figura A.12 é possível visualizar uma representação gráfica de uma hierarquia de classes e propriedades em RDFS, onde se pode observar as classes *Automóvel* e *Veículo* (sendo *Automóvel* uma subclasse de *Veículo*) e *Modelo* que representa uma propriedade pertencente à classe *Automóvel*.

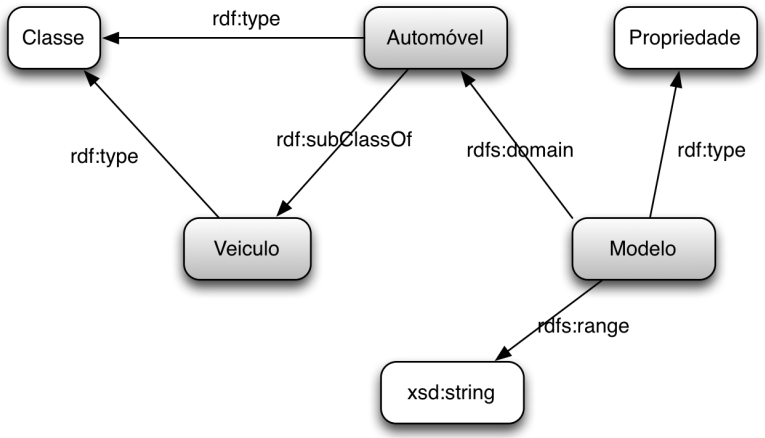

Figura A.12: Representação gráfica de hierarquia de classes e propriedades em RDFS.

A seguir é evidenciada a notação RDF/XML correspondente ao grafo anterior:

```
<rdfs:Class rdf:ID="Veiculo"/>
<rdfs:Class rdf:ID=" Automovel ">
      <rdfs:subClassOf rdf:resource="#Veiculo" />
</rdfs:Class>
<rdf:Property rdf:ID = "modelo">
        <rdfs:domain rdf:resource = "#Automovel" />
        <rdfs:range rdf:resource = "xsd:string" />
</rdf:Property>
```

Figura A.13: Representação de hierarquia de classes e propriedades em RDF/XML.

A.2.5 Camada Ontologia: OWL

Todas as camadas mencionadas nas secções anteriores evidenciam formas estruturadas de representação de conhecimento e são consequentemente interpretáveis por máquinas. A camada ontologia pretende organizar e integrar essas informações, sem a ocorrência de ambiguidades ou conflitos de terminologia, possibilitando assim a definição de relações entre conceitos distintos.

Definição de Ontologia

A definição mais usada para uma ontologia no contexto da WS é a de Gruber "*Uma ontologia é uma especificação explícita de uma conceitualização*" (Gruber (1995)). Ou seja, uma ontologia permite a representação de conhecimento de um dado domínio de forma explícita (uma vez que todos os elementos se encontram claramente definidos de forma a evitar ambiguidades) através da criação de um vocabulário comum para partilha de informação, que pode ser comunicado entre diferentes pessoas ou aplicações.

Em (Uschold & Gruninger (1996)) é constatado que as principais motivações associadas ao uso de ontologias devem-se ao facto de possibilitarem a comunicação entre sistemas e humanos, possibilitarem a realização de inferências, bem como promoverem a reutilização e organização de conhecimento.

Em suma, uma ontologia pode ser considerada como um modelo de dados representativo de um conjunto de conceitos associados a um domínio específico, sendo utilizada para realizar inferências sobre os elementos desse domínio.

Componentes de uma Ontologia

De acordo com (Daconta, Obrst & Smith (2003)) os principais componentes de uma ontologia são os seguintes:

- **Classes**: Representam os conceitos, podendo ser consideradas como um grupo de diferentes indivíduos que partilham características comuns, sejam esses indivíduos objetos concretos ou abstratos. Por exemplo, o conceito *computador* pode ser considerado como uma classe.

- **Instâncias:** Representam os indivíduos pertencentes a uma determinada classe. Pode definir-se que pertencem à classe *computador* as instâncias *Macbook Air*, *Samsung 350V5C*, etc.

- **Propriedades:** As instâncias são descritas através de propriedades, utilizadas para armazenar informação específica para cada instância que representam. Como exemplos de propriedades associadas à classe *computador* pode considerar-se a *memória RAM*, *número de série*, *processador*, etc.

- **Relações:** Descrevem a forma como indivíduos e classes se relacionam. Como exemplo pode considerar-se a relação existente entre as classes *computador* e *hardware*, sendo *computador* uma subclasse de *hardware.* Pode ainda estabelecer-se uma relação entre conceitos e instâncias, como o facto da instância *Macbook Air* ser do tipo *portátil*.

- **Axiomas, regras e restrições:** Conjunto de proposições cujo objetivo é inferir conhecimento que não se encontra indicado de forma explícita na ontologia. Por exemplo, definir como regra que computadores cujo preço seja superior a 1,000 euros são considerados *caros* ou como restrição que o número de núcleos de um processador apenas pode tomar valores entre 1 e 4. Pode ainda definir-se como axioma que um computador deve obrigatoriamente deter um processador.

Tipos de Ontologias

Em termos da natureza dos conceitos e do conteúdo presente nas ontologias, estas podem ser classificadas nos seguintes tipos (Guarino (1998)):

- **Ontologia de nível superior:** Representam conceitos gerais, independentes do domínio (como o tempo, eventos, ações, etc.).

- **Ontologias de domínio:** Especializam conceitos da ontologia de nível superior, usando vocabulário preciso para representação de um domínio genérico (como a medicina ou biologia).

- **Ontologias de tarefa:** À semelhança das ontologias de domínio, tratam-se de uma especialização de uma ontologia de nível superior, no entanto, referem-se a tarefas ou atividades usadas na resolução de problemas (como planos, processos, etc.).

- **Ontologia de aplicação**: Descrevem conceitos que dependem tanto de um domínio particular como de uma tarefa, sendo muitas vezes especializações de ambas as ontologias relacionadas.

Exemplos de Ontologias

Algumas das ontologias mais conhecidas, cujo vocabulário utilizado poderá vir a ser útil no âmbito do trabalho a desenvolver na presente tese, são apresentadas de seguida:

- **FOAF[92] (Friend of a Friend)**: Ontologia que disponibiliza vocabulário descritivo específico para pessoas, atividades e relações sociais na Web. É assim particularmente adequada para descrever pessoas em redes sociais, detendo muitas propriedades relacionadas com atividades ou identidades *online*: *foaf:skypeID*, *foaf:firstName*, *foaf:weblog, foaf:mbox*, etc.

- **SIOC[93] (Semantically-Interlinked Online Communities)**: Ontologia que fornece os principais conceitos e propriedades necessários para descrever informação de comunidades online como fóruns, blogues ou *wikis*. Complementa a ontologia FOAF, focando-se na descrição da contribuição dessas comunidades (publicações, conversações entre utilizadores, respostas publicadas, etc.).

- **OPO[94] (Online Presence Ontology)**: fornece os principais conceitos e propriedades necessárias para descrever informação acerca da presença do utilizador no mundo online (mais concretamente em plataformas de troca de mensagens instantâneas e redes sociais). Pretende representar propriedades alteradas frequentemente, como o seu estado (online, ocupado ou ausente), atividades, localização geográfica, mensagens *default* apresentadas, entre outros.

OWL

O RDF Schema, referido na secção anterior, permite a construção de ontologias embora ainda algo limitadas, devido ao seu reduzido vocabulário para realização de inferências. A linguagem OWL (do inglês *Web Ontology Language*) pretende ultrapassar as limitações do RDF Schema, através da inclusão de novo vocabulário que enriquece a especificação das propriedades e estabelece novos

[92] Disponível em *http://www.foaf-project.org/*.
[93] Disponível em *http://sioc-project.org/ontology*.
[94] Disponível em *http://online-presence.net/ontology.php*.

relacionamentos entre classes, consequentemente possibilitando um maior poder de inferência.

Providencia três sub-linguagens com crescente poder de expressividade, cabendo ao utilizador a escolha do módulo mais adequado de acordo com os requisitos exigidos pela sua aplicação (McGuinness & Harmelen (2004)):

- **OWL Lite:** fornece suporte para os utilizadores que necessitem de uma hierarquia de classificações e restrições simples. As propriedades são definidas usando o termo *owl:datatypeProperty*, que permite expressar relações existentes entre instâncias de classes e respetivos valores, ou *owl:objectProperty* para representar uma relação entre duas instâncias de classes. Na OWL Lite são definidos novos termos que permitem expressar igualdades (*owl:equivalentClass*, *owl:equivalentProperty* e *owl:sameAs*); desigualdades (*owl:differentFrom*, *owl:allDifferent*); características de propriedades (*owl:inverseOf, owl:transitiveProperty, owl:symmetricProperty*, *owl:functionalProperty*); restrições de propriedades (*owl:allValuesFrom, owl:someValuesFrom)* e restrições de cardinalidade (*owl:minCardinality* e *owl:maxCardinality* e *owl:cardinality*). Inclui ainda suporte para a inclusão de ontologias (*owl:ontology)* e a respetiva importação (*owl:imports*).

- **OWL DL:** garante a melhor expressividade tendo em conta que todas as computações são realizáveis e terminam em tempo finito. A OWL DL expande a OWL Lite, não efetuando restrições a nível de cardinalidade (sendo que na OWL Lite os termos *owl:minCardinality*, *owl:maxCardinality* e *owl:cardinality* apenas podem tomar o valor 0 ou 1). As classes podem agora ser descritas através da enumeração dos seus indivíduos constituintes usando *owl:oneOf* ou especificar que são disjuntas através de *owl:disjointWith*. É ainda possível definir classes tendo em conta a presença de valores específicos para as propriedades (através de *owl:hasValue*), ou seja, um individuo será considerado membro de uma classe se pelo menos um valor das suas propriedades for igual ao definido em *owl:hasValue*. Permite ainda a definição de expressões booleanas a partir de combinações de classes, através do uso de *owl:unionOf*, *owl:intersectionOf* ou *owl:complementOf*.

- **OWL Full:** garante expressividade máxima mas sem nenhuma garantia de ser computável. A OWL Full expande a OWL DL permitindo que, por exemplo, um recurso possa, ao mesmo tempo, ser tratado como classe ou indivíduo, daí que garanta expressividade máxima, porém acompanhado de maior exigência a nível computacional, não havendo por isso garantias de ser computável em tempo finito.

Em 2009 foi apresentada a OWL 2 (Golbreich, Wallace & Patel-Schneider (2009)), uma atualização da OWL, que incorpora um conjunto de novas funcionalidades requisitadas pelos utilizadores, incluindo um aumento no poder de expressividade providenciado pela adição de novas propriedades; pela extensão do suporte para os vários tipos de dados (maior conjunto de restrições e definição de facetas que permitem também restringir um tipo de dados a um subconjunto dos seus valores) ou ainda pela extensão da capacidade de anotação das ontologias e axiomas, entre outros novos recursos.

Na OWL 2 é definido um novo conjunto de sub-linguagens (OWL 2 EL, OWL 2 QL e OWL 2 RL), sendo que a escolha de qual o perfil a utilizar dependerá da estrutura da ontologia e do poder de expressividade desejado. De forma muito resumida, a OWL 2 EL é particularmente útil em aplicações que empreguem ontologias contendo um grande número de propriedades e classes; a OWL 2 EL é destinada principalmente a aplicações que trabalhem com grandes volumes de instâncias de dados onde a resposta a questões de consulta seja a tarefa mais importante e, por fim, a OWL 2 RL é empregue em aplicações que exijam raciocínio escalável, sem sacrificar muito do poder de expressividade da linguagem (Motik et al. (2009)).

A.2.6 Camada de Regras: SWRL

A tecnologia da WS presente nesta camada é a linguagem SWRL (*Semantic Web Rule Language*) (Horrocks et al. (2004)) que expande a expressividade da OWL através da adição de regras a uma ontologia. É baseado na combinação da OWL DL e OWL Lite com a sub-linguagem *Unary/Binary Datalog* da RuleML (Boley et al. (2001)).

As regras são apresentadas na forma de uma implicação entre um antecedente e consequente, significando que, caso as condições especificadas no antecedente se verifiquem, então as condições especificadas no consequente devem realizar-se. Em suma, numa sintaxe legível por humanos, uma regra possui a seguinte forma:

$$antecedente \Rightarrow consequente$$

As regras SWRL realizam inferências sobre instâncias OWL, principalmente em termos de classes e propriedades. Por exemplo, uma regra que expresse o facto de que uma pessoa cujo pai tenha um irmão, consequentemente tenha um tio, pode ser representada através de uma combinação das propriedades *temPai* e *temIrmão* que implica a propriedade *temTio* (exemplo retirado de (Horrocks et al. (2004))). Esta regra pode então ser escrita da seguinte forma, onde as variáveis são identificadas usando como prefixo o ponto de interrogação:

$$temPai(?x1, ?x2) \wedge temIrmao(?x2, ?x3) \Rightarrow temTio(?x1, ?x3)$$

A regra anterior foi representada numa sintaxe legível por humanos, contudo é também possível efetuar a sua representação através da *XML Concrete Syntax* e *RDF Concrete Syntax* (Horrocks et al. (2004)).

A linguagem SWRL pode ser usada por um motor de inferência (em inglês *Semantic Reasoner*), isto é, um sistema capaz de inferir consequências lógicas a partir de um conjunto de axiomas e regras, geralmente usando um de dois métodos básicos de inferência:

- **Inferência ascendente[95]:** partindo de um conjunto de axiomas, são usadas regras para deduzir novos factos.
- **Inferência descendente[96]:** partindo de um objetivo, são usadas regras para procurar obter suporte para o objetivo.

As principais motivações associadas ao uso de motores de inferência devem-se ao facto de possibilitarem, não só inferir novo conhecimento, como também permitir dar resposta a questões ou servir como forma de validação de dados.

A.2.7 Camada Consulta: SPARQL

A SPARQL (*SPARQL protocol and RDF Query Language*) (Prud'Hommeaux & Seaborne (2008)) trata-se de uma linguagem de consulta de triplos RDF, sendo reconhecida como uma das tecnologias chave da WS.

Ao contrário das convencionais bases de dados relacionais, em que são usadas chave primárias e estrangeiras para estabelecer um relacionamento entre diferentes tabelas, a RDF usa identificadores únicos (URIs). Consequentemente uma base de dados RDF (em inglês denominada *triple store*) pode então vincular-se a qualquer outra base de dados independente para obtenção de dados.

As *queries* SPARQL são definidas de acordo com um padrão semelhante à representação de um triplo RDF, exceto no facto de, quer o recurso, quer a propriedade ou o valor, poderem representar uma variável (identificada na *query* através de um ponto de interrogação (?) ou dólar ($)). Deste modo, pretende verificar-se a existência de uma possível correspondência entre o padrão definido na *query* e os triplos existentes na base de dados RDF. Caso existam triplos que satisfaçam esta condição (ou seja, que as variáveis possam ser substituídas pelos valores correspondentes) são então retornados os valores vinculados às variáveis.

Considerando a existência de uma base de dados RDF com referências de livros (título do livro e preço respetivo) pode ser efetuada a seguinte *query* em

[95] O termo utilizado em inglês é *Forward Chaining*.
[96] O termo utilizado em inglês é *Backward Chaining*.

SPARQL para obtenção de todos os títulos dos livros presentes (exemplos adaptados de (Prud'Hommeaux & Seaborne (2008))):

```
PREFIX dc: http://purl.org/dc/elements/1.1/
SELECT ?title
WHERE { ?book dc:title ?title }
```

Figura A.14: Pesquisa SPARQL.

Podem ainda ser aplicados filtros na pesquisa, de forma a serem retornados apenas os que, por exemplo, detêm um título que se inicia por "SPARQL".

```
PREFIX dc: http://purl.org/dc/elements/1.1/
SELECT ?title
WHERE { ?book dc:title ?title
            FILTER regex(?title,"^SPARQL")}
```

Figura A.15: Pesquisa SPARQL com filtro.

A linguagem SPARQL providencia quatro formas de consulta distintas (Prud'Hommeaux & Seaborne (2008)):

- **SELECT:** Retorna os valores vinculados às variáveis.
- **CONSTRUCT:** Retorna um grafo RDF, produzido através da substituição das variáveis na *query* de pesquisa.
- **ASK:** Retorna um valor booleano a indicar se foram encontrados resultados ou não.
- **DESCRIBE:** Retorna um grafo RDF que descreve os recursos obtidos.

A versão 1.1 do SPARQL (Harris & Seaborne (2010)), apresentada em 2013, providencia um conjunto de novas funcionalidades, entre as quais, a definição de uma linguagem para especificar e executar atualizações de grafos RDF numa base de dados de triplos; inclusão de funções de agregação habitualmente utilizadas em *queries* a base de dados relacionais (como o "COUNT", "SUM" ou "AVG"), bem como permite que os resultados obtidos através de uma *query* SPARQL sejam representados usando os formatos JSON (Crockford (2006)), CSV (*Comma Separated Value*) ou TSB (*Tab Separated Value*).

A.2.8 Outras Camadas

Camada Lógica, Prova e Confiança

De acordo com (Pollock (2009)) o objetivo da camada superior de Lógica é descrever uma lógica matemática formal que permita conciliar os vários modelos semânticos provenientes das várias partes (RDF, RDFS, OWL e SPARQL) numa teoria de modelos global e consistente.

O mesmo autor afirma que a camada Prova destina-se a fornecer uma maneira matematicamente correta de explicar que inferências e que regras de negócio levaram à aceitação de uma determinada conclusão ou recomendação. Por fim, a camada Confiança providencia um meio para avaliar os dados em termos da sua confiabilidade, para que possamos saber se deve "confiar" nas provas ou não.

Camada Criptografia

A função desta camada é incorporar mecanismos de segurança, como a assinatura digital (Artz & Gil (2007)), que garantam a confiabilidade da informação através da sua certificação.

A.2.9 Pesquisa Semântica

De acordo com a IDC[97], cerca de 56% do tempo de um trabalhador é despendido a pesquisar informação. Além disso, um total de 9% do tempo é consumido em pesquisas que conduzem a resultados não desejados, providenciando uma margem considerável para melhoria.

A pesquisa semântica (Guha et al. (2003)) surgiu com o intuito de melhorar os tradicionais métodos de procura, que não efetuam qualquer tipo de interpretação da *query* (como a pesquisa baseada unicamente em palavras-chave[98]) através do uso da semântica. O seu principal objetivo é aperfeiçoar os resultados de pesquisa essencialmente de duas formas (Guha et al. (2003)):

- Regra geral, os resultados provenientes de tradicionais métodos de pesquisa são apresentados sob a forma de uma listagem de documentos ou páginas Web. A pesquisa semântica pretende expandir o conjunto de resultados apresentados através da inserção de dados relevantes obtidos a partir da WS (do conjunto de dados estruturados e interpretáveis por máquinas).

[97] Informação retirada de *http://www.google.com/enterprise/solutions/prof_services/search_roi.html*.
[98] O termo utilizado em inglês é *Keyword-Based Search*.

Por exemplo, caso um utilizador efetue uma pesquisa por um ator seria igualmente interessante o acesso a informação relativa aos filmes em que participou (ou que irá participar), a sua foto, prémios que recebeu, etc.

- As *queries* de pesquisa muitas vezes representam conceitos cuja identificação pode ser bastante útil na filtragem dos resultados apresentados ao utilizador, de modo a mostrar-lhe apenas os dados mais relevantes e significativos. Assim, pretende-se que, através da interpretação semântica dos termos presentes na *query* de pesquisa se consiga melhorar substancialmente os resultados da pesquisa apresentados ao utilizador. Por exemplo, na seguinte frase:

  Setembro 2013 Apple notícias

  Seria detetada a presença da data *Setembro de 2013*, a organização *Apple* e o tópico *notícias*, daí que poderiam ser obtidos a partir de uma base de dados todos os triplos referentes a notícias da Apple ocorridas no mês de Setembro a partir da seguinte *query* em SPARQL:

```
SELECT ?x
WHERE { ?x hasType "noticias"
        ?x hasOrganization #Apple
        ?x hasDate "Setembro 2013"}
```

Em suma, este tipo de pesquisa pretende explorar e interpretar a semântica associada aos dados, com o intuito de entender qual o significado do conteúdo e com isso gerar resultados mais significativos de acordo com aquilo que se supõe ser a real intenção de pesquisa por parte do utilizador. Ou seja, através do significado associado aos termos pesquisados, pretende-se assim gerar resultados mais relevantes no âmbito da pesquisa efetuada.

A.2.10 Recomendação Semântica

Muitas vezes somos forçados a tomar decisões mesmo sem ter conhecimentos suficientes acerca das alternativas existentes, sendo influenciados através daquilo que ouvimos de outras pessoas, artigos de opinião, pesquisas efetuadas, etc. Os sistemas de recomendação visam precisamente filtrar toda a informação apresentada, providenciando ao utilizador recomendações baseadas no conhecimento que é adquirido em *background* acerca das suas preferências ou nas características do próprio conteúdo, permitindo assim uma apresentação de itens ainda não explorados anteriormente e que à partida lhe suscitem interesse e sejam relevantes dado o seu perfil.

Estes sistemas são cada vez mais populares, podendo ser encontrados em *sites* como a *Amazon*[99], onde são sugeridos produtos ao utilizador com base no que outros utilizadores adquiriram juntamente com o item selecionado; no *Netflix*[100] onde são recomendados outros filmes ao utilizador baseados nas suas visualizações e *ratings* anteriores; ou mesmo no *Pandora*[101], onde são propostos outros músicos com características semelhantes aquelas que o utilizador por norma ouve.

Existem dois grandes tipos de sistemas de recomendação (Rajaraman & Ullman (2012)):

- **Baseados em conteúdo:** sistema que se foca nas características particulares de um determinado item, de forma a recomendar um conjunto de outros itens que possuem propriedades similares. Pode ainda ser sugerido conteúdo tendo em conta as preferências evidenciadas pelo utilizador no passado. Num cenário de recomendação de filmes, caso o utilizador tenha comprado muitos livros de ficção científica, teria consequentemente disponíveis recomendações de outros livros também do género ficção científica.

- **Filtragem Colaborativa:** sistema que foca na relação entre utilizadores e itens. Consiste na recomendação de itens que outros utilizadores, com gostos semelhantes, mostraram igualmente interesse no passado. Será assim expresso na forma de: Se um utilizador gostou dos itens X e Y um outro utilizador que goste de X pode também gostar de Y.

Por sua vez, os sistemas de recomendação semântica são caracterizados pela incorporação de conhecimento semântico no seu processo, de forma a melhorar a qualidade da recomendação feita. Desta forma, o uso de tecnologias da WS permite efetuar a representação formal dos dados e isso confere algumas vantagens, nomeadamente pelo facto de ser possível contextualizar dinamicamente os interesses do utilizador num determinado domínio e permitir também efetuar a inferência de nova informação a partir dos dados recolhidos (Codina & Ceccaroni (2010)).

A.3 Web Social

A Web Social (também conhecida como Web 2.0) (Appelquist et al. (2010)) pode ser definida como o conjunto de relações sociais que os utilizadores estabelecem e

[99] Disponível em *http://www.amazon.com/*.
[100] Disponível em *http://www.netflix.com/*.
[101] Disponível em *http://www.pandora.com/*.

constroem entre si através da Internet, englobando também ferramentas e *sites* que são desenvolvidos de forma a fomentar esta interação social.

Inicialmente a Web tratava-se de um meio bastante limitado onde grande parte do conteúdo era estático, daí que o utilizador se limitasse unicamente a receber informação, não existindo qualquer tipo de interatividade com a página. A Web Social procura inverter essa tendência, de modo a que o utilizador deixe de ser um mero recetor de informação e passe também ele próprio a ser o emissor, permitindo-lhe expressar preferências e opiniões que passam a ser tidas em conta.

Em suma, pretende-se o desenvolvimento de aplicações que aproveitem esta inteligência coletiva, sendo dado um ênfase na colaboração e partilha de informação. Algumas das ferramentas introduzidas pela Web Social são sumarizadas nas seguintes categorias (Hu & Liu (2012)):

- **Redes Sociais:** Representam o meio onde as pessoas se podem conectar de acordo com valores e objetivos comuns, permitindo a partilha de informação e conhecimento. As redes sociais podem adquirir diferentes propósitos, algumas funcionando como redes de relacionamentos (por exemplo o Facebook[102]), outras como redes profissionais (como o LinkedIn[103]) e algumas até servindo propósitos comunitários.

- ***Wikis*:** Tratam-se de ferramentas colaborativas que permitem a edição coletiva de documentos, visando a simplicidade e rapidez na troca de conhecimento. Uma das ferramentas mais conhecidas e utilizadas é a Wikipédia[104].

- **Blogues:** Versão abreviada do termo *weblog*, usado para descrever páginas Web que permitem uma atualização rápida de conteúdo. Este espaço pode ser utilizado para várias finalidades – pode ser usado como diário pessoal; para promover e partilhar informação entre pessoas que detêm interesses comuns; expressar opiniões ou simplesmente publicar notícias. São bastante atrativos pela facilidade que oferecem na sua criação e manutenção, visto existirem bastantes ferramentas disponíveis para essa finalidade como o Wordpress[105] ou o Blogger[106].

- **Microblogging:** Forma de *blogging* onde o tamanho das mensagens inseridas pelos utilizadores é limitado, usualmente suportando até 200 caracteres ou menos. O serviço de *microblogging* mais popular é o

[102] Disponível em *https://www.facebook.com/.*
[103] Disponível em *https://pt.linkedin.com/.*
[104] Disponível em *http://pt.wikipedia.org/.*
[105] Disponível em *http://pt.wordpress.org/.*
[106] Disponível em *http://www.blogger.com/.*

Twitter[107], embora o Facebook possua também recursos de *microblogging*, denominado de atualização de estado.

- **Notícias Sociais** (ou *Social News*)**:** Páginas Web que encorajam os seus utilizadores a efetuarem a submissão de conteúdo que é depois classificado de acordo com a sua popularidade. As histórias mais populares ganham maior destaque e relevância na página principal, fomentando a chamada inteligência coletiva. Um dos exemplos mais populares é o Reddit[108].

A.3.1 Redes Sociais

Dada a importância das redes sociais no âmbito do projeto, serão de seguida abordadas duas das mais populares – o Facebok e o Twitter. Além de uma breve explicação do funcionamento e propósito de cada uma, serão ainda definidos alguns dos termos mais utilizados e conhecidos pelos seus utilizadores.

Facebook

O Facebook é atualmente uma das mais populares redes sociais cujo objetivo passa por conectar amigos, família e colegas. Dados obtidos do DMR[109], referentes ao segundo trimestre de 2013, apontam para a existência de cerca de 1,26 mil milhões de utilizadores ativos, mais de mil milhões de publicações diárias, tendo sido divulgadas cerca de 350 milhões de fotos.

Cada utilizador registado pode criar o seu perfil onde é possível visualizar informação sobre si mesmo (nome, morada, idade, etc.), os seus amigos, atividades e interesses, fotos e as suas publicações (texto, fotos ou vídeo). Um utilizador pode apenas visualizar o perfil de outro, caso ambos sejam amigos, sendo que, para que isso aconteça, é necessária a autorização prévia por parte do recetor do pedido de amizade.

Existe a possibilidade de criação de grupos para utilizadores que partilham *hobbies* ou interesses comuns, bem como de eventos (usados na organização de festas, concertos, etc.). Todas as interações do utilizador são registadas no seu *feed* de notícias, que é partilhado em tempo real com todos os seus utilizadores amigos. Alguns dos termos mais comuns usados no Facebook são:

- ***Timeline*:** Página de perfil do utilizador que contém todas as suas publicações.

[107] Disponível em *https://twitter.com/*.
[108] Disponível em *http://pt.reddit.com/*.
[109] *Digital Marketing Ramblings*, disponível em *http://goo.gl/hQ5ENF*.

- **Amigo:** Alguém ao qual o utilizador se encontra conectado e que, por isso, pode visualizar o seu perfil e inserir publicações na sua *timeline*. Desta forma estabelece-se uma relação bidirecional, onde um utilizador se conecta a outro que lhe concede uma conexão.

- ***Feed* de notícias:** Área onde é possível visualizar em tempo real um fluxo contínuo de todas as atividades publicadas pelos amigos do utilizador.

- **Atualização de estado:** Trata-se de uma curta publicação no Facebook onde o utilizador partilha conteúdo de texto, imagens ou vídeo com os amigos.

- **Gosto (*like*):** Forma do utilizador expressar que gosta do conteúdo publicado.

Twitter

O Twitter é uma das redes sociais de *microblogging* mais amplamente utilizada, sendo geradas, de acordo com dados referentes ao terceiro trimestre de 2013, cerca de 500 milhões de publicações diárias[110]. Nesta rede cada utilizador publica as suas mensagens (limitadas unicamente a um texto de 140 caracteres) que poderão depois ser visualizadas pelos utilizadores que o "seguem", permitindo interagir em tempo real tanto com pessoas que pertencem à sua rede como fora dela. Alguns dos termos mais habitualmente usados no Twitter são:

- ***Tweet*:** Publicação de texto limitada a 140 caracteres.

- **Seguir (*Follow*):** Relação unidirecional em que um utilizador "segue" outro, podendo consequentemente visualizar todos os seus *tweets* publicados.

- ***Timeline*:** Ordenamento temporal dos *tweets* (provenientes das pessoas que o utilizador segue) sendo listados primeiro os que foram inseridos mais recentemente.

- **Seguidor (*Follower*):** Designação dada a alguém que segue um utilizador, ou seja, que pode visualizar todos os *tweets* partilhados.

- ***ReTweet* (RT):** Republicar um *tweet* de outro utilizador, de modo a que os seus seguidores o possam visualizar.

[110] Informação retirada de *Digital Marketing Ramblings*, disponível em *http://goo.gl/pbXxHh.*

- **Menção (@):** Utilizado para referenciar um outro utilizador num *tweet*. Por exemplo, ao publicar um *tweet* contendo a menção *@dmcrodrigues*, não só o utilizador alvo é notificado, como o *tweet* passa a ser listado na sua *timeline*.

- ***Hashtag* (#):** Utilizado para definição de um ou mais tópicos associados a um *tweet*, para que, quando algum utilizador efetue uma procura, rapidamente consiga encontrar os resultados pretendidos. Por exemplo, quando selecionado *#iPhone* são listados todos os *tweets* que contêm essa *hashtag*.

- ***Link*:** Inclusão de um URL no *tweet*.

A.4 Trabalhos Relacionados

Neste capítulo são apresentados trabalhos de investigação que se encontram relacionados com a presente tese.

São exploradas ferramentas que permitem efetuar extração e armazenamento de dados na forma de triplos RDF, assim como recursos e metodologias que auxiliam na realização de algumas tarefas da PLN - desde tarefas mais simples como *tokenização*, a tarefas mais complexas como extração de tópicos e análise de sentimentos.

Todos os trabalhos apresentados fazem uso de publicações provenientes de redes sociais, blogues e notícias online exclusivamente em português, à exceção da secção de extração de triplos RDF, onde o foco não é o idioma português, essencialmente devido à escassez de trabalhos desenvolvidos nesta área.

A.4.1 Extração de Triplos RDF

Em (Mendes et al. (2010)) é apresentado o Twarql, um projeto que visa efetuar a extração de informação a partir de publicações do Twitter, estruturando-a sob a forma de triplos RDF e consequentemente possibilitando a consulta de dados através de *queries* SPARQL. O principal objetivo deste projeto é lidar com o vasto conjunto de informação associado ao elevado número de publicações existentes no Twitter, permitindo realizar a monitorização e filtragem de conteúdo de acordo com os interesses do utilizador. O Twarql possui um módulo apenas para obtenção de dados dos *tweets*, como o nome do autor, menções, data de publicação do *tweet* e informação geográfica. São ainda obtidas as *hashtags* e a sua descrição respetiva usando o serviço disponibilizado no *site http://tagdef.com/*. Recorrendo a este serviço é possível determinar que a *hashtag #fomof* significa *fear*

of missing out on football ou que *#yolo* significa *you only live once*, possibilitando assim complementar a *hashtag* com informação adicional importante para a descrição e compreensão do seu conteúdo. Além disso são extraídos os URLs presentes no texto e é efetuada extração de entidades com base num dicionário previamente populado com entidades obtidas da DBpedia[111] (base de dados de triplos construída a partir de informação estruturada presente na Wikipédia[112]). Depois de efetuada a extração de informação, os *tweets* são representados sob a forma de triplos RDF, sendo usadas as seguintes ontologias: (1) **FOAF** (Friend Of A Friend) - para representação dos utilizadores e da sua rede social; (2) **SIOC** (Semantically-Interlinked Online Communities) - para representação das publicações do Twitter; (3) **OPO** (Online Presence Ontology) - para descrever informação acerca da presença do utilizador nas redes sociais, de forma a perceber a sua situação atual, como a sua localização geográfica corrente; (4) **MOAT**[113] (Meaning Of A Tag) - para atribuição de significado às *tags* presentes nas publicações do Twitter, fazendo para isso uso de URIs provenientes de bases de conhecimento como a DBpedia, para enriquecimento dos dados. O facto de serem usados recursos pertencentes a entidades já conhecidas, que detém um contexto, representa uma escolha importante, visto os *tweets* conterem um texto bastante reduzido e consequentemente não providenciarem um contexto informativo suficiente. A combinação das várias ontologias permite integrar os vários elementos envolvidos numa aplicação de *microblogging*, possibilitando que o Twarql aceda a informação de diversos domínios, através da análise de diferentes tipos de dados - isto é, utilizadores, tópicos, tempo, localização dos *tweets*, etc. Esta funcionalidade é assim possível graças à anotação de *tweets* usando diferentes modelos e bases de conhecimento (cada uma com o seu propósito definido). O Twarql trata-se de uma ferramenta *open-source* desenvolvida na linguagem de programação Java e encontra-se disponível para *download* em *http://twarql.sf.net*.

Em (Pierre (2012)) é apresentada uma *framework* para extração de triplos RDF (compostos por sujeito, predicado e objeto) a partir de texto não estruturado, presente nos artigos da Wikipédia. O objetivo é que estes triplos sejam depois adicionados à ontologia da DBpedia de forma a enriquecer o seu conteúdo, visto esta apenas conter a informação estruturada da Wikipédia, descartando a informação que poderia ser extraída a partir do texto dos artigos. Numa primeira fase são removidas todas as anotações e etiquetas HTML de forma a obter-se unicamente o texto do artigo. Na segunda fase, é extraído conhecimento do texto usando *frames* semânticas (Filmore (1982)), ou seja, através de um conjunto de

[111] Disponível em *http://dbpedia.org/*.
[112] Disponível em *http://pt.wikipedia.org/*.
[113] Disponível em *http://www.w3.org/2001/sw/wiki/MOAT*.

predicados e argumentos anotados de acordo com a nomenclatura Proposition Bank ou PropBank (Palmer et al. (2005)) usando a *framework* Athena[114]. Esta nomenclatura associa a cada predicado um conjunto de sentidos, por exemplo, *nascer* encontra-se associado a 6 sentidos distintos denominados *nascer.01*, *nascer.02*, ..., *nascer.06*. Associados a cada um destes sentidos, encontram-se argumentos específicos - *nascer.02*, por exemplo, possui associados dois argumentos principais: A_0 que representa a mãe e A_1 que representa o filho. Assim para cada predicado são identificados até 6 argumentos denotados por $A_0, A_1 \ldots A_5$, sendo também identificados modificadores de predicados, como os adjuntos adverbiais de tempo e de lugar. Estes papéis são fundamentais para realizar a extração de entidades, pois permitem a identificação precisa da localização e período temporal do acontecimento. Na nomenclatura PropBank é usada a notação A_0 para representar um argumento que descreve agentes ou causadores, enquanto argumentos que usam a notação A_1 descrevem entidades que são afetadas por uma ação. Desta forma, ambos os argumentos A_0 e A_1 podem ser considerados como sujeitos RDF no triplo. Depois de efetuada a extração do sujeito, os restantes argumentos são examinados de forma a descobrir potenciais objetos. São então procurados os argumentos temporais (representados por AM-TMP), de localização (AM-LOC) ou entidades mencionadas no texto. Depois de definido qual o sujeito e objeto, é efetuada a extração dos URIs providenciados no próprio artigo, uma vez que a partir deles é possível estabelecer uma correspondência direta entre as entidades da DBpedia e os recursos da Wikipedia. No entanto, caso estes não sejam disponibilizados diretamente no artigo, é usada a ferramenta Wikifier[115] para obtenção dos *links* da Wikipédia associados a entidades ainda não anotadas. Se ainda assim não for possível obter o URI associado a uma entidade, é efetuada a resolução da correferência (isto é, da relação existente entre dois ou mais termos que referenciem a mesma entidade) recorrendo à ferramenta Standford CoreNLP[116]. O objetivo é "propagar" o *link* de uma entidade anteriormente identificada ao argumento para o qual ainda não foi extraído nenhum *link* e que remeta para a mesma entidade. Esta situação surge frequentemente em argumentos compostos por um único pronome. Um possível exemplo de extração de entidades é visível de seguida:

Kurt Cobain (nasceu a 20 de Fevereiro de 1967) foi compositor e músico.

Cobain nasceu em Washington, filho de um mecânico e de uma empregada doméstica.

As frases anteriores são anotadas de acordo com a notação PropBank:

[114] Disponível em *http://semantica.cs.lth.se/athena*.
[115] Disponível em *http://cogcomp.cs.illinois.edu/page/software_view/Wikifier*.
[116] Disponível em *http://www-nlp.stanford.edu/downloads/dcoref.shtml*.

$$\frac{\boldsymbol{A_1}}{Kurt\ Cobain}\left(\frac{\boldsymbol{nascer.02}}{nasceu}\ \frac{\boldsymbol{AM-TMP}}{20\ de\ Fevereiro\ de\ 1967}\right)\ foi\ compositor\ e\ músico.$$

$$\frac{\boldsymbol{A_1}}{Cobain}\ \frac{\boldsymbol{nascer.02}}{nasceu}\ em\ \frac{\boldsymbol{AM-LOC}}{Washington},\ filho\ de\ um\ mecânico\ e\ de\ uma\ empregada.$$

Sendo extraídos os seguintes triplos:

<dbpedia:Kurt_ Cobain> <nascer.02.AM-TMP> "20-02-1967"

<dbpedia:Cobain> <nascer.02.AM-LOC> <dbpedia:Washington>

A última fase do processo tem como objetivo efetuar o mapeamento dos predicados, descritos usando a nomenclatura Propbank, no espaço de nomes (ou *namespace*) da DBpedia, para que os triplos possam ser adicionados nessa ontologia. A abordagem para deteção dos possíveis mapeamentos existentes, passa pela extração dos triplos da DBpedia para os quais existe correspondência entre o sujeito e predicado. Desta forma é assim possível criar um conjunto de mapeamentos, através da generalização de URIs da DBpedia (do sujeito e objeto), para as 43 classes de topo da ontologia da DBpedia. Por exemplo, considerando a seguinte frase:

Cobain casou com Courtney Love a 24 de Fevereiro de 1992.

É extraído o triplo:

<dbpedia:Kurt_ Cobain> <casar.01.A1> <dbpedia:Courtney_ Love>

que corresponde ao seguinte triplo existente na DBpedia:

<dbpedia:Kurt_ Cobain > <dbpedia-owl:cônjuge> <dbpedia:Courtney_ Love>

sendo efetuada a generalização do sujeito e objeto:

<dbpedia-owl:Pessoa> <casar.01.A1> <dbpedia-owl:Pessoa>

que é depois mapeada em:

<dbpedia-owl:cônjuge>

É assim criado o mapeamento visível na Tabela A.1. Desta forma passa a ser possível efetuar o mapeamento de frases que descrevam a mesma relação, mas para os quais não exista ainda o triplo correspondente na DBpedia, possibilitando a sua adição à ontologia e consequente enriquecimento da mesma.

Sujeito	**Predicado**	**Objeto**	**Mapeamento**
dbpedia-owl:Pessoa	casar.01.A1	dbpedia-owl: Pessoa	dbpedia-owl:cônjuge

Tabela A.1: Exemplo de mapeamento da ontologia DBpedia.

A.4.2 Tarefas de PLN

Em (Laboreiro et al. (2010)) é apresentada uma ferramenta que permite efetuar a *tokenização* de conteúdo gerado pelo utilizador através de redes sociais de *micro-blogging* como o Twitter ou o Facebook. Revela-se bastante importante dado que, contrariamente ao texto tradicional, o conteúdo presente nas redes sociais tende a ser bastante irregular e ambíguo possibilitando a cada autor criar o seu próprio estilo de escrita. Consequentemente, os *tokenizadores* construídos a partir um de conjunto de regras manualmente definidas, têm que lidar com bastantes condições e exceções o que torna difícil a sua construção e manutenção. De modo a contornar este problema, foi criada uma coleção de 2500 *tweets*, manualmente tokenizados de acordo com determinadas regras – menções, *hashtags*, URLs, datas e horas são consideradas palavras regulares, a pontuação é agrupada num único *token* ("...","?!?" ou "!") tal como os acrónimos ("U.K." ou "U.S.A.") e sendo também corrigidos erros ortográficos frequentes e normalizadas as palavras com um dicionário de pitês[117]. O objetivo é treinar um classificador de texto binário SVM (Support Vector Machine) (Joachims (1998)) que determine para um conjunto de pontos de decisão existentes na mensagem (que correspondem geralmente aos pontos que cercam os caracteres não-alfanuméricos) se deve ser inserido (ou não) um espaço em branco para delimitação de cada *token*. Os módulos foram desenvolvidos em PERL, sendo disponibilizados no seguinte *site* *http://labs.sapo.pt/2011/11/sylvester-ugc-tokenizer/*.

(Sarmento & Nunes (2009)) apresenta o Verbatim, um sistema para extração e classificação de citações obtidas a partir de RSS *feeds* provenientes de notícias em português. Disponibiliza uma interface Web[118] onde é possível visualizar as várias citações, entidades mencionadas, bem como os vários tópicos associados às citações. Desta forma são continuamente recolhidos *feeds* provenientes dos principais meios de comunicação social, sendo apenas consideradas as notícias que explicitamente mencionam o orador e a sua citação respetiva, como a seguinte frase: *O vice-presidente do PSD, Aguiar-Branco, considerou que o primeiro-ministro perdeu uma oportunidade de "falar a verdade ao país"*, sendo a extração efetuada através do uso de expressões regulares. De seguida é verificada a existência de duplicados, através do agrupamento de notícias que referenciem o mesmo orador e que apresentem uma semelhança considerável entre os textos presentes em ambas as citações, usando como medida de similaridade o coeficiente de Jaccard (Real & Vargas (1996)). As citações que possuam um grau de semelhança superior a um limiar pré-definido são removidas, uma vez que não

[117] Linguagem frequentemente adotada pelos utilizadores das redes sociais, que contém inúmeras abreviações, com o objetivo de permitir uma comunicação mais rápida.
[118] Disponível em *http://irlab.fe.up.pt/p/verbatim/*.

fornecem qualquer informação relevante adicional. É depois efetuado um processamento para extração de tópicos associados a cada notícia, obtendo-os de acordo com o seguinte padrão, muitas vezes usado nos títulos das notícias: "*tópico: título da notícia*" (como por exemplo, "Operação Furacão: Acusações atrasadas" ou "Música: Morreu Ron Ashton"). O objetivo é usar estes dados para treinar um classificador SVM que atribua um tópico a cada uma das notícias/citações.

No entanto, a abordagem apresentada para classificação de notícias apresenta alguns desafios devido sobretudo à especificidade de alguns dos tópicos encontrados, uma vez que podem existir várias fontes a publicar notícias acerca do mesmo assunto detendo uma abrangência/domínio diferente - para uma notícia acerca de um jogo de futebol da liga dos campeões poderão ser obtidos os tópicos de *Desporto*, *Futebol*, *Liga dos campeões* ou então o próprio nome das equipas. Pode ainda acontecer que os tópicos obtidos expressem diferentes perspetivas acerca do mesma notícia, como por exemplo, uma notícia acerca de ataques de piratas na Somália poderá conter no título *Piratas*, *Ataque Pirata* ou então referenciar a área geográfica *Somália* ou *Oceano Índico*. Assim sendo, notícias bastante semelhantes poderão deter diferentes tópicos com abrangências e perspetivas diferentes acerca do mesmo evento. Em (Sarmento et al. (2009)) é proposto um método que pretende reduzir o impacto associada a esta fragmentação, atribuindo automaticamente vários tópicos adicionais a notícias que apresentem conteúdo semelhante, de forma a melhorar ou completar a descrição de cada notícia com alternativas igualmente válidas que expressem diferentes perspetivas. No entanto importa referir que o Verbatim não se encontra atualmente disponível para *download* ou visualização.

Em (Silva (2011)) é apresentada uma ferramenta denominada Twitómetro que permitiu aferir, durante as eleições parlamentares de 2011, o sentimento geral dos portugueses relativamente a cada um dos cinco líderes partidários. A polaridade é obtida a partir da análise das publicações dos utilizadores na rede social Twitter, com base num conjunto de recursos linguísticos, nomeadamente léxicos de polaridade, padrões léxico-sintáticos e regras de inferência. Dos léxicos aplicados fazem parte um conjunto de lemas adjetivais, nominais e verbais, bem como de expressões idiomáticas portuguesas (como "abriu os olhos" ou "não gosta de ouvir as verdades") aos quais é associada uma polaridade positiva, negativa ou neutra. Estes léxicos são depois invocados por um conjunto de regras léxico-sintáticas que determinam a polaridade dos predicadores - caso um adjetivo classificado com polaridade positiva seja antecedido por um advérbio de negação como "não" ou "nunca" (por exemplo, "nunca foi honesto"), será então analisado com polaridade negativa. Foi também tido em conta o tipo de léxico usado nas redes sociais como os *emoticons* ou a pontuação excessiva, que regra geral denotam sarcasmo ou ironia. Também o uso de certas alcunhas para os líderes partidários (como Pinócrates) são usadas para atribuir uma polaridade negativa

à mensagem. É efetuado um cálculo diário do indicador de sentimento para cada político através da diferença entre o número de mensagens positivas e negativas. O Twitómetro encontra-se disponível para visualização na seguinte página Web *http://legislativas.sapo.pt/2011/twitometro/.*

Foi também desenvolvido o Twitteuro[119] para medição da popularidade de cada uma das equipas e jogadores do Euro 2012 através das publicações inseridas no Twitter. O Twitteuro processa em tempo real todos os *tweets* que contenham a *hashtag* #Euro2012 e identifica menções efetuadas à equipa e aos jogadores. Quanto maior for o número de *tweets* que mencionem uma equipa ou um jogador, maior será a sua popularidade. O Twitteuro encontra-se disponível para visualização em *http://twitteuro.sapo.pt/.*

Em (Silva et al. (2010)) é apresentada uma metodologia para expansão automática de léxico sentimental, mais concretamente de adjetivos que expressam opiniões relativamente a entidades humanas. É usado um procedimento de dois passos: primeiro é efetuada uma identificação de potenciais adjetivos associados a um sujeito, através da definição manual de um conjunto de regras léxico-sintáticas. De forma a automatizar o processo, é depois usado um classificador binário (treinado com adjetivos previamente categorizados) para efetuar uma distinção entre os adjetivos que se referem a um sujeito humano/não humano. No segundo passo o objetivo é expandir o léxico de polaridade, através da exploração de um grafo construído a partir de vários tesauros[120] (PAPEL (Oliveira et al. (2008)), TeP (Maziero et al. (2008)) e DicSin[121]), que contém sinónimos dos adjetivos previamente obtidos. É calculada a distância de cada adjetivo, cuja polaridade é desconhecida, aos já previamente classificados (através do algoritmo de *Dijkstra* (Dijkstra (1959))) sendo usada essa distância para treinar um classificador automático de polaridade.

A.5 Recursos Linguísticos

Nesta secção, são apresentados alguns dos recursos linguísticos que se encontram relacionados com o trabalho, sendo depois evidenciada uma secção mais vocacionada para as redes sociais, devido às especificidades necessárias para lidar com este tipo de texto. Estes recursos poderão vir a ser utilizados para treino ou teste dos módulos de PLN desenvolvidos no decurso do projeto.

[119] Disponível em *http://twitteuro.sapo.pt.*

[120] Dicionário onde se encontram agrupadas listas de palavras de acordo com o seu significado.

[121] Disponível em *http://extensions.openoffice.org/en/project/dicsin-dicionario-de-sinonimos-protugues-brasil-2013.*

Os recursos léxico-semânticos encontram-se também relacionados com o trabalho uma vez que permitem a representação de conhecimento. Podem ser usados em várias tarefas de PLN como desambiguação do sentido das palavras, reconhecimento de entidades mencionadas ou extração de informação.

A.5.1 Corpora Linguística

Nesta subsecção serão apresentados vários corpus linguísticos para o Português. Grande parte dos corpus descritos podem ser encontrados através do projeto AC/DC[122] (Acesso a Corpus/Disponibilização de Corpus), que surgiu da necessidade de juntar os recursos disponíveis num único ponto de rede e dessa forma facilitar a comparação e reutilização de material.

Existe ainda uma subsecção denominada Análise de Sentimentos, onde são apresentados recursos especialmente importantes para extração e classificação de sentimentos e opiniões em português.

CETEMPúblico

O CETEMPúblico (Corpus de Extratos de Textos Eletrónicos MCT/Público) trata-se de um corpus de aproximadamente 180 milhões de palavras que inclui o texto de cerca de 2.600 edições do Público[123] entre os anos de 1991 e 1998. Os vários extratos encontram-se classificados por semestre e secção do jornal do qual provêm. A anotação do corpus foi realizada automaticamente pelo analisador sintático para o português PALAVRAS (Bick (2000)), segundo a rotina estabelecida pelo projeto AC/DC (Santos & Sarmento (2002)). Foi criado pelo projeto que deu origem à Linguateca, após a assinatura de um protocolo entre o Ministério da Ciência e da Tecnologia (MCT) português e o jornal Público em Abril de 2000.

Coleção Chave

A coleção chave foi criada a partir de 726 edições do jornal Português Público e 730 edições do jornal Brasileiro Folha de São Paulo[124]. É o resultado da participação da Linguateca na organização do CLEF[125] a partir de 2004. Em Abril de 2007 foi disponibilizada uma versão anotada sintaticamente pelo PALAVRAS (Bick (2000)) e em Janeiro de 2010 foi também disponibilizada uma

[122] Disponível em *http://linguateca.pt/ACDC/.*
[123] Disponível em *http://www.publico.pt/.*
[124] Disponível em *http://www.folha.uol.com.br/.*
[125] Disponível em *http://www.clef-initiative.eu.*

versão anotada automaticamente no que se refere a entidades mencionadas pelo REMBRANDT (Cardoso (2008)).

Coleções Douradas HAREM

O HAREM (Avaliação e Reconhecimento de Entidades Mencionadas) (Santos & Cardoso (2007)) trata-se de uma avaliação conjunta[126] na área de REM em Português, organizado pela Linguateca[127]. O objetivo desta iniciativa é avaliar o sucesso na identificação e consequente classificação automática de nomes próprios na língua portuguesa. O processo de avaliação conjunta iniciou-se com a criação da denominada coleção dourada HAREM, que consiste num conjunto de textos marcados com as entidades mencionadas, identificadas e classificadas corretamente segundo um conjunto de diretivas[128] aprovadas por todos os participantes. Assim, do primeiro evento do HAREM, resultou a coleção dourada do Primeiro HAREM, realizado em 2005, bem como a coleção dourada usada no miniHAREM de 2006. Do segundo evento HAREM (uma avaliação mais abrangente do que a anterior que possui novas diretivas para marcação de entidades) resultou a coleção dourada do Segundo HAREM que teve lugar em 2008, bem como a coleção TEMPO. O texto presente nestas coleções foi obtido a partir de diversas fontes de informação, desde blogues a notícias, entrevistas, artigos de opinião, entre outros, de forma a abranger diversas áreas.

Natura/Minho

O corpus Natura/Minho[129] consiste numa coleção de textos retirados de uma série de edições do jornal Diário do Minho, criado no âmbito do projeto Natura[130]. O corpus contém notícias completas, separadas em várias edições e marcadas pela data, tendo sido feito um esforço para retirar artigos que contenham publicidade, resolução de palavras cruzadas ou de desporto, assim como artigos repetidos. Este recurso teve origem em 2001, foi automaticamente anotada pela primeira vez em 2008, encontrando-se atualmente na quinta versão.

---

[126] Modelo de avaliação em que vários grupos comparam, com base num conjunto de tarefas consensuais, o progresso dos seus sistemas numa dada área, usando um conjunto de recursos comuns e uma métrica consensual.

[127] Disponível em *http://www.linguateca.pt/*.

[128] Disponível em *http://www.linguateca.pt/primeiroHAREM/harem_classificacao.html*.

[129] Disponível em *http://www.linguateca.pt/acesso/desc_corpus.php?corpus=NATMINHO*.

[130] Disponível em *http://natura.di.uminho.pt/wiki/doku.php*.

Dicionário de Pitês

O neologismo pitês refere-se à linguagem maioritariamente adotada pelos utilizadores que frequentam redes sociais, com o objetivo de permitir uma comunicação mais rápida. Frequentemente as vogais são suprimidas, sílabas de palavras substituídas apenas por uma letra e são usados caracteres para representar emoções (os denominados *emoticons*). Desta forma, o dicionário de pitês revela-se um recurso bastante útil pois permite obter a tradução correta deste tipo de vocabulário, possibilitando, a título de exemplo, que "hj" seja traduzido em "hoje", "gnt" em "gente" e "at+" em "até mais tarde". Existem vários recursos deste tipo na Web, sendo o mais completo e estruturado para a língua Portuguesa denominado o Dicionário de Internetês[131].

Lista de *Stopwords*

As *stopwords* representam um conjunto de palavras frequentemente utilizadas e que em algumas tarefas (como extração de tópicos ou pesquisa semântica) são propositadamente ignoradas por não acrescentarem informação útil. São exemplos destas palavras "*as*", "*e*", "*os*" e "*de*". Um dos recursos mais completos considerado na definição desta lista é disponibilizado pela Universidade de Neuchâtel[132], identificando 356 *stopwords* em português.

Análise de Sentimentos

SentiLex-PT[133] (Silva et al. (2012)) trata-se de um léxico de sentimentos para o português, constituído por adjetivos, nomes, verbos e expressões idiomáticas ao qual é atribuída uma polaridade positiva, negativa ou neutra. A informação de polaridade associada às entradas foi, na maioria dos casos, manualmente atribuída. Algumas entradas adjetivais foram automaticamente classificadas pela ferramenta JALC (Judgment Analysis Lexicon Classifier), desenvolvida pela equipa do projeto para este fim. As formas flexionadas dos verbos e das expressões idiomáticas, bem como os respetivos atributos morfológicos, foram extraídos do *LABEL-Lex-sw*[134] (Ranchhod et al. (1999)), um léxico de palavras disponível para o português.

[131] Disponível em *http://linguadedoido.blogspot.pt/2008/07/dicionrio-de-internets.html.*
[132] Disponível em *http://members.unine.ch/jacques.savoy/clef/.*
[133] Disponível em *http://dmir.inesc-id.pt/project/SentiLex-PT_02.*
[134] Disponível em *http://label.ist.utl.pt/en/downloads_en.php.*

Análise da Corpora Linguística

Como visto anteriormente, existem vários recursos de corpora que podem ser usados para extração de informação. Na Tabela A.2 é possível visualizar uma análise geral comparativa dos vários recursos. Mesmo não sendo plausível compará-los, a ideia principal é adquirir uma visão mais ampla de todos os corpus linguísticos, sendo indicada qual a linguagem utilizada (Portuguesa ou Brasileira), tipo de anotação (manual, automática ou semiautomática), disponibilização (domínio público, académica ou proprietária) e uma breve descrição do recurso.

Recurso	**Linguagem**	**Anotação**	**Disponibilização**	**Descrição**
CETEMPúblico	PT	Automática (PALAVRAS)	Académica[135] (Não permitida comercialização)	Jornal Público, dividido em extratos, 1991-1998
Coleção Chave	PT/BR	Automática (PALAVRAS)	Académica[136] (Não permitida comercialização)	Jornais Público e Folha de São Paulo, 1994-1995
Coleções Douradas HAREM	PT/BR	Manual	Domínio Público (Não tem licença)	Excertos de diversas fontes de informação com as entidades mencionadas marcadas
Natura/Minho	PT	Automática	Proprietária (Recurso não se encontra acessível)	Excertos do jornal regional Diário do Minho
Dicionário de Pitês	PT	Manual	Domínio Público (Não tem licença)	Léxico usado em redes sociais
Lista de *Stopwords*	PT	Manual	Domínio Público (Licença BSD 2-Clause)	Lista de *stopwords* em português
Sentilex-PT	PT	Semiautomática	Domínio Público (Licença CC-BY)	Léxico de sentimentos

Tabela A.2: Análise comparativa da corpora linguística.

A.5.2 Recursos Léxico-Semânticos

Esta subsecção introduz um conjunto de recursos, vocacionados para a língua Portuguesa, que permitem representar conhecimento de diferentes formas. O OpenThesaurusPT é o exemplo de recurso mais simples que apenas lida com relações de sinonímia, sendo depois apresentadas outras bases de conhecimento mais abrangentes e com diferentes características.

[135] Disponível em *http://www.linguateca.pt/cetempublico/informacoes.html.*
[136] Disponível em *http://www.linguateca.pt/CHAVE/.*

OpenThesaurusPT

O OpenThesaurusPT (Naber (2004)) trata-se de um dicionário de sinónimos para a língua portuguesa. Por esta razão, a sua unidade base é o *synset*, que representa um conjunto de palavras que detêm o mesmo significado, ou seja, que se encontram ligadas através de uma relação de sinonímia.

Toda a comunidade poderá participar na sua criação e correção de eventuais erros existentes, sendo necessário o registo prévio dos utilizadores que o pretendam fazer e também o respeito das regras definidas, disponibilizadas em FAQ[137]. Este recurso detém atualmente 13.256 palavras e 4.102 diferentes grupos de sinónimos (*synsets*).

WordNet.PT

A WordNet.PT[138] consiste numa base de dados de conhecimento linguístico de Português, desenvolvida no Centro de Linguística da Universidade de Lisboa[139] pelo Grupo de Computação do Conhecimento Léxico-Gramatical[140].

A WordNet.PT segue as linhas gerais da *wordnet* de Princeton (Miller (1990)), que se trata da primeira base de dados de conhecimento linguístico em que o significado lexical é representado através de uma rede de relações lexicais e conceptuais, sendo o significado de cada unidade lexical derivado da sua posição na rede (Marrafa et al. (2005)). A unidade básica da *wordnet* é o conceito, representado por um conjunto de sinónimos (denominados *synsets*). Cada *synset* contém todas as lexicalizações referentes a um conceito, constituindo um nó da rede (por exemplo, as expressões "carro" e "automóvel" encontram-se incluídas no mesmo *synset*). Ao contrário dos dicionários convencionais, nas *wordnets* o sentido é inferido a partir das relações lexicais e conceptuais que são estabelecidas.

Algumas das relações existentes no WordNet.PT encontram-se presentes na Tabela A.3.

[137] Disponível em *http://openthesaurus.caixamagica.pt/faq.php.*
[138] Disponível em *http://www.clul.ul.pt/clg/wordnetpt/index.html.*
[139] Disponível em *http://www.clunl.edu.pt/PT/home.asp.*
[140] Disponível em *http://www.clul.ul.pt/clg/index.html.*

Relações geral/específico	
x é um hiperónimo (é supertipo) de	x é instanciado por
x é um hipónimo (é um tipo) de	x é a instanciação de
Relações todo/parte (holónimo/merónimo)	
x tem como parte	x tem como membro
x é parte de	x é membro de
x tem como parte distinta	x tem como porção
x é parte distinta de	x é porção de
x tem como substância/material	x tem como localização
x é substância/material de	x é localização de
Relações de categorização	
x é caracterizável por	x é característica de
x caracteriza quanto a	x tem como característica ser
x está relacionado com	

Tabela A.3: Algumas das relações presentes na WordNet.PT.

PAPEL

O PAPEL (Palavras Associadas Porto Editora) (Oliveira et al. (2008)) trata-se de um recurso lexical para o português, onde são definidos um conjunto de relações entre os termos (como SINONIMO_DE, HIPERONIMO_DE, PARTE_DE, etc.) extraídos de forma automática a partir do Dicionário da Língua Portuguesa da Porto Editora. A versão 3.5 contém relações entre cerca de 102 mil palavras diferentes e cerca de 191 mil instâncias de relações (ou triplos). Destas, cerca de 83 mil são de sinonímia e cerca de 49 mil de hiponímia. Trata-se de um recurso público e grátis encontrando-se aberto para subsequente melhoria pela comunidade.

DBpedia PT

A DBpedia[141], trata-se de um projeto cujo objetivo é a extração de dados estruturados da Wikipédia[142], como é o caso das *infoboxes* (tabelas existentes no canto superior direito de artigos que visam apresentar um resumo dos seus aspetos mais relevantes), categorias, imagens, coordenadas geográficas e *links* para páginas Web externas. A base de conhecimento da DBpedia, na versão 3.9, é composta por 2.46 mil milhões de dados (triplos RDF) dos quais 470 milhões foram extraídos da Wikipédia versão inglesa e 1.98 mil milhões foram extraídos de outras linguagens. Uma das grandes vantagens da DBpedia é o facto de disponibilizar uma

[141] Disponível em *http://DBpedia.org.*
[142] Disponível em *http://pt.wikipedia.org/.*

grande base de dados sobre os mais variados assuntos, em várias línguas (incluindo o Português presente na DBpedia PT[143]), compartilhados sob uma licença de domínio público e que se encontra em constante desenvolvimento. A principal motivação da DBpedia é que a grande quantidade de informação presente na Wikipédia seja usada de maneiras interessantes que inspirem novos mecanismos de interligação de dados e melhoramento da própria enciclopédia. A informação da DBpedia encontra-se armazenada no formato RDF, permitindo a realização de consultas à base de conhecimento através de SPARQL.

Análise dos Recursos Léxico-Semânticos

Foram anteriormente apresentados alguns recursos que possibilitam a extração de conhecimento semântico, cada um com a sua própria estrutura e forma de representação de conhecimento, possuindo todos suporte para a língua portuguesa. Apesar de não ser plausível efetuar uma comparação entre os vários recursos, uma vez que são significativamente diferentes e cumprem também diferentes propósitos, pretende-se com a Tabela A.4 e Tabela A.5 apresentar algumas das suas características gerais de forma a adquirir uma visão mais ampla de todos eles.

A Tabela A.4 apresenta a forma de construção de cada recurso (manual, automática ou semiautomática) e a sua disponibilização respetiva (domínio público, académico ou privado). Na Tabela A.5 é possível visualizar alguns dados relativos à estrutura de cada recurso – número de nós (isto é, instâncias da estrutura base), número de termos únicos (número de termos não repetidos que o recurso detém), o número de arestas (relações que ligam os nós) e o número de relações definidas (tipos de relações).

Recurso	**Disponibilização**	**Construção**
OpenThesaurusPT 2.0	Domínio Público (Licença GPL v3.0)	Manual
WordNet.PT v1	Domínio Público (Não tem licença)	Manual
PAPEL v3.5	Domínio Público (Não tem licença)	Automática
DBpedia PT v3.9	Domínio Público (Licença GPL v3.0 e CC-SA v3.0)	Semiautomática

Tabela A.4: Análise comparativa dos recursos léxico-semânticos (disponibilização e construção).

[143] Disponível em *http://pt.DBpedia.org/*.

Recurso	**Estrutura**	**Nós**	**Termos Únicos**	**Instâncias de Relações**	**Tipos de relações**
OpenThesaurusPT 2.0	synsets	4,102	13,256	-	1
WordNet.PT v1	synsets	8,715	10,931	11,584	7
PAPEL v3.5	termos	102,000	-	191,000	32
DBpedia PT v3.9	instâncias	493,944	-	4,489,235	620

Tabela A.5: Análise comparativa dos recursos léxico-semânticos.

A.6 Ferramentas e Bibliotecas

Nesta secção são apresentadas algumas ferramentas e bibliotecas, que serão analisadas com o objetivo de compreender a sua possível aplicação no âmbito do projeto.

A.6.1 PLN para Português

As ferramentas e bibliotecas exploradas nesta subsecção permitem auxiliar no desenvolvimento de módulos de PLN, sendo exclusivas para a língua Portuguesa, uma vez que se trata do idioma alvo na presente tese.

Existe também uma subsecção, onde são apresentadas ferramentas mais específicas para lidar com texto proveniente de redes sociais, dada as particularidades presentes neste tipo de discurso e que é necessário igualmente ter em conta no projeto.

OpenNLP

O OpenNLP[144] trata-se de uma biblioteca em Java para Processamento de Linguagem Natural. Suporta as tarefas mais comuns de PLN como a *tokenização*, segmentação por frases, etiquetagem gramatical, *Chunking* ou reconhecimento de entidades mencionadas, que são depois usadas como base na construção de tarefas de maior complexidade. Encontra-se disponível para vários idiomas, nomeadamente o Português, embora o número de funcionalidades disponibilizadas seja inferior (apenas é disponibilizado o *tokenizador*, segmentação de frases e REM).

[144] Disponível em *http://opennlp.apache.org/*.

DBpedia Spotlight

A DBpedia Spotlight[145] é uma ferramenta que permite efetuar a anotação automática de menções referentes a recursos da DBpedia presentes num texto. Desta forma possibilita o reconhecimento de entidades (como por exemplo *Steve Jobs*) e desambiguação das mesmas, atribuindo-lhes um identificador único. Encontra-se disponível para vários idiomas, inclusive o Português.

jSpell

O jSpell (Simões & Almeida (2001)) é um analisador morfológico derivado do corretor ortográfico *open-source* ispell (Gorin et al. (1971)), indicando para uma dada palavra as suas características morfológicas (o género, número, categoria gramatical, etc.). O jSpell possui vários modos de funcionamento, desde a herdada pelo ispell como corretor ortográfico, por acesso à sua interface online[146] ou como biblioteca C ou Perl. O seu principal desenvolvimento tem sido com vista à utilização para a língua portuguesa, embora existam dicionários em outras línguas.

LX-Suite

O LX-Suite (Branco & Silva (2004)) é composto por um conjunto de ferramentas de PLN, das quais se salientam o LX-Chunker que segmenta as frases e parágrafos existentes num texto; LX-Tokenizer, que separa um texto em itens lexicalmente relevantes e o LX-Tagger que se trata de um etiquetador que atribui uma categoria morfossintática a cada palavra, detetando também expressões multipalavra. Esta ferramenta foi desenvolvida usando o etiquetador gramatical MXPOST[147] e um corpus anotado manualmente, composto por aproximadamente 600.000 palavras. É possível visualizar alguns exemplos de etiquetas usadas na Tabela A.6.

O conjunto de ferramentas da LX-Suite foi desenvolvido na Universidade de Lisboa pelo NLX-Grupo de Fala e Linguagem Natural[148].

[145] Disponível em *https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki.*
[146] Disponível em *http://natura.di.uminho.pt/webjspell/jsol.pl.*
[147] Disponível em *http://www.inf.ed.ac.uk/resources/nlp/local_ doc/MXPOST.html.*
[148] Disponível em *http://nlx.di.fc.ul.pt/.*

Etiqueta	**Categoria**	**Exemplos**
ADJ	Adjetivo	Bom, brilhante, eficaz
ADV	Advérbio	Hoje, já, sim, felizmente
CARD	Cardinal	Zero, dez, cem, mil
Expressões multipalavra		
LADV1...LADVn	Advérbios multipalavra	De facto, em suma
LCJ1...LCJn	Conjunções multipalavra	assim como, já que

Tabela A.6: Exemplos de etiquetas usadas no LX-Tagger.

MALLET

O MALLET[149] consiste numa biblioteca Java para Processamento de Linguagem Natural, permitindo, entre outras tarefas, analisar uma larga coleção de textos não categorizados, de forma a extrair tópicos que os caracterizam (tarefa denominada extração de tópicos, ou em inglês *topic modeling*). O conjunto de ferramentas para extração de tópicos, disponibilizado pelo MALLET, implementa o modelo Latent Dirichlet Allocation (Blei et al. (2003). Esta ferramenta disponibiliza ainda a implementação do modelo Conditional Random Fields (Lafferty et al. (2001)), bastante utilizado em tarefas de REM e possui também algoritmos para classificação de documentos com base em exemplos previamente categorizados (como o Naive Bayes (McCallum & Nigam (1998)).

Hunspell

O Hunspell trata-se de um corretor ortográfico e analisador morfológico, usado pelo LibreOffice[150], OpenOffice[151], Mozilla Firefox[152], entre outros. Apesar de ter sido originalmente concebido para a língua húngara, disponibiliza hoje suporte para inúmeras outras línguas, inclusive o português. É baseado no corretor ortográfico MySpell[153], sendo consequentemente compatível com os dicionários MySpell. O Hunspell é disponibilizado na forma de uma biblioteca em C++, existindo também uma *interface* de integração em Java denominada HunspellJNA[154].

[149] Disponível em *http://mallet.cs.umass.edu/.*
[150] Disponível em *http://www.libreoffice.org/.*
[151] Disponível em *http://www.openoffice.org/pt/.*
[152] Disponível em *http://www.mozilla.org/.*
[153] Disponível em *http://ispell-gl.sourceforge.net/agal/myspell.html.*
[154] Disponível em *https://github.com/dren-dk/HunspellJNA.*

Redes Sociais

Sylvester UGC Tokenizer (Laboreiro et al. (2010)) trata-se uma biblioteca capaz de efetuar a tarefa de *tokenização* a partir de textos provenientes de redes de *microblogging* como o Twitter. Desta forma, consegue lidar com várias particularidades existentes neste tipo de texto, preservando URLs, *emoticons*, pontuação pouco usual, que a tornam assim uma ferramenta bastante atrativa para lidar sobretudo com textos curtos, provenientes de redes sociais.

Análise de ferramentas PLN

Como é possível verificar existem várias ferramentas, cada uma com o seu propósito específico. Desta forma, a sua comparação direta torna-se difícil, uma vez que a sua grande maioria diz respeito a diferentes tarefas de PLN. Na Tabela A.7 são apresentadas algumas das características mais relevantes acerca de cada ferramenta, nomeadamente a linguagem de programação em que foram implementadas, os idiomas que suportam, o tipo de disponibilização da ferramenta (domínio publico, proprietário ou académico) e principais tarefas possíveis de serem realizadas.

Recurso	**Linguagem Programação**	**Linguagem**	**Disponibilização**	**Tarefas**
OpenNLP	Java	Várias	Domínio Público (Apache v2.0)	Identificação frases + *Chunker* + *Tokenização* + REM + Etiquetador Gramatical
DBpedia Spotlight	Java e Scala	Várias	Domínio Público (Apache v2.0)	REM + Desambiguação do Sentido das Palavras
jSpell	C e Perl	PT/EN	Domínio Público (Licença própria)	Análise Morfológica
LX-Suite	Perl	PT	Proprietária (Licença própria)[155]	Identificação frases + *Tokenizador* + Etiquetador Gramatical
MALLET	Java	Várias	Domínio Público (CPL v1.0)	Extração de tópicos + Classificação de documentos
Hunspell	C++	Várias	Domínio Público (MPL/GPL/LGPL)	Corretor ortográfico
Sylvester UGC Tokenizer	Python	PT	Domínio Público (AGPL v3.0)	*Tokenizador* vocacionado para texto proveniente de redes sociais

Tabela A.7: Análise comparativa de ferramentas de PLN.

[155] Disponível em *http://lxcenter.di.fc.ul.pt/tools/en/conteudo/LX-Tokenizer_License.pdf.*

A.6.2 Web Semântica

Nesta secção é apresentado o repositório semântico Sesame[156] usado para persistência e manipulação de triplos RDF, bem como o motor de inferência OWLIM[157].

A adoção destas ferramentas deveu-se essencialmente ao facto de serem já tecnologias usadas em outros projetos do Laboratório KIS (Knowledge and Intelligent Systems Lab), pertencente ao CISUC[158], pelo que existe um conhecimento mais aprofundado das mesmas na equipa e daí a sua consequente adoção também neste projeto.

Sesame

Trata-se de um *framework open-source* em Java para armazenamento, consulta e inferência de dados RDF e RDF Schema. Pode ser alojado como um servidor Web ou utilizado como uma biblioteca Java. Foi desenhada com o intuito de ser flexível podendo trabalhar sobre vários sistemas de armazenamento (base de dados relacionais, em memória, em sistema de ficheiros, etc.), possibilitando a leitura e escrita de RDF em diferentes formatos de serialização (RDF/XML, N-triples, Turtle[159], entre outros) e suportando duas linguagens de consulta: SPARQL e SeRQL (Broekstra & Kampman (2004)). Funciona através de uma API de acesso aos dados que suporta comunicação local e remota. O Sesame é disponibilizado mediante uma licença BSD 3-Clause.

OWLIM

O OWLIM é um repositório semântico (componente de software para persistência e manipulação de grandes quantidades de dados RDF) com suporte para a semântica RDFS e OWL. O OWLIM é fornecido na camada de armazenamento e inferência do Sesame designada por SAIL[160] (API de baixo nível cujo propósito é abstrair os detalhes de armazenamento e inferência, permitindo usar vários tipos). Esta *framework* não possui suporte para SWRL, usando por *default* OWL Horst (Urbani et al. (2010)) e possui três edições distintas, sendo que a versão OWLIM-Lite é a única que pode ser adquirida gratuitamente.

---

[156] Disponível em *http://www.openrdf.org/*.
[157] Disponível em *http://www.ontotext.com/owlim*.
[158] Disponível em *https://www.cisuc.uc.pt/*.
[159] Disponível em *http://www.w3.org/TeamSubmission/turtle/*.
[160] SAIL significa *Storage And Inference Layer*.

A.6.3 APIs da Web Social

Nesta secção, são apresentadas as bibliotecas consideradas para a extração de publicações, provenientes das redes sociais Twitter e Faceboook, que serão depois usadas para testar os módulos de PLN desenvolvidos.

Existe uma escolha considerável de ferramentas para extração de dados das redes sociais que, regra geral, apresentam bastantes semelhanças a nível de funcionalidades. No entanto, foram escolhidas as listadas de seguida, porque existe já alguma experiência no seu uso, são ambas consideravelmente populares e também porque são alvo de frequente atualização.

Twitter4j

O Twitter4j[161] trata-se de uma biblioteca Java *open-source* para integração com a Twitter API[162]. Permite, entre outras funcionalidades, acesso à lista de contactos subscritos (os denominados *followers*), acesso aos *tweets* presentes na *timeline* de um utilizador (desde que não seja privados), bem como efetuar pesquisas de *tweets* por termos, *hashtags*, etc. Possibilita também o uso da Streaming API[163] do Twitter, que permite a obtenção de *tweets* em tempo real.

RestFB

O RestFB[164] é uma biblioteca Java *open-source*, para integração com a rede social Facebook. Possibilita a obtenção das publicações pertencentes ao *feed* de notícias do utilizador, listagem de todos os amigos, pesquisas para obtenção de publicações de domínio público, bem como de utilizadores, eventos, grupos, locais, entre outros.

[161] Disponível em *http://twitter4j.org/en/index.html.*
[162] Disponível em *https://dev.twitter.com/.*
[163] Disponível em *https://dev.twitter.com/docs/streaming-apis.*
[164] Disponível em *http://restfb.com/.*

Anexo B
Descrição Detalhada dos Casos de Uso

Neste anexo são descritos de forma textual os vários casos de uso apresentados na Figura 3.1 do documento principal.

Caso de uso: CS01 - Efetuar pesquisa
Objetivos: Obtenção de resultados relevantes no contexto da *query* introduzida pelo utilizador. **Atores:** Utilizador. **Pré-Condições:** Sem pré-condições. **Sequência Típica de Eventos:** 1. O utilizador insere a *query* de pesquisa pretendida, podendo especificar uma ou mais palavras. 2. O sistema retorna o conjunto de resultados relativos à pesquisa realizada. **Pós-Condições:** O conteúdo relevante é apresentado ao utilizador, caso exista.

Tabela B.1: Caso de Uso CS01 – Efetuar pesquisa.

Caso de uso: CS02 - Visualizar informação extraída
Objetivos: Visualização da informação associada aos resultados da pesquisa efetuada. **Atores:** Utilizador. **Pré-Condições:** Uma pesquisa ter sido efetuada pelo utilizador. **Sequência Típica de Eventos:** 1. O utilizador seleciona uma publicação, das várias retornadas pela pesquisa. 2. O sistema retorna informação relevante extraída a partir da publicação selecionada (recorrendo à biblioteca). 3. O utilizador poderá visualizar os vários tipos de informação. **Pós-Condições:** Os vários tipos de informação associados a cada publicação são apresentados.

Tabela B.2: Caso de Uso CS02 – Visualizar informação extraída.

Caso de uso: CS03 – Recomendação de conteúdo relacionado
Objetivos: Visualização de sugestões de conteúdo relacionado com uma determinada publicação. **Atores:** Utilizador. **Pré-Condições:** Uma pesquisa ter sido efetuada pelo utilizador. **Sequência Típica de Eventos:** 1. O utilizador seleciona uma publicação, de forma a ter acesso a todo o seu conteúdo. 2. O sistema retorna um conjunto de sugestões de conteúdo relacionado com a publicação. 3. O utilizador poderá visualizar as publicações associadas a essas sugestões. **Pós-Condições:** Possibilidade do utilizador visualizar sugestões de conteúdo relacionado.

Tabela B.3: Caso de uso CS03 – Recomendação de conteúdo relacionado.

Caso de uso: CS04 - Permitir navegação no conteúdo
Objetivos: Possibilidade de navegação sobre as várias publicações, com o intuito de explorar os dados relevantes de forma mais rápida e fácil. **Atores:** Utilizador. **Pré-Condições:** Uma pesquisa ter sido efetuada pelo utilizador. **Sequência Típica de Eventos:** 1. O sistema retorna um conjunto de filtros para uma pesquisa efetuada. 2. O utilizador poderá selecionar um determinado filtro, de forma a visualizar apenas as publicações pretendidas. **Pós-Condições:** Possibilidade do utilizador navegar nos resultados da pesquisa.

Tabela B.4: Caso de uso CS04 – Permitir navegação no conteúdo.

Anexo C
Descrição Detalhada dos Requisitos Funcionais

Neste anexo são descritos de forma mais pormenorizada cada um dos requisitos funcionais apresentados na Tabela 3.1 do documento principal.

RF 1: Obtenção de conteúdo da Web 2.0
Deverá ser obtida informação proveniente de várias fontes da Web 2.0, posteriormente utilizada para desenvolvimento e teste dos vários módulos que serão criados no âmbito da tese.

RF 2: Limpeza e processamento dos dados recolhidos
Deverá ser realizado um pré-processamento de modo a efetuar uma limpeza inicial dos dados, corrigindo eventual conteúdo causador de ruído, como o caso das *tags* HTML existentes em dados provenientes de páginas Web. Deve também ser efetuada uma seleção, removendo dados duplicados ou incompletos de forma a aumentar a qualidade geral do conteúdo recolhido.

RF 3: Extração de informação a partir dos dados
Deverá ser possível extrair vários tipos de informação a partir dos dados recolhidos no RF 1. De forma a definir o alcance deste requisito, houve a necessidade de selecionar quais os dados mais relevantes e significativos, daí que tenha sido feita uma divisão num conjunto de requisitos mais específicos enunciados abaixo:

> **RF 3.1 Extração de informação acerca do autor**
> Deverá ser obtida informação acerca do autor da publicação de forma a ser possível efetuar a sua identificação.
>
> **RF 3.2 Extração de informação relativa ao conteúdo textual recolhido**
> Pretende-se obter informação referente ao conteúdo recolhido, nomeadamente o texto, a data da sua redação e, caso seja disponibilizada, a localização da publicação.

RF 3.3 Extração de comentários associados a cada publicação
Pretende-se também obter os dados referentes aos comentários associados a cada publicação.

RF 3.4 Extração de dados específicos da *media social*
Na *media social* existe também muita informação passível de ser extraída além do conteúdo textual. Desta informação fazem parte as *hashtags*, menções, URLs e eventuais *emoticons*, que caso estejam presentes no texto deverão também ser extraídos.

RF 3.5 Extração de termos e expressões multipalavra
Deve ser possível efetuar a extração de nomes ou padrões gramaticais (expressões multipalavra) que faça sentido serem mantidos juntos, como *Universidade de Coimbra*, *Engenharia Informática*, etc.

RF 3.6 Reconhecimento de entidades
Pretende-se efetuar a extração de entidades mencionadas nas publicações, possibilitando o reconhecimento de pessoas, produtos, organizações, etc.

RF 3.7 Extração de tópicos
Devem também ser extraídos tópicos, isto é, um conjunto de palavras frequentemente referenciadas no mesmo contexto, que permitam definir quais os conceitos associados a cada publicação.

RF 3.8 Extração de triplos
Deve também ser possível efetuar a extração de triplos na forma de sujeito, predicado e valor. O objetivo passa por reduzir a dimensionalidade dos textos e obter a informação mais revelante de forma sucinta.

RF 4: Representação da informação usando tecnologias da WS
A informação deverá ser representada usando tecnologias da Web Semântica, daí que deva ser possível efetuar a extração de triplos e também garantir a sua persistência. Assim são definidos os dois requisitos mais específicos apresentados abaixo:

RF 4.1 Geração de triplos RDF
Pretende-se efetuar a extração de triplos RDF na forma de recurso, propriedade e valor.

RF 4.2 Persistência da informação numa base de dados de triplos

A informação estruturada deverá ser persistida numa base de dados de triplos.

RF 5: Análise de sentimentos

Deverá ser possível identificar qual o sentimento nutrido relativamente a uma publicação, isto é, definir se a polaridade de um determinado texto é positiva, negativa ou neutra, tendo em conta as várias opiniões presentes no texto.

RF 6: Suporte de pesquisa semântica

Deverá existir a possibilidade de realizar pesquisa de natureza semântica sobre os dados, de forma a aperfeiçoar os resultados de pesquisa e mostrar ao utilizador os dados que lhe são mais relevantes e significativos. O utilizador poderá assim especificar uma palavra ou conjunto de palavras que deseje pesquisar, de forma a aceder a conteúdo relacionados com estas.

RF 7: Recomendação de conteúdo relacionado

Deverão ser disponibilizadas recomendações baseadas em conteúdo, isto é, para um determinado texto devem ser listadas publicações que contenham algum tipo de conteúdo relacionado que possa eventualmente despertar interesse ao utilizador.

RF 8: Suporte de navegação[165] sobre os dados

Deverá ainda ser disponibilizada a funcionalidade de navegação sobre a informação apresentada ao utilizador, através de um conjunto de filtros que permitam a exploração e acesso a dados relevantes de forma mais fácil e rápida.

[165] Em inglês é utilizado o termo *browsing*.

Anexo D

Descrição das Classes da Ontologia

Neste anexo são apresentadas as várias classes e propriedades respetivas referentes à ontologia visível na Figura 4.2 do documento principal.

D.1 Classe *Post*

A classe *Post* representa uma publicação genérica. Esta contém como *datatype properties* os seguintes elementos:

- *hasPostId* campo identificador da publicação;
- *hasPostContent* texto da publicação;
- *hasPostDate* data em que foi disponibilizada a publicação.

Contém ainda referências para as seguintes *object properties*:

- *hasPostAuthor;*
- *hasPostComment;*
- *hasPostEntities;*
- *hasPostMetadata;*
- *hasPostTopic;*
- *hasPostTerms;*
- *hasPostTriple.*

Da classe genérica *Post* fazem parte as subclasses *BlogPost*, *FacebookPost*, *OnlineNewsPost*, *TwitterPost* e *Comment*. A decisão de criar estas subclasses deveu-se ao facto de cada tipo de publicação conter atributos característicos, como, por exemplo, as publicações provenientes do Facebook possuírem o atributo número de *likes* ou os comentários possuírem uma pontuação associada (geralmente dada pelos utilizadores) que indica a sua relevância, entre outros.

D.2 Classe *Author*

A classe *Author* diz respeito ao autor da publicação, ou de forma mais genérica, à fonte de onde foi retirada a notícia, caso não seja possível obter o nome do escritor. Inclui como *datatype properties* os seguintes elementos:

- *hasAuthorId* campo identificador do autor;
- *hasAuthorName* nome do autor ou da fonte de informação;
- *hasAuthorDescription* breve descrição informativa acerca do autor.

D.3 Classe *Entities*

A classe *Entities* contém todo o conjunto de entidades extraídas a partir do processamento textual das publicações. Assim contém como *datatype properties* os seguintes elementos:

- *hasEntitiesId* campo identificador das entidades;
- *hasEvent* identifica as entidades da categoria Acontecimento;
- *hasLocal* diz respeito às entidades da categoria Localização;
- *hasOrg* representa as entidades do tipo Organização;
- *hasOther* apresenta as entidades pertencentes ao tipo Outro;
- *hasPerson* representa as entidades Pessoa;
- *hasTime* indica as entidades Tempo;
- *hasValue* campo referente à entidade Valor.

D.4 Classe *Metadata*

A classe *Metadata* contém informação específica de conteúdo online como URLs, *hashtags* e menções. As *datatype properties* consideradas foram:

- *hasMetadataId* campo identificador da classe Metadata;
- *hasHashtag* representa uma *hashtag;*
- *hasMention* representa uma menção;
- *hasURL* representa um URL.

D.5 Classe *Polarity*

A classe *Polarity* representa a polaridade (que no âmbito da tese pode ser classificada como positiva, negativa ou neutra) bem como a listagem de todas as palavras que detêm uma opinião (as denominadas *opinion words*) que contribuíram para esta atribuição. As *datatype properties* considerados para esta classe foram as seguintes:

- *hasPolarityId* campo identificador da polaridade;
- *hasOpinionWord* palavra detentora de opinião;
- *hasPolarity* polaridade final atribuída com base nas *opinion words* identificadas.

D.6 Classe *Terms*

A classe *Terms* apresenta todos os termos e expressões multipalavra extraídas a partir do processamento textual das publicações. Desta forma as *datatype properties* consideradas para esta classe foram:

- *hasTermsId* campo identificador dos termos;
- *hasAdjective* palavra cuja categoria gramatical é o adjetivo;
- *hasNoun* palavra cuja categoria gramatical é o nome comum;
- *hasProperNoun* palavra cuja categoria gramatical é o nome próprio;
- *hasVerb* palavra cuja categoria gramatical é o verbo;
- *hasMultiwordExpression* conjunto de palavras que representam uma expressão multipalavra.

D.7 Classe *Topic*

A classe *Topic* representa um tópico, ou seja, um conjunto de termos que ocorrem frequentemente juntos e que se encontram conceptualmente relacionados. Esta classe detém as seguintes *datatype properties*:

- *hasTopicId* campo identificador do tópico;
- *hasProb* probabilidade associada ao tópico.

D.8 Classe *Term*

A classe *Term* tem como objetivo representar um termo ou palavra associado a um tópico. Esta classe possui a seguinte *datatype property*:

- *hasTerm* trata-se de uma *string* que representa um termo ou palavra.

D.9 Classe *Triple*

A classe *Triple* representa um triplo, ou seja, uma frase representada através de sujeito, predicado e objeto. Esta classe detém as seguintes *datatype properties*:

- *hasTripleId* campo identificador do triplo;
- *hasPredicate* representa o predicado da frase;
- *hasSubject* identifica o sujeito da frase;
- *hasObject* apresenta o objeto da frase.

D.10 Classe *TermTopicOccurrence*

Esta classe tem como objetivo representar a relação existente entre um tópico e um termo, uma vez que cada tópico é constituído por um conjunto de termos e cada termo possui associado um peso ou relevância, para o tópico em questão. Pretende-se desta forma evitar a duplicação de dados decorrente do facto do mesmo termo poder estar contido em vários tópicos. A classe *TermTopicOccurrence* possui a seguinte *datatype property*:

- *hasWeight* que representa o "peso" do termo.

Contém ainda referências para as seguintes *object properties*:

- *hasTermOccurrence* que contém a referência do termo;
- *hasTopicOccurrence* que contém a referência do tópico associado ao termo.

D.11 Classe *PostTopicOccurrence*

Esta classe pretende representar a relação existente entre uma publicação e um tópico, sendo que cada publicação pode ter associado um ou mais tópicos e cada

tópico contém uma probabilidade representativa da sua relevância na publicação respetiva. A classe *PostTopicOccurrence* possui a seguinte *datatype property*:

- *hasProbability* que representa uma probabilidade associada à relevância do tópico para a publicação.

Contém ainda referências para as seguintes *object properties*:

- *hasPost* que contém a referência da publicação;
- *hasTopic* que contém a referência do tópico associado à publicação.