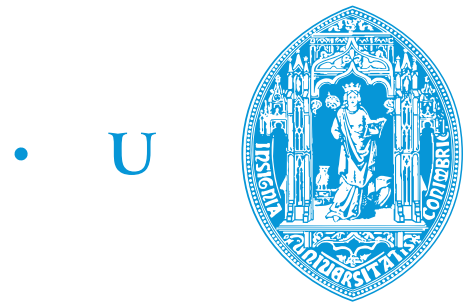

Tese de Mestrado
Engenharia Informática

Processamento de Linguagem Natural e Extração de Conhecimento

Sara Catarina Silva Pinto

Orientador na Wizdee
Doutor Bruno Emanuel Machado Antunes

Orientador no DEI
Doutor Hugo Gonçalo Oliveira

Departamento de Engenharia Informática
Faculdade de Ciências e Tecnologias
Universidade de Coimbra

Julho de 2015

Elementos do Júri:
Professor Doutor Hugo Gonçalo Oliveira
Professor Doutor Tiago Baptista
Professor Doutor Mário Rela

Resumo

A opinião de outras pessoas sempre foi um dado relevante no processo de tomada de decisão. Com o aparecimento da Internet, em especial das redes sociais, a quantidade de comentários de utilizadores sobre a qualidade de serviços e produtos aumentou exponencialmente. Sendo que esta informação começou a ter cada vez mais relevância para os utilizadores que antes de tomarem uma decisão sobre um serviço ou um produto procuram ter mais informação dos comentários e opiniões de outros. A influência que as opiniões das outras pessoas exercem tem feito aumentar o interesse nas ferramentas de análise de opiniões. Muitas vezes essas opiniões são publicadas em redes sociais em que o tipo de texto presente é geralmente não estruturado, apresentando diversos desafios de análise.

O presente trabalho propõe um conjunto de ferramentas capazes de extrair informação de texto que contenha opiniões, recorrendo a técnicas de Processamento de Linguagem Natural e abordagens de *Text Mining*.

Como tal, foi desenvolvida uma biblioteca com um conjunto de ferramentas necessárias para a análise de opiniões. O trabalho foca-se em texto extraído de redes sociais, que se caracteriza como sendo um texto não estruturado, menos cuidado, com abreviaturas, pitês e muitas vezes não respeita as regras ortográficas e sintáticas. Todas as ferramentas desenvolvidas permitem a análise de texto escrito na Língua Inglesa bem como na Língua Portuguesa. Para além do tipo de texto que se analisa, um dos principais desafios foi o desenvolvimento das ferramentas para a Língua Portuguesa, uma vez que existem relativamente menos recursos disponíveis, o que se refletiu nos resultados obtidos que foram sempre inferiores aos alcançados na Língua Inglesa.

Todas as ferramentas aqui desenvolvidas estão integradas com a plataforma *Wizdee* preparadas para serem usadas em produtos comerciais.

Palavras-Chave: Processamento de Linguagem Natural, Extração de Opiniões, Redes Sociais, Aprendizagem Automática, Extração de Informação, *Text Mining*

Abstract

The opinion of others has always been an important element in the process of making decisions. With the advent of the Internet, and in particularly, social networks, the amount of comments from users, regarding the quality of services and products, has increased exponentially. Following this, information began to have an increasing importance for users. Now, a user looks for more information before making a decision about a service or product, by using reviews and the opinions of others. The influence that the opinion of others exert, resulted in an increasing interest for tools capable of opinion mining. Often, we can find these opinions on social networks, where the challenge of unstructured text must be dealt.

The work presented in this thesis proposes a set of tools to extract information from subjective text, using Natural Language Processing techniques and Text Mining approaches.

As such, a library containing a set of tools for opinion mining was developed. The supported languages are English and Portuguese. As mentioned, the work focuses on text extracted from social networks, which is characterized as being unstructured text. Often it does not respect the syntactic rules of the language and contains spelling errors.

Furthermore, while there are challenges concerning the handling of unstructured text in both languages, one of the major challenges was the development of tools for the Portuguese language, since there are relatively fewer resources available. This was reflected in the results, where the Portuguese results were always lower than those achieved by the English tools.

All tools developed during this project are integrated with the platform Wizdee and are prepared for its use in commercial products.

Keywords: Natural Language Processing, Opinion Mining, Social Networks, Machine Learning, Information Extraction, Text Mining

Agradecimentos

Gostaria de começar por agradecer tanto ao Professor Doutor Paulo Gomes como ao Dr. Bruno Antunes, pelo acompanhamento, apoio incondicional e disponibilidade demonstrada ao longo de todo o ano. Os seus apoios e conselhos permitiram-me superar as dificuldades encontradas durante este trabalho e proporcionaram-me sempre uma motivação constante. Queria também agradecer ao Professor Doutor Hugo Oliveira, pelo conselhos e todo o acompanhamento e disponibilidade que ofereceu durante todo o projeto.

Aos meus pais e ao meu irmão, por todo o acompanhamento e dedicação que sempre me ofereçam, mesmo nos momentos mais difíceis. Por todos os valores e ensinamentos que sempre me transmitiram, acreditando sempre em mim.

Ao João agradeço toda ajuda e conselhos transmitidos ao longo do ano. Obrigado pela força e confiança que me transmites em todos os momentos.

Por fim, agradeço a todos os meus amigos pela paciência e compreensão.

Índice

Lista de Figuras

Lista de Tabelas

Acrónimos

PLN Processamento de Linguagem Natural

REM Reconhecimento de Entidades Mencionadas

ACP Análise dos Componentes Principais

LDA *Linear Discriminant Analysis*

CRF *Conditional Random Fields*

MVS Máquina de Vector de Suporte

DSP Desambiguação do Sentido das Palavras

GPU *Graphics Processing Unit*

API *Application Programming Interface*

WE *Word Embeddings*

RNC Redes Neuronais Convolucionais

RI Recolha de Informação

Capítulo 1

Introdução

As redes sociais, como o *Twitter* e o *Facebook*, têm apresentado um aumento de popularidade, o que originou uma grande quantidade de informação disponível em forma de texto (Aggarwal, 2011). Na última década as redes sociais têm sido alvo de estudos que pretendem analisar as relações entre as pessoas e extrair novas informações através de padrões nessas interações (Aggarwal, 2011). Naturalmente, a extração automática de informação a partir de texto é um desafio, principalmente quando se refere a texto proveniente das redes sociais, pois este caracteriza-se como sendo um texto pouco estruturado, muitas vezes com erros ortográficos, com diversas abreviações e com a introdução de novos tipos de *tokens*, como os *smiles*, *hashtags*, entre outros (Hu and Liu, 2012). Em algumas redes sociais, como o *Twitter*, é imposto um limite ao tamanho do texto a publicar, o que introduz outros desafios, uma vez que se pode perder a contextualização dessa mensagem.

A área de Processamento de Linguagem Natural (PLN) (Jurafsky and Martin, 2008) ocupa-se do estudo da linguagem dos seres humanos, com o objetivo de construir ferramentas que permitam compreender, analisar e gerar linguagem natural. Ao longo dos anos, um conjunto de tarefas foram criadas e desenvolvidas, como por exemplo a extração de classes gramáticas, identificação de frases, desambiguação de palavras, entre outras. Uma outra área que também se ocupa da análise de texto, é a Extração de Conhecimento em Texto ou *Text Mining* (Aggarwal and Zhai, 2012a), que pretende extrair informação útil do texto, tendo como algumas aplicações a classificação de texto, extração de tópicos, reconhecimento de entidades mencionadas, entre outras (Aggarwal and Zhai, 2012a). Estas duas áreas, são trabalhadas muitas vezes em conjunto, sendo que *Text Mining* utiliza muitas análises introduzidas pelo PLN (Aggarwal and Zhai, 2012a).

O objetivo deste trabalho foi a criação de ferramentas de *Text Mining* para o processamento de texto proveniente de redes sociais. Mais especificamente, foi desenvolvido um conjunto de ferramentas, que extraem informação de texto que contém opiniões. A análise de opiniões tem vindo a tornar-se importante, pois saber o que outras pessoas pensam sempre foi um tipo de informação muito requisitado no processo de tomadas de decisão (Pang and Lee, 2008). Ainda antes da Internet, recorria-se à opinião de amigos e familiares para recomendar determinados produtos, explicar qual era o partido em que pretendia votar, etc. A Internet trouxe a possibilidade de captar essas opiniões de uma quantidade de pessoas muito maior, que podem ser experientes no assunto, de diversas culturas e contextos (Pang and Lee, 2008). Estudos avançam que muitas pessoas recorrem a opiniões publicadas na Internet para o seu processo de decisão. Cerca de 81% dos utilizadores de Internet já fizeram, pelo menos uma vez, pesquisa online sobre um determinado produto, e 20% dos utilizadores fazem-no no seu dia a dia (ComScore and Group, 2007) (Horrigan, 2008). O interesse manifestado pelos utilizadores nas opiniões online sobre produtos e serviços, e a influência que essas opiniões exercem, é algo que as empresas que vendem esses

produtos e serviços têm prestado cada vez mais atenção (Hoffman, 2008). O Texto que expressa uma opinião/sentimento é caracterizado como texto subjetivo. As opiniões estão geralmente associadas a uma entidade (por exemplo um produto ou uma empresa), a um aspeto (característica especifica da entidade) e possuem uma polaridade (se a opinião é positiva, negativa ou neutra) (Liu and Zhang, 2012).

É este tipo de problema que este trabalho se propôs a resolver. Mais especificamente, para se atingir os objetivos deste trabalho este trabalho, foi necessário tratar dos seguintes pontos:

- Obtenção de textos de várias redes sociais, como o *Facebook* e o *Twitter*;
- A extração da polaridade de texto subjetivo, classificando-o em positivo, negativo ou neutro;
- A extração das entidades a que uma opinião se refere;
- A extração do aspeto específico da entidade a que a opinião se refere;
- A extração da polaridade associada ao aspeto ou entidade.

Todas as ferramentas desenvolvidas foram integradas no produto *Wizdee*, e suportam duas línguas diferentes: o Português e o Inglês. O trabalho proposto é composto por uma forte componente de investigação, que começou pelo estudo de diferentes abordagens usadas em diversos trabalhos, como por exemplo o uso de abordagens mais clássicas da aprendizagem computacional, como as Máquinas de Vetor de Suporte ou abordagens que estão a ser utilizadas em trabalhos mais recentes, como o uso de algoritmos de *Deep Learning*. As abordagens exploradas para o desenvolvimento das ferramentas podem-se, essencialmente, dividir em duas categorias: técnicas de aprendizagem automática e abordagens linguísticas. A ferramenta de extração de polaridade, foi desenvolvida usando Máquinas de Vetor de Suporte, um algoritmo de Aprendizagem Automática. Já o resto das ferramentas foram desenvolvidas usando técnicas de Processamento de Linguagem Natural e *Text Mining*, optando assim por uma abordagem mais linguística.

Resumidamente, as contribuições deste trabalho são as seguintes:

- Estado da arte no domínio do Processamento de Linguagem Natural, *Text Mining* e Análise de Opiniões.
- Biblioteca desenvolvida para a extração da polaridade de uma opinião tanto para Inglês como para Português, com resultados alcançados no valor de 63% e 59% de Medida-F, respetivamente.
- Biblioteca que permite a extração de informações relevantes sobre uma opinião, como as entidades e os aspetos, tanto para Inglês como para Português, com resultados entre os 65-67% e 57% de Medida-F, respetivamente.
- Conjunto de dicionários relevantes para cada ferramenta desenvolvida, criados para ambas as línguas suportadas.
- Experimentação realizada nas diferentes ferramentas, e o desenvolvimento de diferentes corpora anotados manualmente e desenvolvidos para cada ferramenta.
- Integração de todas as ferramentas com a plataforma Wizdee.

Por fim, quanto à estrutura do presente documento, este está dividido em oito capítulos que serão descritos de seguida:

Capítulo 2 - Conhecimento Prévio e Estado da Arte: Neste capítulo descrevem-se os principais conceitos relevantes para o presente trabalho. Começa-se pela introdução da área de Processamento de Linguagem Natural, que é a base do processamento de qualquer texto. De seguida, é feita uma introdução ao *Text Mining* e as suas aplicações. Ainda nessa secção, é apresentado, um conjunto de algoritmos de aprendizagem automática aplicados em texto. Posteriormente, é introduzido o tópico de extração de opiniões, onde é feita uma análise do tipo de texto alvo e é apresentado o processo de criação de informação estruturada e relevante a partir de texto que contém opiniões. Ainda nesta secção, é feito um levantamento de alguns trabalhos realizados na área de extração de opiniões. Por fim, é feita uma introdução das redes sociais, seguido da apresentação de alguns recursos e ferramentas relevantes para o desenvolvimento do trabalho proposto.

Capítulo 3 - Análise de Competidores: Neste capítulo, são apresentados alguns produtos já existente no mercado que fornecem o mesmo tipo de análises que este trabalho se propõe a desenvolver. É feita uma análise a cada uma delas, comparando-as no fim do capítulo.

Capítulo 4 - Abordagem: Este capítulo começa por uma análise aos requisitos do trabalho, de seguida são apresentados os riscos envolvidos no seu desenvolvimento e os testes que serão feitos. É também feita uma apresentação da arquitetura do sistema, onde as ferramentas desenvolvidas serão integradas, e os componentes que serão desenvolvidos.

Capítulo 5 - Metodologia e Planeamento: É descrita a metodologia adotada durante o desenvolvimento deste trabalho. É apresentado o planeamento seguido e a distribuição do trabalho feita durante do desenvolvimento do mesmo.

Capítulo 6 - Implementação: Neste capítulo são apresentados as abordagens e detalhes da implementação das diferentes ferramentas desenvolvidas, bem como a descrição de cada dicionário desenvolvido.

Capítulo 7 - Testes: Neste capítulo são apresentados todos os testes desenvolvidos para cada ferramenta. Para cada uma das ferramentas, é feita uma descrição dos testes realizados e uma apresentação do corpus de teste usado. De seguida, são apresentados os resultados obtidos em cada teste, assim como, a análise de cada um deles.

Capítulo 8 - Conclusão: Por fim, neste capítulo, é apresentado um resumo do trabalho proposto, assim como as contribuições resultantes. Ainda neste capítulo são sugeridos alguns melhoramentos possíveis a realizar futuramente.

Capítulo 2

Conhecimento Prévio e Estado da Arte

Neste capítulo são apresentados os conceitos e estudos mais relevantes para a realização e compreensão do trabalho proposto. Na secção 2.1 é introduzido o conceito de Processamento de Linguagem Natural, os diferentes Níveis de Conhecimento e as principais tarefas associadas. Na secção 2.2, são apresentados os objetivos e principais processos de *Text Mining*. Esta secção apresenta ainda a descrição das principais aplicações de *Text Mining* e os diferentes algoritmos de aprendizagem automática geralmente aplicados a texto. Posteriormente, na secção 2.3, sobre a extração de opiniões, é feita uma análise da sua representação e os passos principais, como a extração da polaridade e a extração de aspetos e entidades. Nestes últimos, é feito um estudo onde se expõe diversos trabalhos que tentam resolver este problema. O conceito de redes sociais e o seu efeito nas tarefas de PLN são introduzidos na secção 2.4. Por fim, na secção 2.5 e 2.6 são apresentados, de forma sucinta, alguns recursos e ferramentas disponíveis, tanto para a Língua Portuguesa como para a Língua Inglesa.

2.1 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) é um campo da Inteligência Artificial (Jurafsky and Martin, 2008) cujo o objetivo é compreender, analisar e gerar a língua natural para os Humanos, de forma a que, eventualmente, seja possível nos dirigirmos a um computador da mesma forma que nos dirigimos a uma pessoa. Naturalmente, nos últimos anos, esta área tornou-se bastante popular, devido ao acesso massivo e facilitado a informação através da Internet, tornando-se, para além de cativante, imprescindível (Jackson and Moulinier, 2007).

Uma das grandes dificuldades no PLN é o facto da linguagem natural possuir uma característica que a torna difícil de interpretar: a ambiguidade (Jackson and Moulinier, 2007). A ambiguidade existe quando não é possível atribuir apenas um significado a uma determinada expressão. Normalmente, uma pessoa consegue facilmente interpretar corretamente o que lhe está a ser comunicado devido à sua experiência, às regras sintáticas, ao contexto ou à cultura. No entanto um computador não tem a capacidade que lhe permite ter em conta todos esses pontos. A ambiguidade pode ocorrer nos diferentes níveis de conhecimento: fonético, morfológico, sintático, semântico, pragmático e de discurso. De seguida, são apresentados com mais detalhe os diferentes níveis de conhecimento e as suas principais aplicações.

2.1.1 Níveis de Conhecimento

Para que um sistema de PLN seja robusto necessita de, pelo menos, contemplar algumas tarefas base. Cada uma dessas tarefas concentra-se em resolver parte de um problema maior. Por exemplo, a análise fonética pretende analisar as palavras tendo em conta a maneira como estas são pronunciadas. A análise morfológica estuda as palavras de forma isolada, classificando e extraindo a classe gramatical, o lema e o radical. Já a análise sintática, faz o estudo das palavras tendo em conta a relação entre elas numa frase. A análise semântica pretende dar significado às palavras de forma a que seja possível perceber o que realmente o texto pretende transmitir. Por fim, a análise pragmática e de discurso preocupa-se essencialmente com o contexto do texto. De seguida, são descritas com mais detalhe as análises base necessárias num processador de linguagem natural.

Análise Fonética

A análise fonética consiste no estudo dos sons de uma língua. A fonética divide o discurso em fonemas que podem ser, essencialmente, vogais ou consoantes. Neste nível, a ambiguidade está presente através das palavras homófonas, ou seja palavras cujas pronúncias são semelhantes. Embora o som seja semelhante, as palavras possuem significados e representações diferentes. É fácil encontrar este tipo de palavras no nosso vocabulário, como por exemplo, *cozer* e *coser* ou *houve* e *ouve*. Este tipo de análise não será muito aprofundada, uma vez que esta incide sobre o discurso e neste trabalho pretendemos lidar exclusivamente com texto escrito.

Análise Morfológica

Esta tarefa concentra-se no estudo e classificação de palavras isoladas. Separa o texto em átomos ou *tokens*, fazendo um estudo a cada um desses átomos isoladamente. Os átomos podem ser palavras, sinais de pontuação, dígitos entre outros. Aos átomos que formam palavras é feita a identificação da sua classe gramatical, o seu lema e o seu radical.

A classe gramatical identifica qual é a função dessa palavra, ou seja, se é um verbo, nome, pronome, etc. Para além da classe gramatical também identifica outras características como o género (masculino ou feminino), o número (singular ou plural) e o grau (1^o pessoa, 2^o pessoa ou 3^o pessoa). Aos verbos acrescenta-se ainda o tempo (por exemplo, pretérito perfeito) e o modo (por exemplo, modo indicativo).

Quanto à ambiguidade, esta acontece quando a mesma palavra pode pertencer a diferentes classes gramaticais. Um exemplo disso é a palavra "caminho", que tanto pode ser um verbo, tal como na frase "*Eu caminho muito pouco.*", como um substantivo, como na frase "*O melhor caminho é pela autoestrada.*".

Análise Sintática

A análise sintática, ou *parsing*, pretende estudar a relação entre as palavras na frase. Permite, por exemplo, reconhecer que um determinado adjetivo está a classificar um determinado nome numa frase. Saber que uma determinada palavra pertence a uma classe gramatical ajuda a determinar que tipo de palavras são as mais prováveis de serem suas vizinhas (Jurafsky and Martin, 2008). Por exemplo, na Língua Portuguesa, um pronome pessoal é geralmente seguido de um verbo.

Quanto à ambiguidade, é fácil perceber que esta tarefa consegue resolver a ambiguidade existente na análise morfológica descrita anteriormente. Por exemplo, na frase "*Eu canto todas as noites.*" a palavra "*canto*" tanto poderia ser um verbo como um substantivo, no entanto a análise sintática classifica-a como um verbo, pois o seu precedente é um

pronome pessoal. Como se pode perceber esta tarefa tem uma importância significante, uma vez que é fulcral tanto para tarefas de Recolha de Informação (RI), como para a tarefa de Desambiguação do Sentido das Palavras (DSP). Apesar de resolver alguns níveis de ambiguidade, esta continua presente. Por exemplo, na frase "*Eu ontem vi-te no carro.*" não é possível determinar com certeza quem é que estava no carro, se era *eu* ou a pessoa que eu vi.

A análise sintática permite construir uma árvore de constituência[1] e uma árvore de dependência[2]. Na árvore de constituência, exemplificada na Figura 2.1, a representação de uma frase passa por explicitar as relações entre os constituintes dela, como por exemplo o sintagma adverbial ou sintagma nominal. A árvore de dependência representa unicamente a relação entre as palavras, como é exemplificado na Figura 2.2 [3].

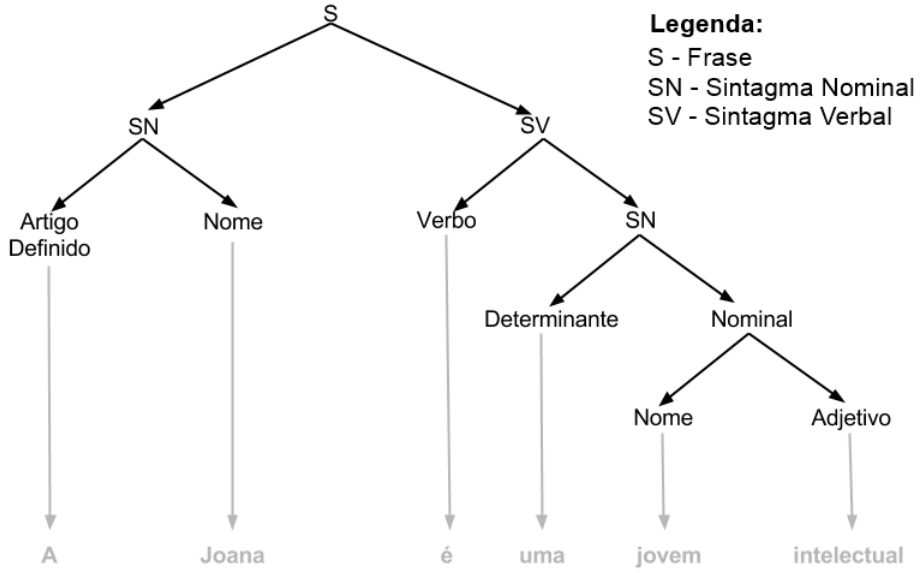

Figura 2.1: Árvore de Constituência para a frase "*A Joana é uma jovem intelectual.*"

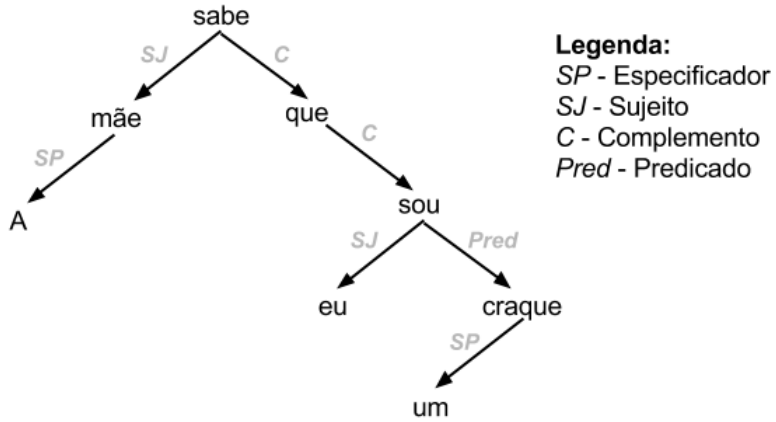

Figura 2.2: Árvore de Dependências para a frase "*A mãe sabe que eu sou um craque.*"

[1] Em inglês, *Constituency-based Parse Tree.*
[2] Em inglês, *Dependency-based Parse Tree.*
[3] Anotação do CINTIL-Treebank (Branco et al., 2011)

Análise Semântica

Todas as análises anteriores são importantes para a interpretação e classificação das frases, no entanto o significado do texto ainda não foi abordado. Conseguir perceber a frase "*Eu vou fazer uma viagem pela Europa.*" não é uma tarefa simples, uma vez que é necessária a tradução da linguagem natural para uma linguagem formal, sem ambiguidade, de forma a que as máquinas consigam interpretar o sentido das palavras.

A análise semântica concentra-se nesse objetivo, ou seja, pretende clarificar o significado das palavras num texto. Após esta análise obtém-se um texto em linguagem formal, passível de ser compreendido por um computador, ou seja, um texto sem ambiguidade, pois só assim é possível representá-lo em linguagem formal.

Ao longo dos anos foram criadas diversas representações formais como: lógica de predicados (Smullyan, 1995), grafos direcionais e *frames* semânticas (Fillmore, 1982). Na Figura 2.3, na Figura 2.4 e na Figura 2.5, são exemplificadas as diferentes representações.

é(biblioteca, local)

tem(biblioteca, livros)

Figura 2.3: Representação do significado da frase "*A biblioteca é um local com livros.*" usando lógica de predicados.

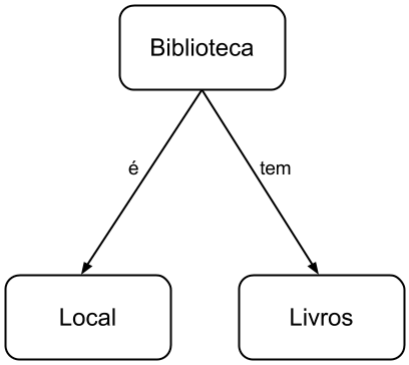

Figura 2.4: Representação do significado da frase "*A biblioteca é um local com livros.*" usando grafos direcionais.

Biblioteca:
é: local
tem: livros

Figura 2.5: Representação do significado da frase "*A biblioteca é um local com livros.*" usando *frames* semânticas.

Análise Pragmática e de Discurso

A análise pragmática consiste em determinar a relação entre a linguagem e o contexto. O contexto inclui perceber como a língua se refere a pessoas e objetos, como o discurso está estruturado e como este é interpretado pelo ouvinte (Jurafsky and Martin, 2008). A pragmática defende que o significado do texto também depende do contexto e do conhecimento prévio entre as partes envolvidas.

Esta análise permite resolver algumas situações de ambiguidade sintática, no entanto, num contexto em que os intervenientes não partilham o mesmo contexto, a ambiguidade poderá continuar presente. Um exemplo é a frase "*Hoje está fresco!*" que só é totalmente interpretada se ambos os agentes conhecerem o conceito de "*estar fresco*" do emissor. Para um esquimó estar fresco pode querer dizer que estão -15°C, no entanto para um português pode apenas querer dizer que estão 0°C.

Neste tipo de análise é necessário compreender que a linguagem não consiste em palavras/frases isoladas mas em frases relacionadas e agrupadas. A esse grupo de frases chamamos-lhe discurso que segundo a pragmática, a sua interpretação é influenciada pelo contexto (Jurafsky and Martin, 2008).

A análise a nível do discurso estuda as relações entre as frases, identificando relações entre unidades maiores que uma frase, de forma a compreender o contexto necessário à interpretação da frase. Algumas das suas funções passam pela identificação de anáforas (expressões que se referem a outras na mesma frase ou texto), a identificação de figuras de estilo, a perceção de coerência no discurso, entre outras (Jurafsky and Martin, 2008).

2.1.2 Tarefas

Os níveis de análise anteriormente descritos dividem-se muitas vezes em tarefas mais pequenas, como por exemplo o isolamento dos átomos. Em seguida, são descritas algumas dessas tarefas base.

Identificação de Frases e Átomos[4]

Esta tarefa é uma das mais simples, cujo objetivo é dividir o texto em elementos atómicos. Esses elementos podem ser palavras, números, pontuação, entre outros. Apesar de simples, ainda apresenta alguns desafios. Assunções básicas, como assumir que a pontuação assinala o fim de uma frase são revistas. Um exemplo é "*O Sr. Augusto vai hoje ao médico!*", em que o primeiro sinal de pontuação não representa o fim da frase. As redes sociais também vieram dificultar esta tarefa, pois foram introduzidos novos tipos de átomos, como por exemplo os *smiles*[5] e as *hashtags*[6]. Por exemplo, a frase:

"O Alfredo não está a trabalhar."

é transformada no seguinte conjunto de átomos:

[O] [Alfredo] [não] [está] [a] [trabalhar] [.]

Identificação da Classe Gramatical[7]

Esta tarefa pretende identificar a função de cada palavra na frase. Perceber qual é a classe gramatical de uma palavra pode diminuir a ambiguidade a nível semântico. Cada átomo é representado por uma etiqueta que define a sua função, tal como se pode ver no exemplo em baixo. Um dos conjuntos de etiquetas mais usados é o *Penn Treebank TagSet*[8] .

A frase:

"Eu hoje vou às compras."

[4]Em inglês, *Sentence Splitting* e *Tokenization*.

[5]Por exemplo, :) ou :D.

[6]Uma *hashtag* é representada pelo símbolo # seguido de uma palavra, por exemplo #viagem ou #tenhofome.

[7]Em inglês, *Part-of-Speech Tagging* ou *POS Tagging*.

[8]Ver conjunto de etiquetas em `https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`

é anotada da seguinte forma[9]:

[Eu/PRP] [hoje/RB] [vou/VB] [às/DT] [compras/NN]

Legenda:

PRP Pronome Pessoal
RB Advérbio
VB Verbo
DT Artigo Definido
NN Substantivo

Lematização e *Stemming*

A Lematização consiste na identificação do lema de uma palavra, ou seja a sua forma canónica. No caso dos verbos, geralmente o lema é representado pelo verbo no seu infinitivo. No caso dos substantivos e adjetivos são, geralmente, representados pela palavra no modo masculino e singular. Por exemplo, o lema de "*gostei*" é "*gostar*", o lema de "*boas*" é "*bom*" e o lema de "*mesas*" é "*mesa*".

Já o *Stemming* converte a palavra no seu radical, ou seja, na sua forma mínima. Por exemplo, o radical de "*gostar*" é "*gost*", o radical de "*esperto*" e "*espertinho*" é "*espert*" e o radical de "encantar" é "*encant*".

Desambiguação do sentido das palavras

Desambiguação do Sentido das Palavras (DSP) é uma tarefa que pode ser considerada um problema de classificação, onde se pretende desambiguar palavras que podem assumir vários sentidos (Yarowsky, 2010). Um exemplo básico, é a desambiguação tendo em conta as classes gramaticais, como se pode ver na Figura 2.6, em que a palavra "*baixo*" pode ser desambiguada sabendo apenas que é um adjetivo. No entanto, no mesmo exemplo, à palavra "banco" deve se aplicar outro método de desambiguação, pois a classe gramatical não é suficiente para distinguir os diferentes conceitos que esta pode assumir. A partir de uma palavra e uma lista de possíveis significados, a DSP escolhe o significado mais apropriado tendo em conta o contexto. O contexto tem em conta as palavras mais próximas da palavra em questão dando-lhes uma maior importância comparado com as palavras mais afastadas (Yarowsky, 2010).

Resolução de Co-referências

Nesta tarefa pretende-se identificar expressões que se referem a entidades no mesmo texto. Por exemplo, na frase "*O Rui mudou de casa. Ele agora vive no centro.*" pretende-se perceber que a expressão "*Ele*" se refere à entidade "*R*ui".

2.2 *Text Mining*

O aparecimento das redes sociais introduziu um crescimento significativo da quantidade de informação em texto existente, o que leva a que cada vez mais sejam procuradas formas de aglomerar essa informação de uma forma mais estruturada, com o intuito de a tornar útil. *Text Mining* assemelha-se muito ao *Data Mining* ou Extração de Conhecimento de

[9] Anotações de acordo com o sistema disponibilizado pelo LX-Center em `http://lxcenter.di.fc.ul.pt/`

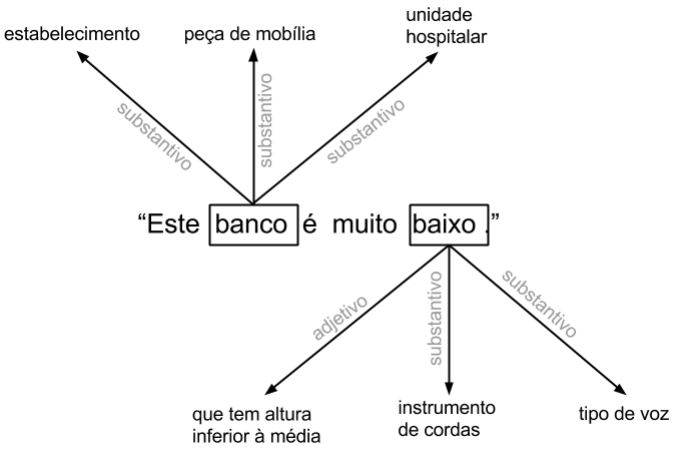

Figura 2.6: Exemplo de desambiguação do sentido da palavra "*banco*" e "*baixo*".

Base de Dados, que pretende extrair informação útil através de um conjunto de dados, no entanto o primeiro concentra-se em dados textuais, ou seja aplica-se a informação não estruturada, ao contrário do último que se direciona para dados estruturados (Feldman, 2006). Naturalmente, esta área oferece grandes vantagens a nível comercial, uma vez que a maior percentagem de informação produzida, hoje em dia, é na forma de texto (Tan, 1999).

A interpretação do texto é um grande desafio, uma vez que quando escrevemos, o nosso intuito é este ser lido por uma pessoa e não por uma máquina. Ou seja, quando lemos um texto temos algum conhecimento extra que nos permite interpretar mais facilmente a informação transmitida, e por isso somos melhores a identificar ironia do que uma máquina.

O texto possui características que tornam a sua análise mais difícil. O texto é esparso e de grande dimensionalidade, ou seja, uma língua é capaz de conter mais de 1 milhão palavras diferentes mas apenas uma percentagem relativamente baixa é usada frequentemente (Aggarwal and Zhai, 2012a). Assim, se escolhermos representar texto usando o modelo Saco de Palavras ou *Bag-of-Words*[10] teremos um vetor com milhares de entradas em que maior parte delas estão a zero. O *Text Mining* pode-se dividir nos seguintes passos:

- **Extração de Características**[11]: É feita a transformação de texto para um conjunto de características[12] que vão representar o texto. Conseguir representar toda a informação do texto num vetor é uma das tarefas mais complexas e também das mais importantes. Geralmente esta transformação é feita usando técnicas de Processamento de Linguagem Natural faladas em 2.1.2. Nesta fase, obtém-se um vetor de representação do texto, tal como exemplificado na Figura 2.7.

- **Criação do Modelo**[13]: Usando as características extraídas no ponto anterior, aplicam-se algoritmos de aprendizagem automática[14], que criam um modelo de acordo com o objetivo do sistema.

Na Figura 2.8 é apresentado o processo de extração de conhecimento em texto numa forma simplificada.

[10]Representa o documento num vetor com valores 0 ou 1 que determina a presença ou ausência de determinados termos no documento. Esses termos são palavras que aparecem, pelo menos uma vez, numa coleção de documentos.
[11]Em inglês, *Feature Extraction*.
[12]Em inglês, *features*.
[13]Em inglês, *Feature Extraction*.
[14]Em inglês, *Machine Learning*.

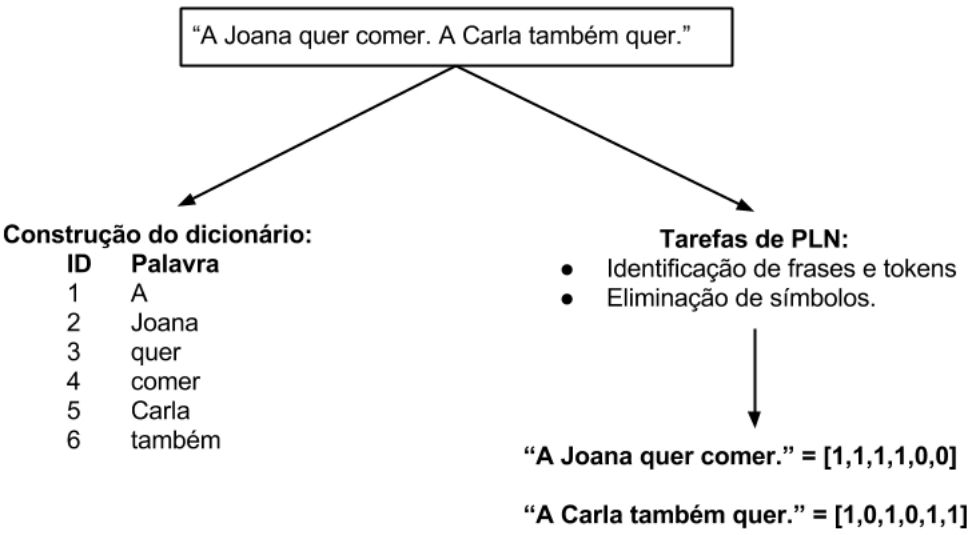

Figura 2.7: Exemplo simplificado de extração de características de um texto.

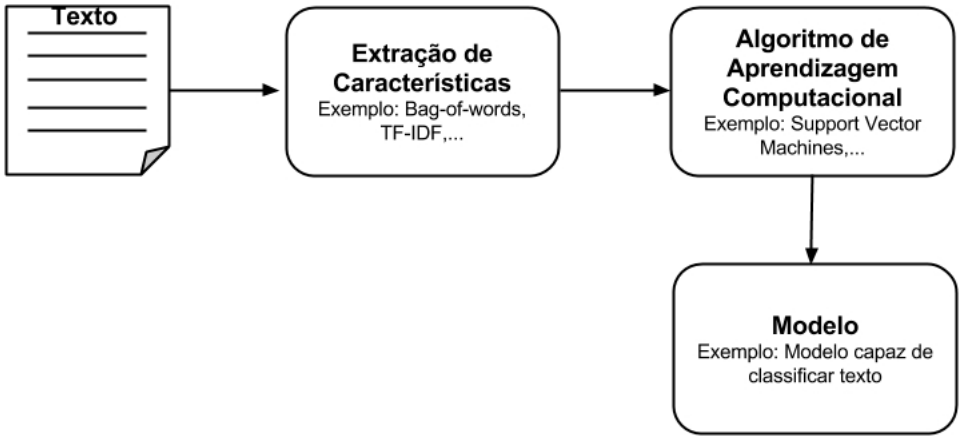

Figura 2.8: Processo simplificado usado para análise de texto.

2.2.1 Aplicações

A área de *Text Mining* tem diversas aplicações, sendo que muitas delas usam como base tarefas de Processamento de Linguagem Natural (PLN). Algumas das aplicações mais populares são descritas em baixo.

Classificação de Texto

A Classificação de Texto (Feldman, 2006) permite a partir de um conjunto pré-definido de classes (por exemplo, email "*Importante*" e "*Não Importante*"), classificar o texto de acordo com essas classes. Geralmente, necessita de uma coleção de documentos já classificados de forma a construir um modelo e sempre que tiver um novo documento este é classificado com o modelo criado, atribuindo-lhe assim pelo menos uma classe. Classificar documentos permite filtrar ou até organizar hierarquicamente documentos.

Uma tarefa relevante para este trabalho é a classificação de polaridade. A classificação de polaridade pretende, através de um texto, extrair a polaridade da opinião expressa relativamente a uma entidade (por exemplo, um produto) ou um aspeto (por exemplo, a bateria de um produto). O objetivo é conseguir classificar o texto tendo em conta a opinião

do autor, por exemplo, opinião positiva, negativa ou neutra. Esta tarefa está descrita com mais detalhe na secção 2.3.2.

Agrupamento de Texto

O Agrupamento de Texto[15] (Weiss et al., 2005) tem como objectivo agrupar textos que sejam semelhantes entre si. A maior diferença entre a classificação e o agrupamento de texto é que na primeira é necessário construir previamente um conjunto de tópicos pretendidos enquanto que no agrupamento isso não é necessário. Por exemplo, uma empresa que recebe milhares de reclamações quer perceber que tipo/categorias de reclamações tem de uma forma automática, não querendo restringir à partida as categorias.

O Agrupamento de Texto, usa informação de semelhança entre os textos para os agrupar, ou seja cada texto pertence a um grupo ou *cluster* que contém textos semelhantes entre si. *Soft Clustering* (Kobayashi and Aono, 2007), tem um processo semelhante, no entanto permite que cada documento esteja associado a vários grupos, ou seja calcula o grau de semelhança que o documento tem com diversos grupos.

Extração de Tópicos[16]

O processo de Agrupamento de Texto pode ser visto como uma representação de pouca dimensionalidade do documento, no entanto é muito difícil caracterizar cada grupo ou *cluster* de uma forma percetível.

Extração de Tópicos (Crain et al., 2012) tenta de certa forma integrar *Soft Clustering* com redução de dimensionalidade. Cada tópico representa um conjunto de palavras que são frequentemente encontradas no mesmo contexto, e cada palavra tem associada uma probabilidade de pertença a um tópico (Aggarwal and Zhai, 2012c). A Figura 2.9 começa por representar dois tópicos diferentes, com diversas palavras associadas a cada um deles e as suas probabilidades. Por exemplo, no primeiro tópico a palavra "*futebol*" tem uma associação mais forte do que a palavra "*taça*". Ainda na mesma figura, é representado um documento que também está associado aos dois tópicos, no entanto com uma probabilidade maior para o primeiro, cerca de 75%.

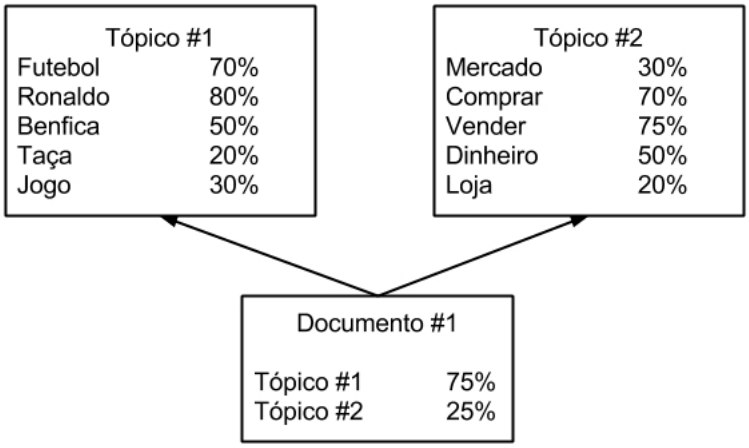

Figura 2.9: Exemplo de Extração de Tópicos.

[15]Em inglês, *Text Clustering*
[16]Em inglês, *Topic Modeling*.

Extração de Informação

O objetivo da Extração de Informação (Jiang, 2012) é representar, de uma forma automática, informação não estruturada de uma forma estruturada, que pode depois ser usada nos motores de pesquisa. Extração de Informação é composta por diversas tarefas, como Reconhecimento de Entidades Mencionadas e Extração de Relações.

- **Reconhecimento de Entidades Mencionadas:** O Reconhecimento de Entidades Mencionadas (REM) tem como objetivo reconhecer e classificar entidades do mundo real num texto, como por exemplo, nome de pessoas, organizações, localizações, entre outras (Chinchor, 1997). O REM pode inclusive ajudar na tarefa de Identificação de Classes Gramaticais, identificando desde logo algumas entidades o que pode ajudar na classificação dos átomos vizinhos. O uso de listas pré-concebidas de entidades conhecidas para comparação de átomos, muitas vezes não é suficiente para as detetar todas, uma vez que o conjunto de entidades possíveis vai sendo alargado (Jiang, 2012). Um exemplo de extração de entidades observa-se na frase: "A Carolina é de Aveiro." em que é possível reconhecer duas entidades tal como se pode verificar na Figura 2.10.

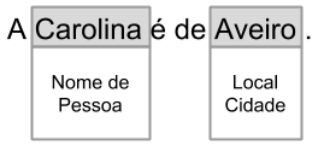

Figura 2.10: Exemplo de REM.

- **Extração de Relações:** Identificar relações entre entidades, como "casado com" ou "fundador de". Por exemplo, na frase "A Maria comprou um carro", é possível extrair uma relação, como se pode ver na Figura 2.11.

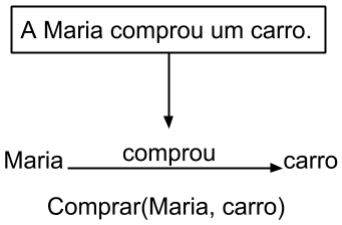

Figura 2.11: Exemplo de Extração de Relações.

2.2.2 Algoritmos

De forma a extrair informação relevante de um texto, começa-se por transformar esse texto num vetor de características de modo a que lhe seja aplicado um algoritmo que consiga aprender e extrair a informação importante. O tipo de algoritmo usado é dependente da finalidade do sistema, por exemplo se o objetivo for classificar o texto entre "*spam*" e "*não spam*", devemos usar um algoritmo de classificação, como uma Máquina de Vetores de Suporte[17] (Feldman, 2006). Basicamente, como entrada no sistema, é necessário uma matriz ou corpus de instâncias. Cada instância é um caso independente representado por um vetor, por exemplo num texto uma instância pode representar uma frase. Ao conjunto

[17] Em inglês, *Support Vector Machines*.

de instâncias dá-se o nome de *dataset* ou corpus que é representado por uma matriz. Cada instância tem um número fixo de atributos, geralmente numéricos, que são características representativas dessa instância. Na Figura 2.12 pode se observar um exemplo de corpus. Por fim esse corpus, serve como entrada a um determinado algoritmo que tenta aprender um ou mais conceitos.

	Atributo #1	Atributo #2	...	Atributo #M
Instância #1	3	32.2	...	501
Instância #2	7	56.5	...	658
Instância #3	1	74.9	...	978
	⋮			⋮
Instância #N	2	54.3	...	546

Figura 2.12: Demonstração de um corpus e as suas instâncias.

O modo de aprendizagem pode ser dividido em três grandes grupos: Aprendizagem supervisionada[18], não-supervisionada e semi-supervisionada. Aprendizagem supervisionada (Witten et al., 2011) é uma aprendizagem feita a partir de corpus já pré-classificado, ou seja para aprender é necessário um corpus cujas instâncias já têm a sua classe definida. Já a aprendizagem não supervisionada (Witten et al., 2011), não necessita de corpora pré-classificados. Por fim, uma aprendizagem semi-supervisionada (Witten et al., 2011) permite que sejam utilizados corpora classificados e também não classificados. Devido à dificuldade de encontrar um corpus pré-classificado, muitas vezes é usado um pequeno corpus etiquetado juntamente com um corpus muito maior não etiquetado (Witten et al., 2011).

Classificação

Os algoritmos de classificação pretendem atribuir, a determinados casos ou instâncias, uma classe dentro de um conjunto de classes. De seguida, são descritos alguns dos principais algoritmos aplicados na análise de texto.

- **Algoritmos que geram Árvores de Decisão** Uma Árvore de Decisão (Feldman, 2006) é uma estrutura que cujos nós internos correspondem às características ou atributos das instâncias, as ligações entre nós têm pesos associados aos atributos e as folhas representam as classes (ver Figura 2.13). Muitas vezes as Árvores de Decisão são aplicadas a problemas de classificação binária. Uma das vantagens é permitir que se perceba quais as decisões que o sistema tomou, ou seja, facilmente se percebe qual foi a características mais importante para o sistema, ao contrário de outros sistemas de classificação. Para a construção da árvore podem ser usando vários algoritmos como:

  - **ID3:** O algoritmo ID3 (Bell, 2014) Calcula a entropia de cada atributo e depois divide o corpus em subgrupos de acordo com o menor valor de entropia calculado. Este cálculo é repetido iterativamente usando os outros atributos. O nó raiz é escolhido de acordo com a informação ganha, ou seja é o nó com maior informação.

[18] Em inglês, *Supervised Learning*.

– **C4.5:** Tal como o ID3, o algoritmo C4.5 (Bell, 2014) é baseado na métrica de quantidade de informação. Uma das maiores diferenças é o facto de conseguir trabalhar com atributos contínuos.

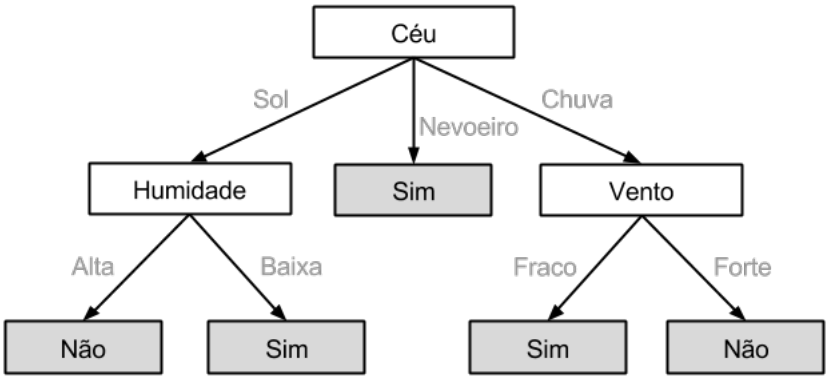

Figura 2.13: Exemplo de Árvore de Decisão.(Witten et al., 2011)

- **Máquinas de Vetores de Suporte**[19] Máquina de Vector de Suporte (MVS) (Feldman, 2006) é um dos algoritmos mais popular para problemas de classificação de polaridade de texto. Geometricamente, as MVS binárias produzem um hiperplano que separa as instâncias positivas das negativas. Esse hiperplano é escolhido durante o treino do modelo e é visto como o único hiperplano que separa as duas classes de instâncias com a maior distância entre elas. Tal como se pode ver na Figura 2.14, a distância entre as classes ou margem é vista como a distância entre o hiperplano e o ponto positivo e negativo mais próximo dele. Usando diferentes *ker-*

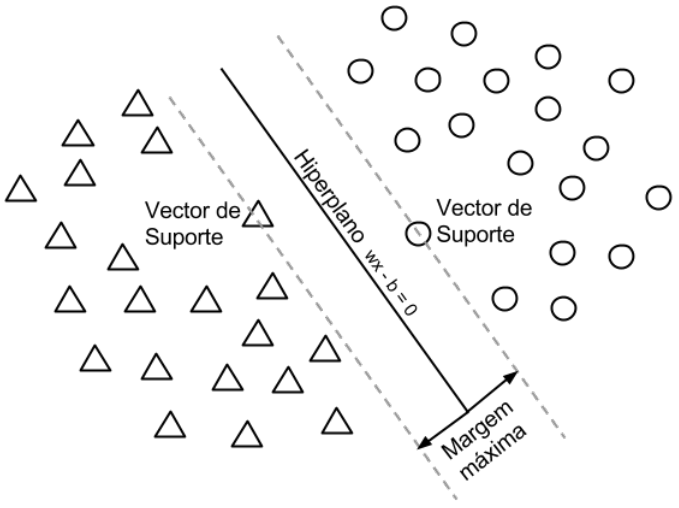

Figura 2.14: Visualização do melhor hiperplano encontrado pela MVS.

nels é possível gerar separações não lineares das classes, como por exemplo usando o *kernel* polinomial, RBF, entre outros. Os *kernels* são funções que permitem mapear os dados numa dimensão diferente à original. Como se pode ver na Figura 2.15, inicialmente (imagem à direita) temos os dados em duas dimensões que não são linearmente separáveis. Como tal, são mapeados para três dimensões usando

[19] Em inglês, *Support Vector Machines*.

uma função de *kernel* e aí já são linearmente separáveis (ver figura à esquerda) pelo plano representado pela cor verde.

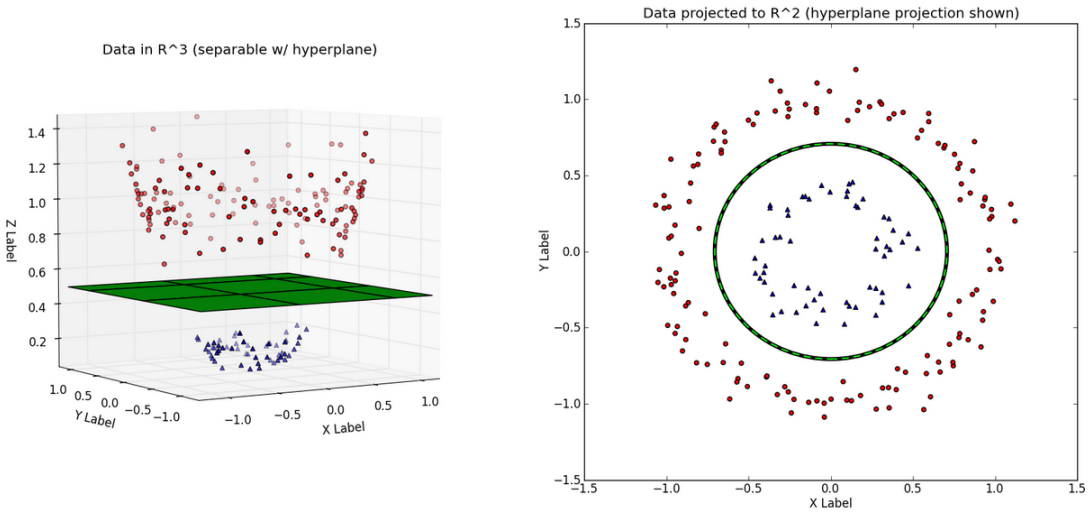

Figura 2.15: Exemplo de utilização de uma função *kernel* nas MVS (Kim, 2013).

- **Redes Neuronais** As Redes Neuronais (Aggarwal and Zhai, 2012b) são geralmente representadas por um sistema de neurónios interligados entre si que transformam os valores de entrada, de forma a produzir valores de saída. O neurónio é a unidade básica das Redes Neuronais, e cada um tem um peso associado. Usando o peso e o valor de entrada associado ao neurónio em questão, calcula-se o valor de saída de acordo com a função de ativação escolhida. Basicamente, é necessário treinar o sistema de forma a que ele escolha os melhores pesos associados a cada neurónio. Uma das ideias é inicializar esses pesos com valores aleatórios e gradualmente esses pesos são retificados sempre que um erro acontece. Quando a Rede Neuronal é composta por diversas camadas, esse erro tem de ser propagado por todas as camadas (ao que chamamos *back propagation*), obtendo assim uma rede *feedforward*. O grau de atualização dos pesos é regulado pela velocidade de aprendizagem pré-definida. Ou seja, se o grau de aprendizagem for muito alto os valores dos pesos vão ser alterados de forma mais abrupta. A rede neuronal mais simples é o perceptrão, representado na Figura 2.16, que possui apenas duas camadas, a de entrada e saída. Este tipo de rede é basicamente um classificador linear.

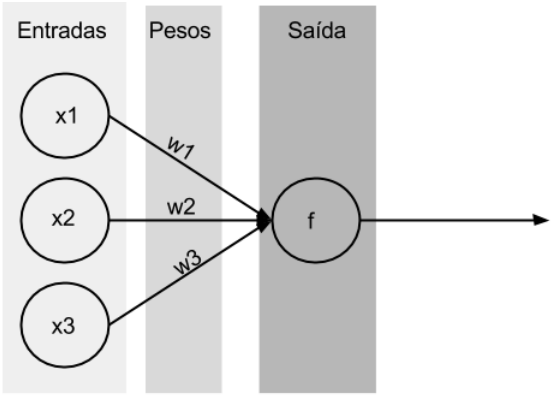

Figura 2.16: Exemplo de um perceptrão.

- **K-Vizinhos Mais Próximos**[20] K-Vizinhos Mais Próximos (Witten et al., 2011) é, para além de um classificador, um algoritmo baseado nas instâncias[21], em que os novos casos são classificados através das instâncias já classificadas e mais parecidas com a nova . Basicamente, no K-Vizinhos Mais Próximos, os novos casos são comparados com as K instâncias mais próximas já classificadas, usando uma métrica de distância. A métrica de distância deve ser pré-definida, sendo que geralmente usa-se a distância euclidiana (ver fórmula em 2.1) (Witten et al., 2011). As K instâncias mais próximas do novo caso ajudam a classificar a nova instância. Existem vários métodos para classificar a nova instância, sendo que o mais simples é atribuir a classe mais comum no grupo de instâncias mais próximas (Jackson and Moulinier, 2007).

$$\text{Distância Euclidiana} = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} \tag{2.1}$$

- ***Naïve Bayes*** *Naïve Bayes* é uma classificador que assume uma abordagem probabilística. Usa a Regra de Bayes (ver fórmula em 2.2), que representa a probabilidade de uma instância X pertencer a uma determinada classe C_i. Assume a independência das variáveis representativas das instâncias, ou seja assume que cada caraterística contribui independentemente para a probabilidade da instância pertencer a uma determinada classe, o que na vida real é uma assunção simplista, no entanto funciona eficientemente nos *datasets* reais (Witten et al., 2011).

$$P(C_i|X) = \frac{P(X|C_i) \times P(C_i)}{P(X)} \tag{2.2}$$

  Um problema que pode ocorrer é o caso de, numa nova instância, existir um valor de um atributo que nunca foi associado a determinadas classes durante o treino (Witten et al., 2011). Imaginemos um caso cujas variáveis são as condições meteorológicas e as classes são: poder ir jogar ou não ("*sim*" e "*não*"). Nos casos de treino, a variável *"céu"="sol"* está sempre associada à classe "*não*", então a $P(sol|sim) = 0$. Ou seja, a partir desses casos de treino sempre que estiver "*sol*", a instância nunca vai ser associada à classe "*sim*" segundo a Regra de *Bayes*, o que pode representar uma falha no modelo, porque os casos de treino não apresentaram todas as possibilidades (Witten et al., 2011).

Redução de Dimensionalidade

É comum representar documentos usando o modelo de Saco de Palavras ou *Bag-Of-Words*, no entanto esta representação resulta em vetores de grande dimensionalidade (cada dimensão é um termo da língua), o que pode tornar o sistema computacionalmente pouco eficiente. Uma das soluções é reduzir a dimensionalidade desse vetor (Crain et al., 2012). Existem diferentes métodos que o permitem fazer, sendo que em alguns se perde o significado de cada dimensão, como no caso dos métodos apresentados em baixo. Idealmente, cada nova dimensão devia estar associado a um novo e compreensível conceito. Para tal, podemos usar métodos de extração de tópicos, sendo os tópicos a nova dimensão.

- **Análise dos Componentes Principais** A ideia principal da Análise dos Componentes Principais (ACP) (Jolliffe, 2002) é reduzir o número de dimensões do corpus

[20] Em inglês, *K-Nearest Neighbor*.
[21] Em inglês, *Instance-Based Algorithm*.

retendo a máxima variância[22] nos dados. Basicamente, a ACP projeta os valores na direção onde estes mais variam. Essa direção é calculada através da matriz de covariância e os seus *eigenvetor*. Uma representação do ACP é observada na Figura 2.17, que transforma um corpus de duas dimensões num corpus de uma dimensão.

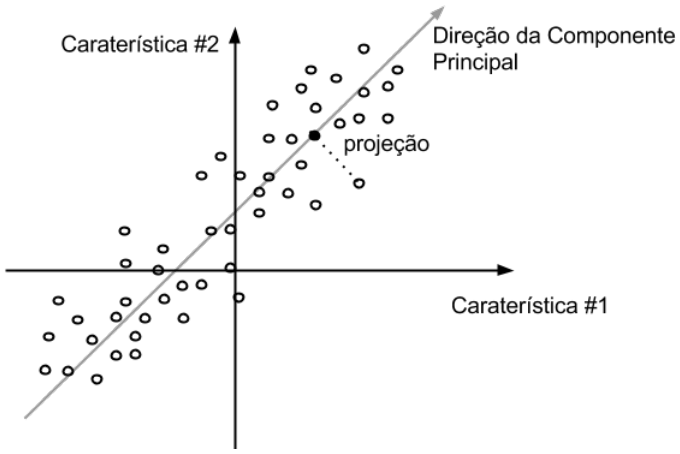

Figura 2.17: Representação da análise de componentes principais.

- ***Linear Discriminant Analysis*** A ideia principal é, como na ACP, procurar a melhor representação linear do corpus, no entanto *Linear Discriminant Analysis* (LDA) (Aggarwal and Zhai, 2012b) pretende encontrar uma projeção que evidencie a diferença entre as classes desse corpus, ao contrário de ACP que ignora a informação sobre as classes. Na Figura 2.18 é possível visualizar a transformação conseguida usando o LDA.

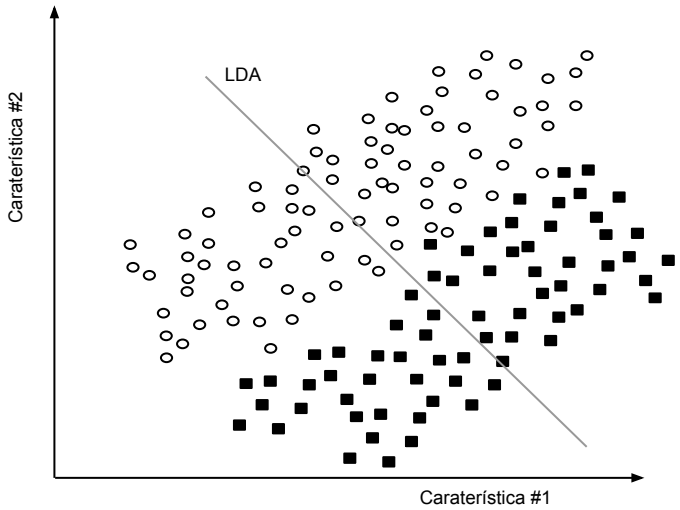

Figura 2.18: Representação da *Linear Discriminant Analysis*.

[22] Medida que representa o quão dispersos são os valores. Uma variância de 0 significa que todos os valores são idênticos, enquanto que uma variância alta significa que os valores estão todos longínquos.

Previsão de Estrutura

Num contexto textual, muitas vezes é importante não só as características de um instância em particular mas também das instâncias vizinhas. Por exemplo, se pretendemos construir um sistema de identificação de classes gramaticais, para além de definir a palavra a classificar é importante saber quais são as palavras anteriores na frase. Se a palavra anterior for um artigo definido, a palavra posterior tem uma grande probabilidade de ser um substantivo.

- ***Conditional Random Fields*** Quando se pretende classificar determinadas palavras de acordo com as suas classes gramaticais, é normal existir uma correlação entre as características que selecionamos para representar essas palavras. Quando uma característica tem uma grande correlação com outras características, esta torna-se é redundante. Como tal, um dos grandes problemas de ter características correlacionadas é que estas estão a acentuar determinados aspetos nos dados, o que pode influenciar a performance do sistema, principalmente se este assume que as características são totalmente independentes umas das outras. Vários modelos tentam criar uma distribuição de polaridade conjunta $P(y, x)$, através das variáveis de entrada e saída. No entanto, se as características tiverem dependências complexas, construir essas probabilidades é difícil. Uma das soluções é modelar através de uma distribuição condicional $P(y|x)$, onde as dependências deixam de ser importantes, que é basicamente o que as *Conditional Random Fields* (CRF) fazem (Sutton and McCallum, 2010).

Deep Learning

Deep learning é um conjunto de técnicas de aprendizagem automática que pretendem explorar diferentes níveis de representação e abstração, com o objetivo de captar complexas relações existentes nos dados (Deng and Yu, 2014). As representações de níveis mais altos são definidas pelas representações dos níveis mais baixos. O interesse nestas técnicas tem crescido nos últimos anos e uma das razões é o facto da capacidade de processamento ter aumentado drasticamente (Deng and Yu, 2014). Estas técnicas têm provado o seu sucesso em diversas áreas como, o reconhecimento de voz e escrita, processamento de áudio, processamento de linguagem natural, ente outras (Deng and Yu, 2014). *Deep Learning* é composto por diversos algoritmos como *Deep Belief Networks* (Hinton and Salakhutdinov, 2006), que é, basicamente, uma pilha de Máquina Restritas de *Boltzman* (Bengio, 2007), também uma técnica de *Deep Learning*. Outra técnica é o *Deep Autoencoder* (Deng and Yu, 2014), muito semelhante a um perceptrão com diversas camadas escondidas, no entanto a informação de saída é os próprios dados de entrada (Deng and Yu, 2014). Por fim, outra técnica que tem vindo a aumentar de popularidade na tarefa de extração de polaridade são as Redes Neuronais Convolucionais (RNC). Vários trabalhos com essa abordagem e resultados promissórios foram publicados como: (Ebert et al., 2015), (Severyn and Moschitti, 2015), (Chintala, 2012) e (dos Santos and Gatti, 2014). Como tal, as RNC são explicadas em mais detalhe de seguida.

- **Redes Neuronais Convolucionais** Um dos tipos de RNC (LeCun et al., 1998) mais populares é a LeNet-5. Na Figura 2.19 é possível observar uma LeNet-5 de uma forma simplista e aplicada a imagens para uma melhor compreensão, no entanto a sua aplicação a texto é semelhante.

  Recebe como entrada uma imagem, ou texto (representado por vetores/matrizes). Cada camada seguinte recebe um conjunto de variáveis (por exemplo, os valores de um pixel e os seus vizinhos). De forma a oferecer independência de relações

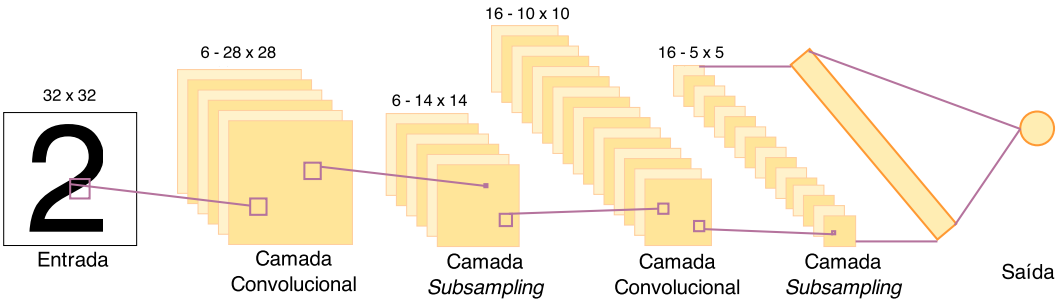

Figura 2.19: Arquitetura de uma RNC, mais concretamente uma *LeNet-5*.

espaciais (numa imagem, conseguir detetar o número "2"independente se ele está localizado mais à esquerda ou mais à direita), RNC usa a camada convolucional, que funciona como os filtros usuais das imagens (Deng and Yu, 2014). A camada de *subsampling* reduz os dados, por exemplo, agrupando os pixeis dois a dois, ficando com uma imagem com metade do tamanho original (Deng and Yu, 2014). Na última camada pode ser aplicado um algoritmo de classificação. RNC tem-se mostrado um algoritmo com bons resultados principalmente quando aplicado a reconhecimento de imagens (Ciresan et al., 2012).

2.2.3 Avaliação

De forma a percebermos se o sistema que construímos satisfaz o objetivo é necessário avaliá-lo. Existem vários métodos de avaliação e alguns deles são descritos de seguida.

- **Treino, Validação e Teste** Uma das métricas mais básicas é a medição do erro, que é basicamente a percentagem de casos num corpus que o sistema erra (Witten et al., 2011). Embora tenhamos um corpus de treino, este não é fiável para avaliação, uma vez que o sistema já aprendeu aqueles casos, e por isso o normal é obter um resultado otimista. Se esse sistema fosse avaliado com os dados de treino, este poderia estar a fazer *overfitting*, o que se traduz numa performance fraca quando o sistema encontra casos que nunca processou. De forma a prevenir o *overfitting*, para além do corpus de treino, deve-se criar um corpus de teste cujas instâncias nunca serão usadas para treino. Basicamente, treina-se com o primeiro corpus e, de seguida, processa-se as instâncias do corpus de teste, aplicando um método de avaliação. Embora o erro obtido seja mais relevante do que no primeiro caso, o sistema pode ainda fazer *overfitting*, uma vez que a tendência é ajustar os parâmetros do sistema de acordo com o erro no corpus de teste. Para resolver este problema, acrescenta-se mais um corpus para validação. Basicamente, o sistema é composto por 3 fases, inicialmente treina-se o modelo com o corpus de treino, de seguida ajusta-se os parâmetros de acordo com os erros obtidos com o corpus de validação e, por fim, calcula-se o erro no corpus de teste que representa o erro final e mais generalizado do sistema.

- ***Cross-Validation*** Uma das formas de estimar o erro do modelo criado é separar dados de treino em dados de teste usando o método *Holdout* (Witten et al., 2011). *Holdout* é um dos métodos mais básicos que permite reservar uma quantidade de instâncias para teste, por exemplo 70% das instâncias são para treino e os restantes para teste. Naturalmente, este tipo de separação pode resultar num corpus pouco representativo, por exemplo se na separação resultar a existência de poucas instâncias de uma determinada classe no corpus de treino, o que irá refletir no sistema de

uma forma negativa. Um outro algoritmo mais popular é *K-folds*, onde se divide igualmente os dados num número fixo de partições[23], e em cada iteração é escolhido uma partição dos dados para teste e o resto para treino. A partição escolhida para testes é sempre diferente até todas as partições terem sido usadas para teste. Por exemplo, no *10-fold cross validation*, divide-se as instâncias em 10 partições iguais, testa-se com $^1/_{10}$ dos dados e itera-se 10 vezes alterando sempre a partição de teste. Em cada uma dessas iterações obtém-se o erro e, no fim, é feito uma média desse erro.

- **Estimar o Custo** Num problema de classificação binário, ou seja com apenas duas classes, cada valor previsto pelo sistema pode resultar em Falso Negativo (FN), Falso Positivo (FP), Verdadeiro Positivo (VP) ou Verdadeiro Negativo (VN) como se pode observar na Figura 2.20 (Witten et al., 2011).

  O sucesso do sistema pode ser medido usando a seguinte fórmula:

$$\text{Exatidão} = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.3}$$

  O sucesso é basicamente o número de classificações corretas, dividido pelo número de classificações corretas e incorretas.

  Naturalmente, o erro é calculado pela fórmula:

$$\text{Erro} = 1 - \text{Exatidão} = 1 - \frac{VP + VN}{VP + VN + FP + FN} \tag{2.4}$$

  Outras das medidas usadas é o cálculo da Precisão[24], Abrangência[25] e Medida-F[26]:

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{2.5}$$

$$\text{Abrangência} = \frac{VP}{VP + FN} \tag{2.6}$$

$$\text{Medida-F} = 2 \times \frac{\text{Precisão} \times \text{Abrangência}}{\text{Precisão} + \text{Abrangência}} \tag{2.7}$$

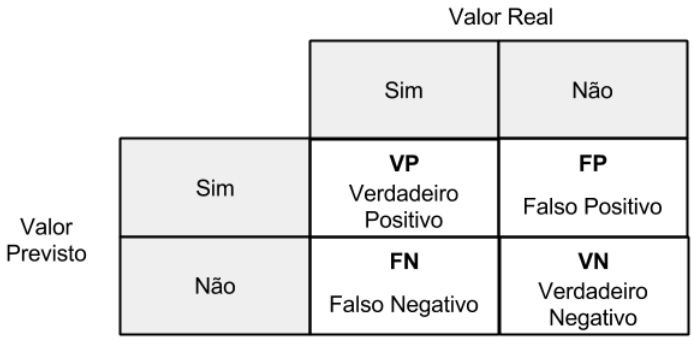

Figura 2.20: Matriz de Confusão para um problema de classificação binário.

[23] Em inglês, *folds*.
[24] Em Inglês, *Precision*.
[25] Em Inglês, *Recall*.
[26] Em Inglês, *F-Score*.

2.3 Extração de Opiniões

O texto pode se dividir em dois grandes grupos: texto objetivo e texto subjetivo (Liu, 2010). O primeiro expressa uma informação factual, como por exemplo uma notícia. Uma frase objetiva por vezes pode conter uma opinião implícita, como por exemplo a frase "*O meu telemóvel estragou-se em 2 dias.*" é objetiva, no entanto apresenta uma opinião. O texto subjetivo expressa uma opinião.

Extração de Opiniões ou *Opinion Mining* é o estudo computacional de opiniões expressas em forma de texto (Liu, 2010). Até há pouco tempo esta área tinha muito pouco desenvolvimento, porque antes da Internet muitas opiniões não eram registadas, eram apenas expressas em formato oral. Conseguir interpretar as opiniões dos outros é importante e útil, pois sempre que queremos tomar uma decisão, por exemplo, se devemos comprar ou não um determinado produto, gostamos de ter outras opiniões.

As opiniões podem ser dividas em dois tipos: opinião direta ou opinião comparativa (Liu, 2010). As opiniões diretas ocorrem quando se expressa uma opinião sobre um objeto de forma isolada, como por exemplo, na frase "O meu telemóvel é mau!". A opinião comparativa, tal como o nome sugere, ocorrem quando se expressa a opinião usando uma relação entre dois ou mais objetos, por exemplo, a frase "O meu telemóvel é pior que o teu.".

Tendo um texto subjetivo, para além de o interpretar, pretende-se conseguir representá-lo de uma forma mais estruturada. Uma das representações mais popular é o quíntuplo de uma opinião, cujas características são: Data, Autor, Entidade, Aspeto e Polaridade. A data representa a data em que a opinião foi publicada, enquanto que o autor é a pessoa ou organização que a expressa. Nem sempre o autor da publicação é o autor da opinião, como por exemplo na seguinte frase: "*O João odiou o gelado.*" em que o "*João*" é o autor da opinião e não a pessoa que transmitiu a informação. A extração de entidades, aspetos e polaridade são tarefas mais complexas, como tal, são explicadas em mais detalhe na secção 2.3.1 e 2.3.2

2.3.1 Extração de Entidade e Aspeto

As opiniões expressas têm um determinado alvo a que se chama entidade. As entidades podem ser produtos, serviços, pessoas, organizações, entre outros. Podem ser simples ou compostas por relações de pertença, como por exemplo a entidade "*computador*" tem diversos sub componentes como peso, memória, processador, etc (Liu, 2010). Na frase: "O meu *Iphone* funciona bem.", a palavra *Iphone* é uma entidade, neste caso um produto. O facto dos utilizadores muitas vezes se referirem à mesma entidade de formas diferentes, como por exemplo "*Motorola*" pode ser apresentado como "*Moto*" ou "*Mot*", torna esta tarefa mais complexa (Liu and Zhang, 2012).

O autor pode não querer expressar a sua opinião sobre toda a entidade, mas apenas sobre uma determinada característica da entidade. Neste caso, a opinião refere-se a um aspeto de uma entidade específica. Exemplificando, na frase "*A câmara do meu Iphone é muito boa!*", o aspeto é a "*câmara*", que pertence à entidade "*Iphone*".

Numa opinião é possível encontrar aspetos explícitos ou implícitos. O primeiro, o mais fácil de interpretar, é um aspeto expresso de uma forma clara, como por exemplo, na frase "*A velocidade da minha Internet é boa.*". Já numa frase com aspeto implícito, o aspeto tem que ser extraído através de outros indicadores, como adjetivos e advérbios (Liu, 2010). Por exemplo, na frase "*O meu telemóvel é muito grande.*", o adjetivo "grande" não é um aspeto, no entanto refere-se ao aspeto "*tamanho*".

Trabalhos Relacionados

Em (Hu and Liu, 2004) é proposto um sistema que usa métodos não supervisionados para encontrar aspetos numa opinião, assumindo que esses aspetos são sempre substantivos. Começa por extrair todos os substantivos, ordenando-os pela frequência em que aparecem no corpus, para mais tarde descartar os menos frequentes o que permite remover os substantivos irrelevantes. No entanto, a frequência não é suficiente para extrair todos os aspetos, por isso é proposto uma segunda extração usando as relações dos aspetos com as palavras que exprimem sentimento[27]. Basicamente, assume que a mesma expressão de sentimento usada para um determinado aspeto, é também usada para outros aspetos. Na Figura 2.21 observa-se a extração de um novo aspeto ou entidade em duas frases com semelhantes árvores de dependência.

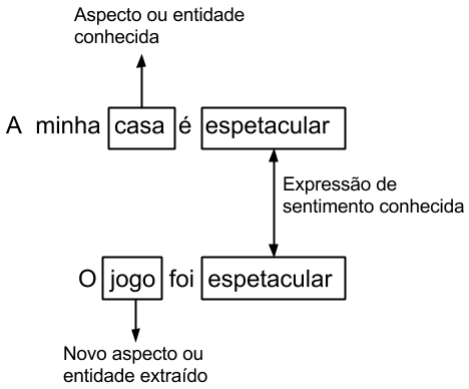

Figura 2.21: Extração de novos aspetos partindo de relações conhecidas.

Em (Popescu and Etzioni, 2005) sugerem-se alguns melhoramentos ao sistema anterior, usando o *Pointwise Mutual Information* (ou PMI) para remover alguns substantivos que podem não ser aspetos ou entidades. O PMI é calculado usando o substantivo e vários merónimos discriminantes.

$$PMI(a, d) = \frac{hits(a, d)}{hits(a) \times hits(d)}$$

Legenda:
hits - número de ocorrências
a - candidato a aspecto/entidade
d - merónimo

Figura 2.22: Cálculo de *Pointwise Mutual Information*.

Por exemplo, imaginemos um candidato a aspeto como "teclado", tendo como merónimos discriminantes a seguinte lista: "*é*", "*tem*" e "*de*". Usando a Web registamos o número de ocorrências de "*teclado é*", "*teclado tem*" e "*teclado de*", e para cada um calculamos o seu PMI (ver fórmula na Figura 2.22). Se esse valor for baixo quer dizer que o candidato poderá não ser uma aspeto ou entidade.

Em (Qiu et al., 2009) foi criado um sistema capaz de extrair tanto expressões de sentimento como aspetos simultaneamente, baseando-se no facto das opiniões terem um aspeto associado, ou seja, existir uma relação entre as expressões de sentimento e o aspeto. Inicialmente requer uma pequena lista de expressões de sentimento e aspetos que é usada

[27] Em inglês, *opinion words*.

para extrair novos aspetos e expressões que também serão usados para extrair outros aspetos e expressões. O processo é repetido até não existir nada de novo para extrair. A extração é conseguida através da análise das relações entre as expressões de sentimento e os aspetos, que resulta num conjunto de regras. Por exemplo, na frase "*O Iphone tira boas fotografias.*", o adjetivo "*boas*" é diretamente associado ao substantivo "*fotografias*" através da relação modificador. Ao conhecermos o adjetivo "*boas*" como uma expressão de sentimento e a regra do modificador, extrairmos o novo aspeto "*fotografias*".

Em (Jakob and Gurevych, 2010) é proposto um sistema que usa *Conditional Random Fileds* para extrair os aspetos. Usando como corpus de treino documentos provenientes de diferentes domínios, criaram várias características para entrada no sistema, como por exemplo: átomo, classe gramatical, dependência (perceber se o átomo tem uma dependência direta com uma expressão de sentimento) e opinião (se a frase exprime uma opinião).

Em (Su et al., 2008) foi proposto um método de *clustering* para identificar aspetos implícitos. Propõe que os aspetos possam ser deduzidos através das expressões de sentimento e a sua relação com aspetos explícitos.

Muitos dos trabalhos mencionados não fazem uma distinção clara entre entidades e aspetos. Em (Li et al., 2010) propõe-se um sistema que pretende, a partir de um conjunto de entidades extrair outras. No entanto, é um sistema dependente do domínio, ou seja, as entidades pré-listadas devem ser do mesmo tipo (por exemplo, marcas de telemóvel) que as entidades a extrair. Visto como um problema de expansão, tenta perceber qual a probabilidade de uma candidata a entidade ser mesmo uma entidade.

2.3.2 Extração de Polaridade

A polaridade representa a orientação da opinião. Geralmente a orientação está dividida em 3 classes: opinião positiva, negativa, ou neutra. No entanto, há opiniões mais fortes que outras, e essas três classes não chegam para classificar essas diferenças (Liu, 2010). Por exemplo, a frase "Eu *adooooooro* vir aqui comer!" expressa uma opinião positiva forte, enquanto que a frase "Eu até acho que se come bem." expressa uma opinião positiva mas não tão entusiasta. Uma das soluções, no entanto mais complexa, que permite ter em conta essas forças, é dividir cada classe em diferentes níveis, ou seja, classificar uma opinião positiva em, por exemplo, contente, feliz, encantada, excitada, sendo que o nível mais fraco seria contente e o mais forte excitada (Liu, 2010).

Esta área apresenta ainda grandes desafios, sendo que a deteção de ironia é um dos maiores. Outros desafio é a extração da polaridade em línguas que não sejam inglês, pois o número de recursos disponíveis é menor, dificultando os avanços. Classificar opiniões de acordo com a polaridade é uma tarefa subjetiva, ou seja, por exemplo, o que para uns pode ser positivo para outros pode ser neutro.

Trabalhos Relacionados

Em (Pang et al., 2002) é criado um sistema clássico de aprendizagem supervisionada para classificar opiniões sobre filmes em positivas e negativas. Mostrou que usando apenas unigramas se conseguia ultrapassar um sistema que escolhesse de forma aleatória as classes. Como entrada no sistema usou as seguintes características: unigramas, bigramas, classes gramaticais, adjetivos e posição. Embora tenha experimentado três algoritmos diferentes de classificação, o que se mostrou melhor foram as Máquinas de Vetor de Suporte, tendo como características unigramas e bigramas.

Em (Pang and Lee, 2005) pretendeu-se alargar o conjunto de classes (por exemplo, classificar com pontuação de 1 a 5 estrelas). Neste caso, o problema é visto de uma perspetiva de regressão que assume que as classes são extraídas, tornando-as discretas, a

partir de uma função contínua.

Muitas vezes um classificador que foi treinado num corpus concentrado num determinado domínio, mostra maus resultados quando lhe são apresentados novos casos de domínios diferentes. Isto deve-se ao facto da mesma palavra poder ser positiva num contexto mas negativa noutro. Em (Pan et al., 2010) pretende-se minimizar esse erro, separando, em grupos, palavras que diferem de contexto para contexto e palavras independentes do contexto. A informação extraída desses grupos é usada em classificadores que assim estão melhor preparados para textos de diferentes domínios.

O sistema Pulse (Gamon et al., 2005) classifica opiniões em positiva, negativa e outra, usando uma aprendizagem semi-supervisionada. Começando com poucas frases etiquetadas, treina-se um classificador *Naïve Bayes*, que é usado para estimar a distribuição probabilística das classes nos documentos não etiquetados.

Em (Ding et al., 2008) é proposta uma abordagem baseada em léxicos. É um sistema capaz de lidar com expressões de sentimento dependentes do contexto. Por exemplo, na frase "*A bateria dura muito.*" é pouco claro se "*muito*" é positivo ou negativo. O sistema tenta encontrar outros casos em que "*muito*" é positivo para a duração da bateria. Se por exemplo existir um caso como: "*A câmara tira boas fotografias e a bateria dura muito.*" é possível extrair que se a duração da bateria for muita é positivo porque a frase possui a expressão positiva "*boas*". É assumido que uma frase só pode transportar uma polaridade, excepto se estiver subdividida através de expressões tipo "*mas*"[28], que sugerem diferentes polaridades na mesma frase. Este sistema também tem em conta expressões que podem mudar o sentido de uma palavra[29]), como por exemplo "*não*" e "*sem*".

Um outro sistema, com uma abordagem bastante diferente das anteriores, é apresentado em (Tang et al., 2014). O trabalho propõe dois tipos de representações das opiniões. Primeiro, a representação mais clássica, ou seja, com características linguísticas, como *smiles*, léxicos de sentimento, negações, pontuação, entre outras. A segunda representação começa por traduzir os casos transformando-os em *word embeddings* contendo informação sobre a polaridade e usando redes neuronais. Essas duas representações são agrupadas e são usadas para treino em Máquinas de Vetor de Suporte.

2.4 Redes Sociais

As redes sociais apresentam uma grande variedade de informação passível de ser extraída e estudada. A quantidade de informação disponível leva a que técnicas como a análise de dados, pesquisa, análise de texto, análise de imagens, entre outras, possam ser aplicadas a este tipo de texto.

Redes sociais podem ser definidas como uma rede de interações e relações, que partem de atores (pessoas, organizações) para atores (Aggarwal, 2011). Naturalmente, uma rede social não se restringe apenas a redes construídas na web, no entanto é nessas que este trabalho se foca. Alguns exemplos das redes sociais mais populares são *Facebook*, *Twitter* e *LinkedIn*[30]. Também se pode considerar como redes sociais, serviços que permitam a partilha de conteúdo como principal foco, tal como o *Youtube*[31], o *Instagram*[32], entre outras (Hu and Liu, 2012). No entanto, mais uma vez, o foco deste trabalho concentra-se apenas em redes sociais com partilha de texto, mais precisamente no *Facebook* e *Twitter*.

[28] Em inglês, *But-clauses*

[29] Em inglês, *Opinion shifters*.

[30] Rede social para profissionais. Disponível em `https://pt.linkedin.com/`

[31] Permite a partilha de vídeos. Disponível em `https://www.youtube.com/`

[32] Permite a partilha de fotografias e vídeos e também a aplicação de filtros digitais. Disponível em `instagram.com`

O texto produzido nas redes sociais apresenta características bastante distintas do texto mais tradicional encontrado por exemplo, em livros, o que acrescenta grandes desafios na sua análise. Uma das características é que o texto é sensível ao fator tempo. Muitas vezes, o mesmo utilizador é capaz de partilhar informações várias vezes ao dia, e é importante perceber a ordem dessas publicações dado que muitas vezes pode fornecer contexto útil, principalmente na monitorização de eventos. Em (Sakaki et al., 2010) é apresentado um estudo que afirma ser possível estabelecer uma relação entre informação publicada no *Twitter* e eventos reais, dando como exemplo que quando um terramoto ocorre é possível detetá-lo simplesmente observando os novos *tweets*[33].

Outra característica importante é o tamanho do texto, que é normalmente bastante mais reduzido. Inclusivé, algumas redes sociais limitam o tamanho do texto a publicar, tal como o *Twitter* que limita o texto a 140 caracteres. Pequenos textos podem-se traduzir em textos que contêm apenas a informação mais relevante, no entanto isso revela mais desafios para métodos tradicionais da análise de texto, como *clustering* de texto, classificação, extração de opinião (Hu and Liu, 2012).

As redes sociais também permitiram a entrada de novos átomos, como os *smiles*, as *hashtags* e menções (ver Figura 2.23). Esses novos átomos devem ser analisados à parte, uma vez que muita das vezes não fazem parte do texto em si, mas são uma informação complementar.

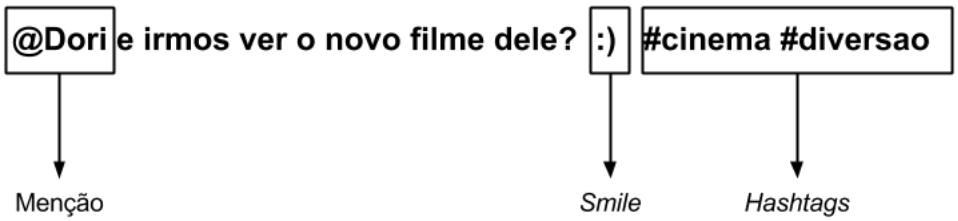

Figura 2.23: Exemplo de um *tweet*.

Outro grande desafio é a falta de estrutura nos textos partilhados. Embora se encontre alguns textos bem estruturados, existem muitos que não o são, dificultando a sua análise. Para além da estrutura, os textos passaram a ter mais erros, abreviações, acrónimos, etc. Por exemplo, textos como "*Eu n kero k tu me xateies!*", apresentam palavras que não o são realmente, o que dificulta a interpretação destas. Embora existam algumas regras já conhecidas, como a substituição de "que" por "k", cabe a cada um inventar uma nova representação dessa palavra o que torna difícil encontrar todas as regras para resolver as situações (Hu and Liu, 2012).

Facebook[34]

O *Facebook* é atualmente o segundo site mais visitado da Internet[35], sendo assim a rede social mais popular. Cada utilizador gere o seu próprio perfil, onde se guarda informação sobre a pessoa (nome, idade, trabalho, etc), interesses e amigos. Essas informações podem ser definidas como privadas, em que por exemplo apenas os seus amigos podem aceder, ou públicas em que todos os utilizadores da rede social têm acesso. Cada utilizador pode partilhar texto, fotografias ou vídeos. O *Facebook* permite a criação de eventos e grupos de utilizadores que partilham interesses em comum. O utilizador tem acesso ao *feed* de notícias que lista todas as partilhas e alterações ao perfil dos outros utilizadores seus

[33]Nome dado ao conteúdo partilhado na rede social *Twitter*.
[34]Disponível em `https://www.facebook.com/`
[35]Informação extraída de do site Alexa. Disponível em `http://www.alexa.com/`

amigos. Sempre que um utilizador faz uma publicação, é permitido aos outros utilizadores fazerem gosto[36] ou partilhar a publicação.

Twitter[37]

O *Twitter*, em contraste com o *Facebook*, é uma rede social mais focada nas publicações do que no perfil de cada utilizador. Permite apenas fazer publicações até 140 caracteres, a que chamam de *tweet*. Num *tweet* é possível mencionar outros utilizadores (menções) e colocar *hashtags* que geralmente associam o *tweet* a um tópico. Os utilizadores podem seguir e ser seguidos[38] por outros utilizadores.

2.5 Recursos Linguísticos

Nesta secção são apresentados alguns recursos linguísticos que a *Wizdee* já possui, e outros que poderão ser relevantes para o trabalho proposto.

2.5.1 Corpora Linguístico

Nesta secção são apresentados diversos corpora linguísticos tanto para a Língua Portuguesa como para Língua Inglesa. No fim da secção é feita uma análise comparativa dos corpus apresentados.

Língua Portuguesa

CETEMPúblico[39] (Rocha and Santos, 2000) É um corpus que possui cerca de 180 milhões de palavras em português, extraídas de 2.600 edições do Jornal Público[40], desde 1991 e 1998. Cada extrato está dividido em parágrafos e frases, cujos títulos e autores são conhecidos. A cada palavra está associada a informação gramatical, que foi etiquetada de forma automática usando o analisador sintático PALAVRAS (Bick, 2000).

CHAVE[41] Disponibilizado pela *Linguateca*[42], é uma coleção que contém textos completos (cerca de 1400 edições) tanto do jornal português Público, como do jornal brasileiro Folha de São Paulo, nas datas 1994 e 1995. Em 2007 foram acrescentadas anotações sintáticas, usando o analisador PALAVRAS (Bick, 2000). Em 2010, a coleção passou a ter anotações automáticas de entidades mencionadas, usando REMBRANDT (Cardoso, 2008).

BOSQUE[43] Bosque é um recurso criado no Projeto Floresta Sintática[44] (Afonso et al., 2002). Tal como a coleção CHAVE, o Bosque também possui textos do Jornal Público e do Jornal Folha de São Paulo, e foi etiquetado usando o PALAVRAS (Bick, 2000). Uma das grandes vantagens deste corpus é que foi revisto na integral por linguistas, ao contrário dos dois anteriores.

[36] Botão *like* do *Facebook* que exprime que o utilizador gostou ou concorda com a publicação.
[37] Disponível em `https://twitter.com/`
[38] Em inglês, *follow* e *follower*.
[39] Disponível em `http://www.linguateca.pt/ACDC/`.
[40] Disponível em `http://www.publico.pt/`.
[41] Disponível em `http://www.linguateca.pt/ACDC/`.
[42] Disponível em `http://www.linguateca.pt/`
[43] Disponível em `http://www.linguateca.pt/floresta/corpus.html`.
[44] Disponível em `http://www.linguateca.pt/floresta/`.

Língua Inglesa

Corpus de Linguagem independente de Reconhecimento de Entidades Mencionadas[45] Este corpus que foi criado para a Conferência CoNLL 2003[46]. É um subconjunto de um corpus maior o *RCV1*[47]. Contém entidades, como organizações, pessoas, tempo, quantidades, etc. Uma das suas vantagens é que foi etiquetado manualmente.

Corpus de Opiniões de Filmes[48] (Maas et al., 2011) Este recurso apresenta cerca de 50.000 opiniões sobre filmes extraídos da web. Classifica as opiniões em positivo ou negativo.

Corpus de Opiniões de Texto do *Twitter*[49] (Nakov et al., 2013) Este corpus foi criado pelos organizadores do *Workshop on Semantic Evaluation* (SemEval-2014[50]). Apresenta mais de 10.000 textos publicados no *Twitter* que foram etiquetados como positivos, negativos ou neutros, usando uma aplicação de *crowdsourcing*.

Corpus de Extração de Aspectos Implícitos[51] (Ivan Omar Cruz-Garcia, 2014) É baseado num outro corpus de opiniões de clientes, extraindo apenas o texto e etiquetando manualmente os aspetos implícitos. Os autores publicaram também uma ferramenta que extrai automaticamente esses aspetos, no entanto não está disponível para contexto comercial.

Análise aos Corpus Identificados

Na Tabela 2.1 é feita uma análise geral dos vários corpus apresentados. Embora o trabalho proposto se foque no texto extraído de redes sociais, muitos dos corpus usam um tipo de texto um pouco mais formal. Isto deve-se ao facto da escassez de recursos para o tipo de texto pretendido, principalmente para a Língua Portuguesa.

Sobre os corpus de Língua Portuguesa e comparando os primeiros quatro que oferecem informações semelhantes, é possível observar que muitos deles provêm de uma fonte jornalística, sendo que, maior parte, são anotados automaticamente, à exceção do Bosque que foi revisto manualmente. É relevante apontar que nenhum desses corpus pode ser usado num âmbito comercial.

Sobre os corpus ingleses, a maior parte dos apresentados foram conseguidos de forma automática, sendo que o tipo de texto é mais variado que nos corpus portugueses.

2.5.2 Dicionários

De seguida, são apresentados alguns dicionários já utilizados pela *Wizdee*. Estes dicionários são úteis principalmente em tarefas de PLN. Os recursos apresentados estão disponíveis tanto na Língua Portuguesa como na Inglesa.

[45] Disponível em `http://www.clips.uantwerpen.be/conll2003/ner/`.
[46] Disponível em `http://www.clips.uantwerpen.be/conll2003/`
[47] Disponível em `http://trec.nist.gov/data/reuters/reuters.html`
[48] Disponível em `http://ai.stanford.edu/~amaas/data/sentiment/`.
[49] Disponível em `http://alt.qcri.org/semeval2014/task9/index.php?id=data-and-tools`.
[50] Mais informações em `http://alt.qcri.org/semeval2014/`
[51] Disponível em `http://www.gelbukh.com/resources/implicit-aspect-extraction-corpus/`.

Recurso	Língua	Tipo de Anotação	Licença	Origem do Texto
CETEMPúblico	PT	Automática	Académica	Jornais
BOSQUE	PT e BR	Revista Manualmente	Académica	Jornais
CHAVE	PT e BR	Automática	Académica	Jornais
Linguagem Independente de REM	ING	Manual	Académica	Jornais
Opinião de Filmes	ING	Automática	Domínio Publico	Informal
Opiniões de Tweets	ING	Manual	Domínio Publico	Informal
Aspetos Implícitos	ING	Manual	Académica	Informal

Tabela 2.1: Análise comparativa dos diferentes corpus linguísticos.

Lista de Abreviações Este dicionário, para além das abreviações mais conhecidas, também contém *pitês*. *Pitês* é uma linguagem muito popular na web usada geralmente para permitir a comunicação mais rápida. Nessa língua é frequente a substituição de sílabas e a supressão de variáveis. É um dicionário bastante importante, principalmente na análise de texto extraído de redes sociais. Por exemplo, "*qt*" representa "*quanto*", "*aki*" representa "*aqui*", "*ixo*" representa "*isso*".

Lista de Palavrões É uma lista que contém expressões geralmente associadas ao insulto, tais como "*estúpida*", "*idiota*", etc.

Lista de *Stopwords*[52] Dicionário contém as palavras portuguesas/inglesas mais frequentes. Em tarefas de PLN é comum que estas palavras sejam ignoradas pelo analisador.

Lista de Verbos Regulares e Irregulares Este dicionário contém os verbos regulares e irregulares da Língua Portuguesa ou inglesa na sua forma infinitiva. No caso dos verbos irregulares contém as suas diferentes conjugações

Lista de Lemas Dicionário que traduz palavras nos seus lemas de acordo com a sua parte de discurso. Por exemplo, "*ameaçou*" é convertido em "*ameaçar*", ou "*ampliado*" em "*ampliar*".

Lista de Géneros Lista que possui palavras que são consideradas exceções das regras de transformação de género. Por exemplo, a palavra "*ladrão*" e "*ladra*".

Lista de Singularidades Tal como a anterior, é uma lista palavras que são consideradas exceções das regras de singularização.

***JaSpell*[53]** Possui diferentes dicionários sendo que o seu objeto é a deteção e correção de erros de escrita comuns. Oferece um dicionário da Língua Portuguesa, outros dicionários de palavras que devem ser ignoradas quando se tenta fazer correção de erros e também

[52]As palavras mais frequentes do português/inglês que não acrescentam informação, como por exemplo "*e*", "*de*", "*na*".

[53]Disponível em `http://jaspell.sourceforge.net/`

uma lista dos erros mais comuns na Língua Portuguesa, tais como: "*bébé*" que deveria ser "*bebé*", "*beje*" que seria "*bege*","*femenino*" que seria "*feminino*". Este recurso existe apenas para o português.

Dicionários de Polaridade Portugueses

- **SentiLex-PT**[54] O SentiLex (Silva et al., 2012) é um léxico de sentimento em português, particularmente interessante para tarefas de extração de opiniões. Para além da polaridade, descreve o lema, o alvo da polaridade, a classe gramatical e o método de atribuição da polaridade. Maior parte do léxico foi etiquetado manualmente, sendo que algumas entradas adjectivais foram classificadas usando uma ferramenta criada pelos autores do projecto.

- **Lista de Polaridades** Este recurso foi construído dentro pela *Wizdee*, com o auxílio de dois recursos: o SentiLex-PT e o MPQA descrito na secção de dicionários ingleses. O processo de criação deste novo léxico passou pela tradução do MPQA, uma vez que é um léxico de sentimento direcionado para a Língua Inglesa. De seguida, fez-se a reunião dos dois léxicos, eliminando os casos em duplicado.

Dicionários de Polaridade Ingleses

Na Língua Inglesa existem uma quantidades de recursos muito maior comparativamente com a Língua Portuguesa. Na Tabela 2.2 são apresentados alguns dos corpus mais populares.

Recurso	Licença	Descrição
Opinião de *Bing Liu*[55]	Domínio Publico	Cerca de 2.000 palavras positivas e 4.800 negativas.
Lista *AFINN*[56]	Domínio Publico	Cerca de 2.500 palavras com nível de polaridade entre -5 e 5.
Sentiment140	Contactar	Unigrams, bigramas e pares com polaridade de $-\infty$ a ∞.
Hashtags de NRC	Contactar	Divide em categorias: raiva, medo, alegria, surpresa,...
Subjetividade MPQA	Académica	Cerca de 8.000 palavras de forte e fraca subjetividade.

Tabela 2.2: Análise comparativa dos diferentes dicionários de polaridade.

2.6 Ferramentas

Nesta secção estão listadas algumas ferramentas que são úteis para a realização do trabalho proposto. Algumas já são usadas pela *Wizdee*, sendo que se teve o cuidado de listar as que podem ser usadas comercialmente.

OpenNLP[57] É uma ferramenta da *Apache*[58], em Java, que permite executar várias tarefas de PLN usando algoritmos de aprendizagem automática[59]. Oferece separação

[54]Disponível em `http://dmir.inesc-id.pt/project/SentiLex-PT_02`
[55]Disponível em `http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon`
[56]Disponível em `http://neuro.imm.dtu.dk/wiki/AFINN`
[57]Disponível em `https://opennlp.apache.org/`
[58]Ver em `http://www.apache.org/`
[59]Em Inglês, *Machine Learning*.

de *tokens* e frases, identificação de partes do discurso, extração de entidades, entre outras. Suporta diferentes línguas, incluindo o Português e o Inglês, no entanto na primeira o número de funcionalidades disponíveis é menor. De momento, a *Wizdee* usa esta ferramenta, tanto para a Língua Inglesa, como para a portuguesa.

***Jazzy* e *JaSpell*[60]** Ambas as ferramentas permitem a correção de erros em texto. *Jazzy* é uma API para aplicações Java para a Língua Inglesa. Já o ferramenta *JaSpell* é direcionada, tanto para a Língua Portuguesa, como para a inglesa.

***Scikit-Learn*[61]** Tal como as ferramentas anteriores, permite o uso de algoritmos de aprendizagem automática, como redução de dimensionalidade, pré-processamento, seleção de modelos, etc. É uma ferramenta bastante completa e popular, pela sua facilidade de uso e organização. Ao contrário das anteriores, esta ferramenta não fornece API para Java, apenas para Python.

***Theano*[62]** É uma biblioteca Python que permite definir, optimizar, avaliar expressões matemáticas, tirando vantagens dos *Graphics Processing Unit*s (GPUs) mais recentes. Implementa diversos algoritmos de *deep learning* usando técnicas de optimização.

***Word2vec*[63]** Esta ferramenta implementada em C, permite a criação de vectores de representação de palavras, chamadas também *Word Embeddings* (Bengio et al., 2013). Implementa dois tipos de algoritmos: *bag-of-words contínuo* e *skip-gram* (Mikolov et al., 2013). Já possui um modelo treinado usando um corpus da *Google News*[64].

Análise das Ferramentas Identificadas

Na Tabela 2.3 podemos fazer uma análise comparativa das diferentes ferramentas. Todas elas permitem a sua utilização num ambiente comercial e fornecem API em Java [65], à exceção da *Scikit-Learn*, Word2vec e Theano que são direcionadas para Python ou C. As últimas duas ferramentas da lista estão vocacionadas para o tema de deep learning.

Ferramenta	Língua	Serviço	Java	Licença
OpenNLP	PT e ING	Parte de discurso, REM, *tokens*...	Sim	*Apache License v2.0*
Jazzy	ING	Corretor de erros	Sim	*LGPL v2*
JaSpell	PT e ING	Corretor de erros	Sim	*BSD License*
Scikit Learn	–	Aprendizagem automática	Não	*BSD license*
Theano	–	Otimização de expressões, *deep learning*	Não	*BSD license*
Word2vec	–	Vectores de representação de palavras	Não	*Apache License v2.0*

Tabela 2.3: Análise comparativa das diferentes ferramentas.

[60] *Jazzy* disponível em `http://jazzy.sourceforge.net/` e *JaSpell* em `http://jaspell.sourceforge.net/`

[61] Disponível em `http://scikit-learn.org/stable/`

[62] Disponível em `http://deeplearning.net/software/theano/`

[63] Disponível em `https://code.google.com/p/word2vec/`

[64] Disponível em `https://news.google.pt/`

[65] Neste trabalho a linguagem Java tem especial importância uma vez que a plataforma Wizdee está maioritariamente escrita nessa linguagem.

Capítulo 3

Análise de Competidores

Sistemas de extração de opiniões são, hoje em dia, bastante populares, por isso existe uma grande variedade de produtos. No entanto, as funcionalidades oferecidas são geralmente muito semelhantes. Nesta secção, são descritos alguns produtos que se assemelham ao trabalho que é proposto neste documento, ou seja que permitem a extração de opiniões, com especial atenção aos produtos que suportam tanto a Língua Portuguesa como a Língua Inglesa.

3.1 *Semantria*[1]

É uma empresa de análise de texto concentrada na extração de opiniões. O *Semantria* é disponibilizado em forma de API (compatível com Java, Python, C++, e outras) ou em forma de *plugin* para o *Excel*. Permite análises em treze línguas diferentes, incluindo o Português e o Inglês. É necessário que o utilizador forneça os dados a analisar, pois o *Semantria* não extrai automaticamente os dados de redes sociais.

Permite vários tipo de análise como, por exemplo, a extração da polaridade, extração de intensificadores e identificação de expressões que influenciam a polaridade de textos individuais. Faz análises mais globais, tendo em conta a coleção de documentos, por exemplo extração de *facets*[2], que são vistas como categorias, extração de entidades e o seu tipo (por exemplo, a entidade "Las Vegas" é do tipo Localização), e outras.

Uma vez que o Semantria disponibiliza uma demo[3], esta foi testada com o seguinte texto:

"*I think I may have some problems with my new Iphone 5c. My battery is always too hot, i can't even touch my phone!! I don't know what to do... Please, Apple help me.. Also I own a brand new Iphone and the camera takes really bad pictures, it's really disappointing.*"

Na tabela 3.1 é possível observar as informações extraídas. Como se pode ver, a polaridade geral da frase foi extraída de forma correta para ambas as línguas. Para calcular a polaridade o Semantria afirma basear-se numa abordagem linguística[4], onde extrai expressões relacionadas com sentimento e a essas expressões calcula a pontuação de cada uma usando o PMI. Ainda na tabela é possível observar que deteta corretamente a entidade "Apple", no entanto é incapaz de detetar a entidade "Iphone 5c". Para extração de entidades, o

[1]Disponível em `https://semantria.com/`

[2]Palavras mais frequentes na coleção de documentos.

[3]Disponível em `https://semantria.com/demo`

[4]Ver descrições das abordagens em `https://semantria.com/support/resources/technology`

Semantria apenas usa uma lista pré-concebida. Ou seja, se a entidade não estiver na lista não é reconhecida. No entanto, permite a cada cliente registar as suas próprias entidades. Para além das entidades, o Semantria é capaz de extrair a polaridade associada à mesma. O Semantria também permite a extração de temas, que se define como todos os sintagmas nominais do texto. Por fim, o texto é caracterizado dentro de categorias, que são extraídas através de informações do Wikipedia.

Língua	**Polaridade**	**Influenciadores de polaridade**	**Entidade (Polaridade)**	**Temas (Polaridade)**	**Categorias (Relevância)**
EN	Negativa (-0.446)	"*disappointing*" "*brand new*" "*bad*" "*problems*"	"*Apple*" (neutra)	"*bad pictures*" (negativa) "*really disappointing*" (negativa)	"*Mobile Phone*" (0.80) "*Electronics*" (0.76)
PT	Negativa (-0.507)	"acabado" "ajuda" "sei" "desapontado" "alguns problemas"	"*Apple*" (positiva)	"minha bateria" (negativa) "câmara tira" (negativa) "demasiado quente" (negativa)	—

Tabela 3.1: Resultados da demo da *Semantria*.

3.2 *AlchemyAPI*[5]

É um produto da empresa AlchemyAPI, fundada em 2009, que fornece, entre outros produtos, uma API capaz de processar linguagem natural, usando técnicas de aprendizagem automática, estando concentrada em extração de opiniões.

Dado um texto, extrai as suas entidades que possuem uma relevância no documento, uma polaridade associada (positiva, negativa ou neutra), e um tipo (Pessoa, Cidade, Quantidade). Permite a extração de palavras-chave e as suas características, como por exemplo a palavra-chave "céu limpo" possui uma relevância de 0.6 e a sua polaridade é positiva. Permite associar o documento a uma determinada categoria (por exemplo, meteorologia, tecnologia,...) e o seu nível de confiança nessa associação.

O AlchemyAPI suporta diferentes línguas em algumas das funcionalidades, incluindo Português e o Inglês. No entanto, para extração de relações e criação de taxonomia o Português não é suportado.

Tal como para o Semantria, o AlchemyAPI fornece uma demo e como tal, foi testada a mesma frase apresentada no Semantria, cujos resultado se podem observar na tabela 3.2.

Língua	**Polaridade**	**Palavras-Chave (Polaridade)**	**Entidade (Polaridade)**	**Categorias**
EN	Negativa (-0.66)	"*new Iphone*" (negativa) "*bad pictures*" (negativa) "*brand new Iphone*" (negativa) "*Apple help*" (neutra) "*battery*" (neutra) "*camera*" (negativa) "*problems*" (negativa)	"*Apple*" (neutra) "*Iphone*" (negativa)	"*portable entertainment*" "*tablet*" "*computer*"
PT	Negativa (-0.73)	—	"*Apple*" (—)	—

Tabela 3.2: Resultados da demo do *AlchemyAPI*.

[5]Disponível em `http://www.alchemyapi.com/products/alchemylanguage/`

Como se pode observar, a diferença de análise entre as duas línguas línguas é significativa, em que para o Português a ferramenta não conseguiu extrair qualquer palavra-chave e não suporta a extração de categorias (ou taxonomia). Quanto à extração de polaridade, esta ferramenta permite extrair a polaridade do texto, e a polaridade associada à entidade e palavras-chave. Para calcular a polaridade AlchemyAPI afirma usar uma abordagem linguista[6], também semelhante ao Semantria, que se concentra no utilização de palavras de opinião e inversores de polaridade.

3.3 *SAS Sentiment Analysis*[7]

Tal como os anteriores, este produto permite, a partir da análise de texto, classificá-lo de acordo com a sua polaridade, usando técnicas de modelação estatistica e técnicas de PLN baseado em regras. Inclui *plugins* de ligação com as redes sociais mais populares, como o *Facebook*, *Twiiter*, *LinkedIn*, entre outros. Cada documento tem associadas diversas entidades que incluem vários aspetos, em que cada aspeto tem uma polaridade associada. Fornece também informação sobre a categoria em que o documento se insere.

Uma das suas grandes características é o facto de possibilitar feedback por parte do utilizador, de forma a melhor e personalizar o sistema de extração de opiniões. Permite a pesquisa de textos segundo palavras chave, categorias e aspetos.

3.4 *Clarabridge*[8]

É uma coleção de produtos que propõem oferecer uma visão global das opiniões dos clientes de determinada empresa. Transforma texto não estruturado, como e-mails, redes sociais, informação dos *call centers*, em informação estruturada.

É um produto completo em que todas as fases necessárias são processadas pelo mesmo, isto é, recolhe e extrai informação de forma automática. Para além de texto, permite transcrever conversas e depois analisar a sua transcrição. Permite extrair a polaridade segundo um range de valores entre -11 e 11, usando uma abordagem linguística. Para além disso, Clarabridge afirma extrair polaridade tendo em conta o contexto, usando uma lista pré-definida de palavras que em diferentes contextos tenham polaridades diferentes.

3.5 *Lymbix*[9]

É uma ferramenta que promete melhorar o suporte a clientes e a gestão da marca extraindo sentimentos de uma forma mais completa do que a maioria dos produtos no mercado. Defende que extrair apenas uma polaridade positiva e negativa não é suficiente para uma análise correta e confiante. Para tal, além da polaridade básica classificam o texto em diferentes níveis de emoções dependendo da sua intensidade, como por exemplo tristeza, medo, humilhação, contente, entre outros.

[6] Ver abordagens em `http://www.alchemyapi.com/resources/solutions/social-media-monitoring`

[7] Disponível em `http://www.sas.com/en_us/software/analytics/sentiment-analysis.html`

[8] Disponível em `http://clarabridge.com/`

[9] Disponível em `http://www.lymbix.com/`

3.6 Comparação de Competidores

Na Tabela 3.3 é feita uma comparação dos vários produtos competidores mencionados. As colunas da tabela preenchidas com ✗quer dizer que a ferramenta não suporta a funcionalidade ou que essa funcionalidade não é apresentada publicamente.

Nome	Suporte PT	Suporte EN	Redes sociais	Polaridade Geral	Polaridade Entidade-Aspeto	Extração de Entidades	Extração de Aspetos
Semantria	✓	✓	✗	✓	✗	✓	✗
AlchemyAPI	✓	✓	✗	✓	✗	✓	✗
SAS Sentiment Analysis	✓	✓	✓	✓	✓	✓	✓
Clarabridge	✓	✓	✓	✓	✗	✗	✗
Lymbix	✗	✓	✗	✓	✗	✗	✗

Tabela 3.3: Análise comparativa dos diferentes competidores.

Em termos de suporte de língua, todos os produtos fornecem alguma variedade, sendo que todos suportam o Português e o Inglês, à exceção do *Lymbix*. No entanto, o *AlchemyAPI* apenas o Inglês é suportado em algumas funcionalidades, por exemplo na extração de relações. A maior parte dos produtos incorpora a recolha de informação de redes sociais. Quanto à extração da polaridade, todos os produtos extraem 3 níveis de polaridade (positivo, negativo ou neutro), à exceção do *Lymbix* e do Clarabridge que extraem mais níveis que especificam a intensidade da opinião.

Quanto à extração de entidades quase todos os competidores possuem essa funcionalidade. Já na extração de aspetos, embora todas as ferramentas extraiam sintagmas nominais ou palavras-chave estas não são consideradas aspetos, como tal apenas a ferramenta *SAS Sentiment Analysis* possui essa funcionalidade.

O *SAS Sentiment Analysis* é o único produto que oferece a possibilidade de melhoramento dos seus modelos de análise através do *feedback*, personalizando o produto de acordo com as intenções do cliente. Outras características únicas entre os produtos mencionados são a extração de intenções e a análise de voz, possíveis no *Salience Engine* e *Clarabridge*, respetivamente.

Apesar de todos quase todos os sistemas suportarem as duas línguas que este trabalho analisa, apenas uma ferramenta apresenta funcionalidades semelhantes às desenvolvidas neste trabalho. No entanto, todos os sistemas aqui apresentados não são soluções ao que foi proposto neste trabalho, uma vez que implicaria custos para a Wizdee. Uma outra razão é o facto de que ao usar ferramentas externas, a Wizdee perderia o controlo da ferramenta, uma vez que fica sem controlo do código-fonte.

Capítulo 4

Abordagem

O crescente interesse das redes sociais trouxe à *Wizdee* a oportunidade de explorar a extração de conhecimento a partir do texto das redes sociais. A *Wizdee* já possui diversas ferramentas que permitem o processamento básico de texto, que foram as ferramentas base para o desenvolvimento do trabalho, como por exemplo, o identificador de classes gramaticais, frases e átomos.

Mesmo sendo este um trabalho com uma forte componente de investigação, este também possui uma componente de engenharia que tem como objetivo principal a criação de módulos utilizáveis pela *Wizdee* em projetos reais. Nesta secção são descritos os requisitos funcionais e não funcionais (secção 4.1) que este projeto teve que satisfazer. Os requisitos funcionais são, essencialmente, a criação de conetores de ligação entre o sistema e as redes sociais, bem como a extração de conhecimento de opiniões em forma de texto. Na secção 4.2 é apresentada a arquitetura da plataforma e na secção 4.3 são descritos os componentes relevantes para o trabalho desenvolvido. O processo de desenvolvimento de ferramentas capazes de extrair este tipo de conhecimento é muitas vezes dispendioso e acarreta diferentes riscos que são descritos na secção 4.4.

4.1 Análise de Requisitos

Nesta secção são apresentados os requisitos funcionais (secção 4.1.1) e não funcionais (secção 4.1.2) do sistema, de forma a permitir uma melhor compreensão dos objetivos deste trabalho.

4.1.1 Requisitos Funcionais

Nesta secção são descritos os requisitos funcionais de alto nível, que ajudam a perceber os objetivos do trabalho desenvolvido. Os dois primeiros requisitos são conetores para duas das principais redes sociais, o *Facebook* e o *Twitter*. Estes dois conetores devem permitir extrair informação relevante para análise a feita posteriormente. O último consiste na extração de opiniões, nomeadamente a extração da informação presente nas opiniões, representando-as com os quíntuplos. A Tabela 4.1 especifica cada um desses requisitos.

RF.01 - Conetor para o *Facebook*

[1]Disponível em `https://www.facebook.com/`
[2]Disponível em `https://twitter.com/`

ID	Nome	Descrição
RF.01	Conetor para o *Facebook*[1]	Implementação de um conetor para o Facebook que permite a extração de informação disponível nesta rede social.
RF.02	Conetor para o *Twitter*[2]	Implementação de um conetor para o *Twitter* que permite a extração de informação disponível nesta rede social.
RF.03	Extração de Opiniões	Implementação de uma ferramenta capaz de extrair sentimentos associados a uma frase.

Tabela 4.1: Requisitos funcionais.

Deve ser implementado um conetor de ligação entre o sistema e a rede social *Facebook*. Deve permitir extrair informações variadas, como as publicações e comentários feitos por utilizadores, de um conjunto pré-definido de páginas. Por cada comentário ou publicação deve ser possível extrair o autor, o número de gostos e partilhas associadas. Para além do texto publicado, deve ser possível extrair as informações públicas de cada utilizador. Na Tabela 4.2 é possível ver os requisitos associados.

ID	Nome	Descrição
RF.01.01	Extrair publicações de uma determinada página[3]	A partir de um conjunto de páginas pré-definidas, o conetor deve ser capaz de obter publicações e comentários bem como todas as informações associadas a estes, como por exemplo o número de gostos.
RF.01.02	Obter perfil público de um determinado utilizador	O conetor deve extrair a informação pública disponível no perfil de um utilizador, nomeadamente o seu nome e género.

Tabela 4.2: Requisitos de RF.01 - Conetor para o *Facebook* .

RF.02 - Conetor para o *Twitter*

Tal como o requisito anterior, o sistema deve permitir uma ligação ao *Twitter* de forma a extrair conteúdo dessa rede social. Deve ser possível extrair *tweets* que contenham menções pré-determinadas (como por exemplo, *@ronaldo*), *hashtags* (por exemplo, *#oscars*) ou expressões (como por exemplo, *Apple*). Para além da extração de *tweets* deve ser possível extrair as informações permitidas pelos utilizadores, os autores das publicações. Por fim, para cada *tweet* extraído deve ser possível extrair todos os *retweets* associados e as respostas publicadas. Na Tabela 4.3 estão descritos os requisitos necessários para a construção do conetor para o *Twitter*.

RF.03 - Extração de Opiniões

A extração de opiniões está dividida em dois requisitos que diferem na língua que processam: Língua Portuguesa e Língua Inglesa. A Tabela 4.4 especifica os dois requisitos necessários.

RF.03.01 - Extração de Quíntuplos para a Língua Portuguesa

Esta funcionalidade deve permitir a extração de todos os elementos de um quíntuplo representativo de uma opinião escrita na Língua Portuguesa. Deve extrair, sempre que possível, a data e o autor da opinião. A extração de entidades e aspetos é também uma das funcionalidades necessárias. Sempre que um aspeto não esteja presente na opinião este deve ser representado como sendo o aspeto "*GERAL*". Um outro requisito é a extração da polaridade que classifica a opinião tendo em conta três classes: positiva, negativa e neutra.

[3]Uma página está, geralmente, associada a uma empresa/negócio ou produto, organização ou instituição, local, artista ou figura pública, etc.

ID	Nome	Descrição
RF.02.01	Obter *tweets* com determinadas menções	A partir de um conjunto de menções pré-determinadas, o conetor deve ser capaz de obter os *tweets* associados a essas menções e as respostas a esses *tweets*, denominados *retweets*.
RF.02.02	Obter *tweets* com determinadas *hashtags*	A partir de um conjunto de *hashtags* pré-determinadas, o conetor deve ser capaz de obter os *tweets* associados a essas *hashtags* e as respostas a esses *tweets*.
RF.02.03	Obter *tweets* que contenham determinadas expressões	A partir de um conjunto de expressões pré-determinadas, o conetor deve ser capaz de extrair *tweets* que contenham essas mesmas expressões.
RF.02.04	Obter informações sobre um determinado utilizador	O conetor deve extrair a informação pública associada ao perfil de um utilizador, como por exemplo o seu nome.

Tabela 4.3: Requisitos de RF.02 - Conetor para o *Twitter* .

ID	Nome	Descrição
RF.03.01	Extração de Quíntuplos para a Língua Portuguesa	A ferramenta deve ser capaz de construir quíntuplos representativos de uma opinião expressa na Língua Portuguesa.
RF.03.02	Extração de Quíntuplos para a Língua Inglesa	A ferramenta deve ser capaz de construir quíntuplos representativos de uma opinião expressa na Língua Inglesa.

Tabela 4.4: Requisitos de RF.03 - Extração de opiniões.

Por exemplo, da frase "*(20/1/2015 Sara) A luz do meu portátil é fraca.*" deve ser extraída o seguinte quíntuplo: (Autor = "*Sara*", Data = "*20/1/2015*", Entidade = "*portátil*", Aspeto = "*luz*", Polaridade = "*negativa*")

Na Tabela 4.5 estão descritos os requisitos necessários.

ID	Nome	Descrição
RF.03.01.a	Extração do autor da opinião	O sistema deve ser capaz de extrair o autor detentor da opinião.
RF.03.01.b	Extração da data em que a opinião foi transmitida	O sistema deve extrair a data em que a opinião foi publicada.
RF.03.01.c	Extração da entidade a que a opinião se refere	O sistema deve ser capaz de extrair a entidade de uma opinião de forma automática.
RF.03.01.d	Extração do aspeto a que a opinião se refere	O sistema deve ser capaz de extrair, de forma automática o aspeto a que a opinião se refere.
RF.03.01.e	Extração da polaridade da opinião	Criação de sistema capaz de classificar uma opinião de acordo com a sua polaridade, neste caso, positiva, negativa ou neutra.

Tabela 4.5: Requisitos de RF.03.01 - Extração de Quíntuplos para a Língua Portuguesa.

RF.03.02 - Extração de Quíntuplos para a Língua Inglesa

Tal como o requisito anterior, esta funcionalidade deve representar de uma forma estruturada uma opinião expressa na Língua Inglesa, usando os quíntuplos. Deve extrair a data e o autor sempre que essa informação esteja disponível. A extração de entidades, aspetos e polaridade são outras das funcionalidades a desenvolver. Na Tabela 4.6 são especificados os requisitos a desenvolver.

4.1.2 Requisitos Tecnológicos

A Tabela 4.7 apresenta os requisitos tecnológicos exigidos neste trabalho.

ID	Nome	Descrição
RF.03.02.a	Extração do autor da opinião	O sistema deve ser capaz de extrair o autor detentor da opinião.
RF.03.02.b	Extração da data em que a opinião foi transmitida	Deve extrair a data em que a opinião foi publicada.
RF.03.02.c	Extração da entidade a que a opinião se refere	O sistema deve ser capaz de extrair a entidade de uma opinião de forma automática.
RF.03.02.d	Extração do aspeto a que a opinião se refere	O sistema deve ser capaz de extrair, de forma automática o aspeto a que a opinião se refere.
RF.03.02.e	Extração da polaridade da opinião	Desenvolvimento de um sistema capaz de classificar uma opinião de acordo com a sua polaridade, neste caso, positiva, negativa ou neutra.

Tabela 4.6: Requisitos de RF.03.02 - Extração de Quíntuplos para a Língua Inglesa.

ID	Nome	Descrição
RT.01	Linguagens de Programação	Toda a programação desenvolvida deve ser feita usando o Java, Python ou C. No entanto, as funcionalidades devem ser acedidas usando apenas a linguagem Java, uma vez que a plataforma *Wizdee* está, maioritariamente, desenvolvida em Java.
RT.02	Licença	Todas as ferramentas, bibliotecas, recursos usados devem ser permitir o uso comercial sem encargos. Resumidamente, as licenças permitidas são: Licença *BSD*, *LGPL*, Licença *Apache*, Licença *MIT* e *CPL*.

Tabela 4.7: Especificação dos Requisitos Tecnológicos.

4.2 Arquitetura

Nesta secção é feita uma descrição da arquitetura da plataforma *Wizdee*. A arquitetura (ver Figura 4.1) está dividida em três camadas: camada de dados, camada de negócio[4] e camada de apresentação[5]. A camada de dados preocupa-se com a persistência dos dados. A camada de negócio contém a lógica da plataforma e pode ser dividida em quatro segmentos: *Engines*, *Handlers*, *Managers* e *Libraries*. Por fim, a camada de apresentação permite a interação entre o utilizador e o sistema através de uma aplicação web.

4.2.1 Camada de Dados

A camada de dados permite guardar os dados de forma persistente, e é também responsável pela otimização de determinadas tarefas, como a pesquisa e a relação entre dados.

Base de Dados

Este segmento ocupa-se da persistência dos dados necessários para o sistema. A plataforma usa uma base de dados, mais concretamente o *PostgreSQL*[6].

Índice de Pesquisa

O Índice de Pesquisa é um segmento importante, uma vez que permite que a tarefa de pesquisa de texto seja mais fácil e mais eficiente. Atualmente, a *Wizdee* usa a ferramenta

[4]Em inglês, *Business Layer*.
[5]Em inglês, *Presentation Layer*.
[6]Disponível em `http://www.postgresql.org/`

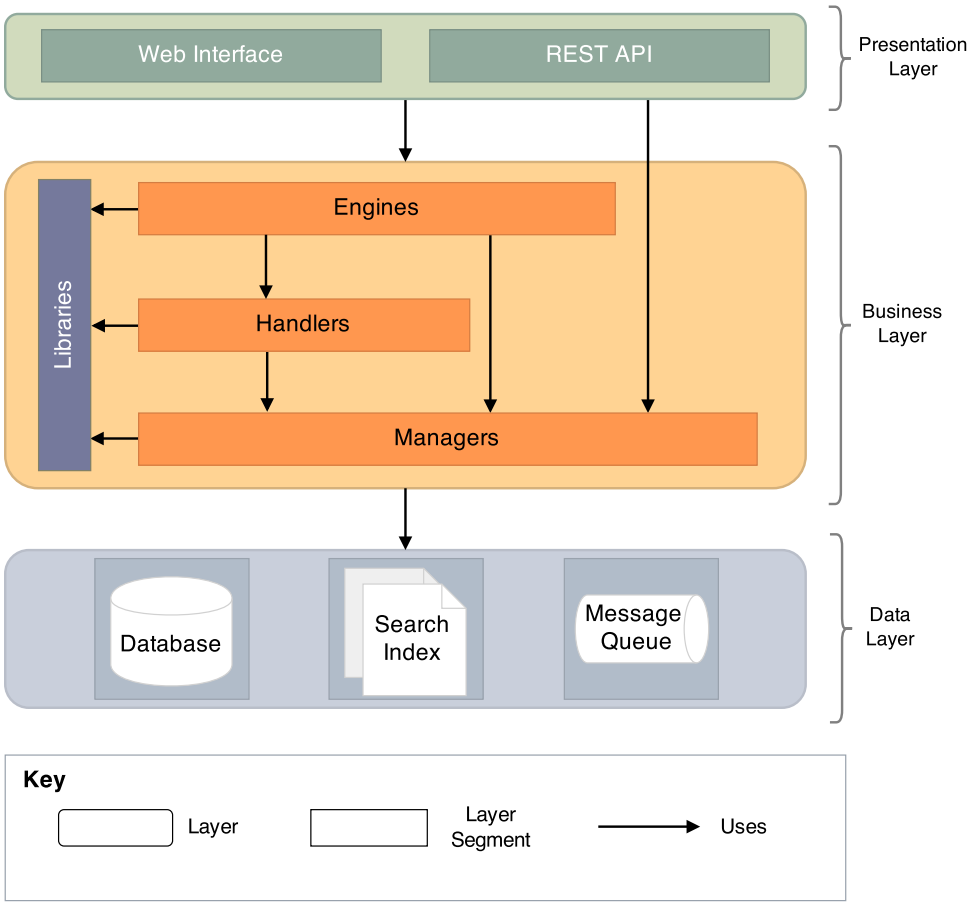

Figura 4.1: Arquitetura do sistema, cortesia da Wizdee.

Solr[7], que é implementada sobre o motor de pesquisa textual *Apache Lucene*[8], para a indexação dos dados e pesquisa.

Fila de Mensagens

A Fila de Mensagens permite a criação e acesso a mensagens de forma assíncrona, ou seja as mensagens são criadas e guardadas em fila até serem consumidas para, posteriormente, serem executadas. Guarda as mensagens de forma persistente garantindo que não existem perdas de mensagens sem que estas sejam consumidas, mesmo que o sistema fique indisponível.

Atualmente, a fila de mensagens é usada, essencialmente, para a sincronização de fontes de dados externas, sendo usada a ferramenta *Apache ActiveMQ*[9].

4.2.2 Camada de Negócio

É na camada de negócio que toda a lógica da plataforma está implementada. Esta camada está dividida em quatro segmentos que interagem uns com os outros, que são descritos de seguida.

[7]Disponível em `http://lucene.apache.org/solr/`
[8]Disponível em `http://lucene.apache.org/`
[9]Disponível em `http://activemq.apache.org/`

Managers

Os *Managers* permitem a interação com a camada de dados e têm como principal objetivo disponibilizar o acesso aos dados de uma forma abstrata, conseguindo assim, transparência para os restantes segmentos. Como tal, os *Managers* executam as operações básicas sobre os dados, conhecidas como *CRUD*[10].

Handlers

Os *Handlers* tem como objetivo processar a informação de entrada e determinar a resposta adequada a esse pedido. Na prática cada *Handler* anota parte do pedido de uma forma específica, tendo em conta o seu propósito. Por exemplo, se for uma pesquisa, o *Handler* tenta criar os cenários[11] que interpretam essa pesquisa.

Engines

Os *Engines* coordenam a geração de resposta aos pedidos vindos da camada de apresentação. São responsáveis pela seleção da resposta a fornecer à camada de apresentação, que é feita com base num mecanismo de prioridade, a natureza de cada *Handler* e a confiança que cada *Handler*. Os *Engines* são também responsáveis pela gestão de sessões e pelo *logging* das mesmas.

Bibliotecas

As Bibliotecas oferecem ao sistema um conjunto de ferramentas para o processamento do texto. Inclui as bibliotecas que executam tarefas de PLN e de *Text Mining*, como o identificador de classes gramaticais. Estas bibliotecas podem ser acedidas a partir de todos os segmentos da camada de negócio.

4.2.3 Camada de Apresentação

A camada de apresentação possui APIs que permitem o acesso a sistemas externos e interfaces que permitem interação com o utilizador.

Interface Web

A Interface Web é uma aplicação Web que permite a interação dos utilizadores com o sistema. Esta aplicação inclui um interface de *frontend*, que é o interface principal do sistema, e um interface de *backend*, que permite a configuração e gestão do sistema.

REST API

O sistema fornece uma REST[12] API que permite a sua integração com sistemas externos. Atualmente, a API é usada essencialmente para o desenvolvimento de *widgets* e clientes móveis.

[10] Em inglês: *Create*, *Read*, *Update* e *Delete*, ou em Português: Criar, Ler, Atualizar e Apagar.
[11] Os cenários representam as diferentes interpretações que um pedido pode ter no sistema
[12] Representational State Transfer

4.3 Componentes

Nesta secção são apresentados em detalhe alguns componentes da arquitetura que foram necessários durante o desenvolvimento deste trabalho (ver Figura 4.2 - componentes delineados com linha branca). Essencialmente, as novas funcionalidades foram implementadas sobre a componente *Language Library*, já existente na *Wizdee* e sobre as componentes *Twitter* e *Facebook Connector* que foram criadas.

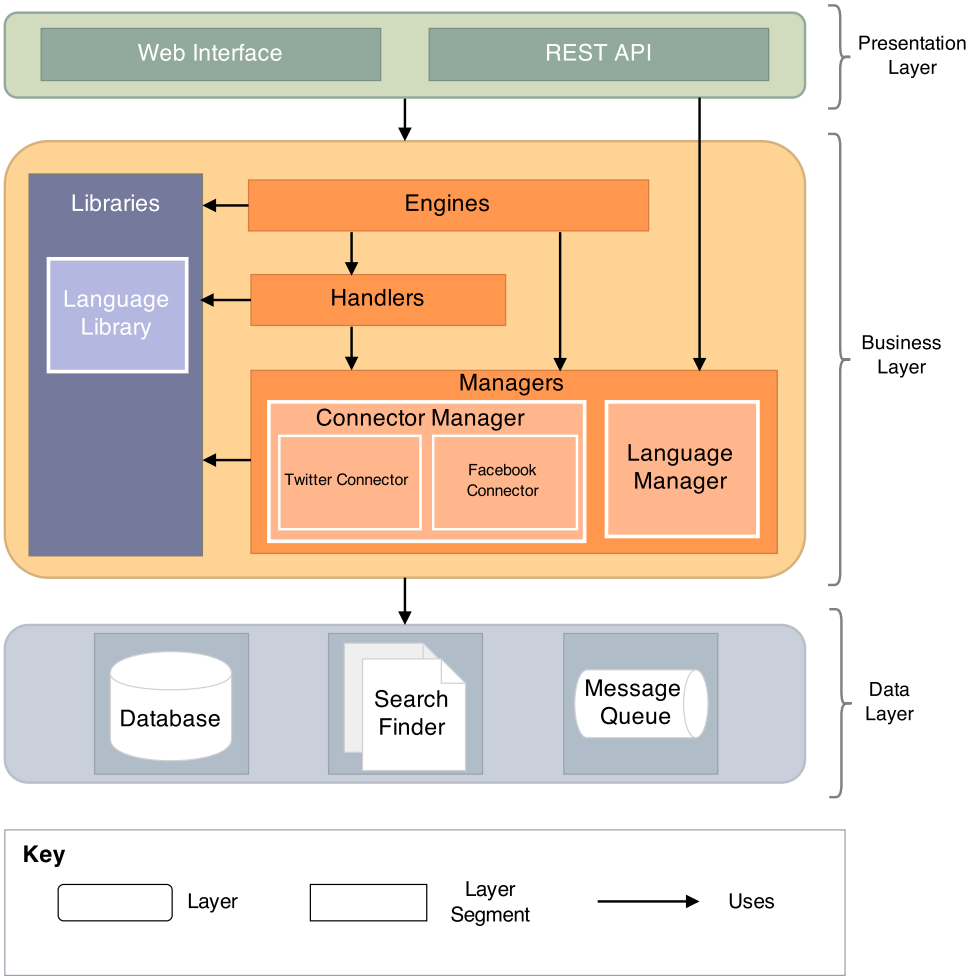

Figura 4.2: Componentes necessários para o projeto.

4.3.1 Componentes Relevantes

Nesta secção são descritos os componentes mais relevantes no âmbito deste projeto. Estes componentes já estão implementados na *Wizdee* e foram usados neste trabalho.

Language Manager

Language Manager permite centralizar as funcionalidades do sistema que são específicas à língua. Basicamente, tem uma funcionalidade de API para o acesso de ferramentas de

Processamento de Linguagem Natural (PLN) implementadas na *Language Library* (ver secção 4.3.2).

Connector Manager

Connector Manager centraliza as funcionalidades associadas aos conetores. Um conetor é um componente que se especializa na ligação e extração de dados de um sistema externo, como o *Facebook* ou o *Twitter*.

4.3.2 Componentes Modificados

De seguida, são descritos os componentes já implementados pela *Wizdee* que foram modificados durante o desenvolvimento, de forma a incluir as novas funcionalidades.

Language Library

Language Library é um componente que contém a implementação de diversas ferramentas de PLN e *Text Mining*, como por exemplo, lematizador, extração de tópicos, remoção de *stop words*, identificação de frases, átomos, classes gramaticais, entre outras.

O componente está dividido em três bibliotecas: *Language Commons Library*, *PT Language Library* e *EN Language Library*. A biblioteca abstrata *Language Commons Library*, contém as ferramentas que são independentes da língua, as classes abstratas e *interfaces* que são usadas pelas outras duas bibliotecas específicas da língua. *PT Language Library* e *EN Language Library* implementam as ferramentas que são específicas da língua, neste caso o Português e o Inglês respetivamente, como por exemplo, a transformação para número singular, o corretor de erros, o identificador de classe gramatical, etc. As alterações que foram feitas nesta componente passaram pela adição de métodos de identificação de polaridade, entidades, aspetos e quíntuplos.

4.3.3 Componentes Desenvolvidos

Por fim, nesta secção, são apresentados os componentes importantes para a satisfação de certos requisitos que ainda não existiam e que foram implementados na Wizdee.

Twitter Connector

O *Twitter Connector* tem como principal objetivo a ligação entre o *Twitter* e a plataforma *Wizdee*. Este conector é, essencialmente, composto por três funcionalidades. A primeira, é a extração de textos da rede social. Para tal, foi implementado uma classe que cria a ligação com o *Twitter* e implementa os métodos para extração de cada tipo de informação, como por exemplo, informações sobre o utilizador. A segunda funcionalidade passa pelo processamento dos textos extraídos, ou seja, foram implementados métodos que permitem processar e analisar cada *tweet*, usando as bibliotecas existentes no componente *Language Library*. Por fim, a última funcionalidade permite guardar todos os dados extraídos e processados. Para tal foram criados comandos que permitem adicionar, modificar ou remover dados da base de dados.

Facebook Connector

Tal como o componente anterior, o *Facebook Connector* deve permitir a ligação ao *Facebook*. É, também composto pelas mesmas três funcionalidades: a ligação e extração de textos

do Facebook, o processamento e análise dos mesmos e por fim, deve permitir gravar esses dados numa base de dados.

4.4 Análise de Riscos

Nesta secção são identificados os riscos associados ao trabalho desenvolvido e os seus planos de contingência previstos.

Tempo gasto na criação de corpus de treino

O corpus de treino é um elemento importante, uma vez que influencia a performance do sistema. Embora para a Língua Inglesa já existam alguns corpus disponíveis, isso já não acontece para a Língua Portuguesa, e por isso foi necessário a criação de um corpus de treino. Naturalmente, a criação de corpus de forma manual pode ocupar demasiado tempo e atrasar outras tarefas dependentes destas, por exemplo a implementação, treino e testes do algoritmo.

Plano de Contingência Uma das soluções é o uso de aplicações *crowdsourcing*[13], como por exemplo o *Mechanical Turk*[14], para a criação de corpus de treino. No entanto, esta solução pode gerar um corpus de treino com menor qualidade e necessita de algum investimento monetário. Uma outra solução passa por usar técnicas de aprendizagem semi-supervisionada que assume o uso de corpus não classificados.

Qualidade dos corpus de treino

Os corpora de treino já disponíveis, principalmente na Língua Inglesa, podem vir a mostrar-se corpus de pouca qualidade, o que pode se transmitir num forte impacto na performance dos modelos. A má qualidade dos corpora passa pela inconsistências das classificações ou também pela fraca distribuição de instâncias em todas as classes.

Plano de Contingência Uma solução é identificar novos corpora de treino disponíveis, o que é difícil uma vez que, muitas vezes, a licença não permite o seu uso em contexto comercial. No caso de má distribuição das instâncias, ou seja o corpus ter poucos casos de uma determinada classe, a junção de diferentes corpus é também uma solução, no entanto pode proporcionar inconsistências. Em último caso, a solução é a criação de um corpus de forma manual.

Qualidade e Performance das ferramentas e recursos

Este trabalho propõe o uso de algumas ferramentas já implementadas pela *Wizdee*, como por exemplo o identificador de classe gramatical, o lematizador, etc. A qualidade dessas ferramentas pode influenciar os resultados dos modelos criados. Essas ferramentas também podem revelar-se demasiado lentas o que pode afetar o tempo de criação de características.

Plano de Contingência Uma das maneiras de resolver a questão é o uso de outras ferramentas ou recursos que tenham o mesmo propósito. No caso de não existir ou não poderem ser usadas, outras soluções passam por tentar melhorar as ferramentas já usadas,

[13]Processo que permite obter serviços, solicitando a contribuição de um grupo extenso de pessoas existentes numa comunidade on-line.

[14]Disponível em `https://www.mturk.com/mturk/welcome`

remover o uso destas no caso de não serem essenciais, ou, em último caso, implementar novas ferramentas. Esta última pode resultar em atrasos excessivos na implementação do resto do sistema.

Tempo gasto na implementação de algoritmos

Alguns algoritmos podem se tornar mais complexos de implementar do que o esperado, o que resulta no atraso de todo o desenvolvimento do projeto.

Plano de Contingência Uma das soluções é tentar usar ferramentas que já possuam a implementação desse algoritmo, ou então escolher um outro algoritmo menos complexo de implementar.

Tempo gasto no treino e nos testes dos algoritmos

O treino do modelo pode se tornar uma tarefa bastante lenta devido a vários factores, como por exemplo, a velocidade de processamento das ferramentas de Processamento de Linguagem Natural (PLN) ou as limitações de hardware. Se o treino ocupar tempo excessivo, a fase de testes será ainda mais demorada, uma vez que é nessa fase que são ajustados os parâmetros do modelo, ou seja são feitos vários treinos com diferentes parâmetros que influenciam a performance do modelo.

Plano de Contingência Uma solução é arranjar disponibilidade de, pelo menos, uma máquina para a tarefa de treino e testes, ou então encontrar soluções como a *Amazon Elastic Compute Cloud*[15].

4.5 Especificação de Testes

Testes serão feitos com o intuito de avaliar os modelos criados para a extração de polaridade, entidades e aspetos. Serão feitos dois tipos de testes: Funcionais e de Performance.

Serão feitos testes funcionais com o objetivo de validar todos os modelos criados, percebendo qual a sua performance. Os testes unitários serão feitos usando a ferramenta de testes *JUnit*[16].

Também serão feitos testes não funcionais, mais exatamente testes de performance, onde serão avaliados o tempo de extração de características e o tempo de criação do modelo.

[15]Disponível em `http://aws.amazon.com/ec2/`
[16]Disponível em `http://junit.org/`

Capítulo 5

Metodologia e Planeamento

Nesta secção será apresentada a metodologia (secção 5.1) seguida neste trabalho e o planeamento seguido em ambos os semestres.

5.1 Metodologia

O desenvolvimento deste trabalho segue uma metodologia baseada no *Scrum*. *Scrum* (Schwaber, 2004) é uma metodologia de desenvolvimento de software que é iterativa e incremental. As metodologias tradicionais, como a metodologia *Waterfall*, assumem que os requisitos são inteiramente conhecidos antes do começo do projeto. No entanto, muitas das vezes, essa assunção não corresponde à realidade. O principio básico do *Scrum* é reconhecer que os requisitos podem mudar durante o decorrer do projeto e que alterações inesperadas podem surgir. Aceita que o problema inicial não pode ser inteiramente definido no início do projeto, e por isso permite a realização de avaliações e ajustamentos durante o desenvolvimento do projeto.

No *Scrum* existem, essencialmente, três papeis: proprietário do produto que mantém a ligação entre a equipa de desenvolvimento e os clientes, o *Scrum Master* que é responsável por manter a equipa focada, e a equipa de desenvolvimento que é o grupo responsável pelo desenvolvimento do produto (Schwaber, 2004). Inicialmente, todas as tarefas a desenvolver são colocadas no *Sprint Backlog*, que em cada Sprint (corresponde a uma iteração no processo de desenvolvimento) são priorizadas através de uma reunião de planeamento do *Sprint*.

A *Wizdee* já implementa uma metodologia baseada no *Scrum*. Neste projeto em particular, os papeis dividem-se da seguinte forma: Prof. Paulo Gomes é o proprietário do produto, o Eng. Bruno Antunes é o *Scrum Master* e a equipa de desenvolvimento é a Sara Pinto. Os *Sprints* na *Wizdee* duram cerca de duas semanas e começam, geralmente, com uma reunião de planeamento do *Sprint*. Todas as semanas é feita uma reunião onde se regista o progresso do *Sprint*. No fim de cada *Sprint* é revisto todo o trabalho desenvolvido. Na *Wizdee* o controlo dos *Sprints* é feito usando a ferramenta *JIRA Agile*[1].

Embora a metodologia *Scrum* geralmente não siga um planeamento gerido por um Diagrama de *Gantt*, foi criado um de forma a identificar o trabalho desenvolvidos em ambos os semestres.

Na Figura 5.1 é apresentado o trabalho feito na primeira etapa do projeto, que corresponde ao primeiro semestre. O início do primeiro semestre foi focado essencialmente na definição do projeto, na pesquisa sobre as áreas relevantes e a investigação de trabalhos na área. Ainda durante este semestre foram realizadas algumas tarefas de desenvolvimento,

[1]Disponível em `https://www.atlassian.com/software/jira/agile`

distribuídas por quatro *Sprints*, que permitiram a implementação de uma abordagem inicial e experimental da ferramenta de extração de polaridade para o inglês.

Na Figura 5.2 é apresentado o trabalho feito no segundo semestre que está dividido em nove *Sprints*. Estes *Sprints* foram utilizados para o desenvolvimento das ferramentas de extração de polaridade, entidades e aspetos para ambas as línguas requeridas.

5.2 *Sprints* Realizados

De seguida são descritos as funcionalidades desenvolvidas em cada um dos *sprints* realizados.

- **Sprint #1 (20/10/2014 a 31/10/2014)**:
  Ferramenta de Extração de Polaridade para o Inglês
  - Pesquisa de Corpus de Treino e Teste
  - Estudo dos Recursos e Ferramentas disponíveis para o Inglês
  - Construção do 1º conjunto de características
- **Sprint #2 (3/11/2014 a 14/11/2014)**:
  Ferramenta de Extração de Polaridade para o Inglês
  - Construção dos *word embeddings* simples
  - Construção dos *word embeddings* com polaridade
  - Testes com as Redes Neuronais Convolucionais
- **Sprint #3 (17/11/2014 a 28/11/2014)**:
  Ferramenta de Extração de Polaridade para o Inglês
  - Melhorar os *word embeddings* com polaridade
  - Construção da 2º versão do conjunto de características
- **Sprint #4 (1/12/2014 a 12/12/2014)**:
  Ferramenta de Extração de Polaridade para o Inglês
  - Testes com os *word embeddings* de polaridade
  - Testes com as Redes Neuronais Convolucionais
- **Sprint #5 (9/2/2014 a 20/2/2014)**:
  Ferramenta de Extração de Polaridade para o Inglês
  - Experiências com os *word embeddings* de polaridade
  - Testes com as Máquinas de Vector de Suporte
- **Sprint #6 (23/2/2014 a 6/3/2014)**:
  Ferramenta de Extração de Polaridade para o Inglês
  - Construção do léxico automático de polaridade
  - Construção da 3º versão do conjunto de características
  - Análise e Melhorias do modelo de classificação
- **Sprint #7 (9/3/2014 a 20/3/2014)**:
  Ferramenta de Extração de Polaridade para o Inglês

  - Finalização e Integração da ferramenta

  Implementação do Conector para o *Facebook*

  - Levantamento das ferramentas disponíveis
  - Planeamento e Desenvolvimento da estrutura de dados

- **Sprint #8 (23/3/2014 a 3/4/2014)**:
  Implementação do Conector para o *Facebook*

  - Finalização e Integração

  Implementação do Conector para o *Twitter*

  - Levantamento das ferramentas disponíveis
  - Planeamento e Desenvolvimento da estrutura de dados
  - Finalização e Integração

  Ferramenta de Extração de Aspetos

  - Levantamento de corpus existentes
  - Estudo das diferentes abordagens

- **Sprint #9 (6/4/2014 a 17/4/2014)**:
  Ferramenta de Extração de Aspetos para o Inglês

  - Finalização e Integração da ferramenta

  Ferramenta de Extração de Entidades para o Inglês

  - Levantamento de corpus existentes
  - Implementação da abordagem linguística

- **Sprint #10 (20/4/2014 a 1/5/2014)**:
  Ferramenta de Extração de Entidades para o Inglês

  - Finalização e Integração da ferramenta

  Ferramenta de Extração de Quíntuplos para o Inglês

  - Extração de relações Entidade-Aspeto
  - Extração do Texto Relevante
  - Finalização e Integração da ferramenta

- **Sprint #11 (4/5/2014 a 15/5/2014)**:
  Parser de Dependências para o Português

  - Pesquisa de corpus de treino e teste disponíveis
  - Implementação e Treino do modelo
  - Testes e Integração

  Ferramenta de Extração de Aspetos para o Português

  - Desenvolvimento e Integração

- **Sprint #12 (18/5/2014 a 29/5/2014)**:
  Ferramenta de Extração de Entidades para o Português
  - Desenvolvimento e Integração

  Ferramenta de Extração de Quíntuplos para o Português
  - Desenvolvimento e Integração

  Ferramenta de Extração de Polaridade para o Português
  - Extração de características
- **Sprint #13 (1/6/2014 a 12/6/2014)**:
  Ferramenta de Extração de Polaridade para o Português
  - Melhoramento das características
  - Testes e Melhoramentos ao modelo de classificação

	Name	Duration
1	Análise de Requisitos	15 days
2	Estado da arte	30 days
3	Análise da Competição	5 days
4	Abordagem Proposta	15 days
5	Implementação	15 days
6	Relatório Intermédio	17 days
7	**Desenvolvimento Fase I**	**40 days**
8	Sprint #1	10 days
9	Sprint #2	10 days
10	Sprint #3	10 days
11	Sprint #4	10 days

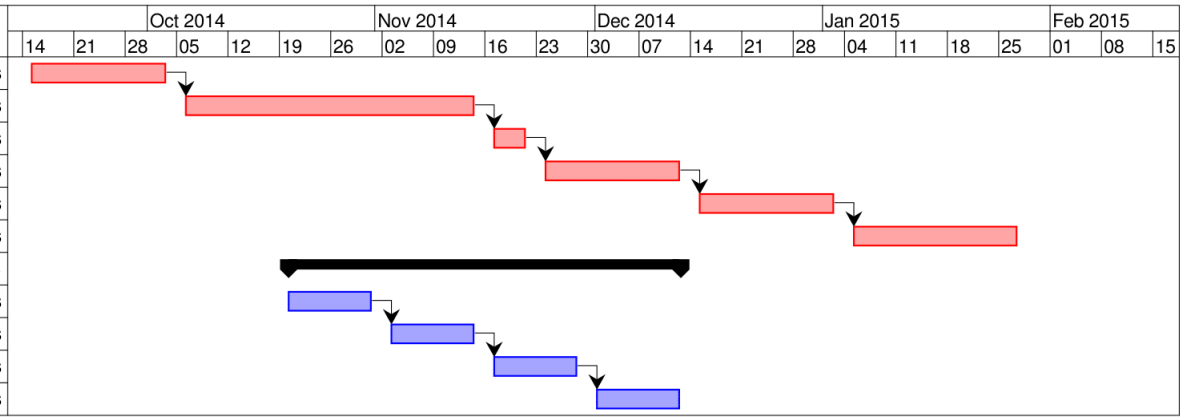

Figura 5.1: Diagrama de *Gantt* para o primeiro semestre.

	Name	Duration
1	**Desenvolvimento Fase II**	**90 days**
2	Sprint #1	10 days
3	Sprint #2	10 days
4	Sprint #3	10 days
5	Sprint #4	10 days
6	Sprint #5	10 days
7	Sprint #6	10 days
8	Sprint #7	10 days
9	Sprint #8	10 days
10	Sprint #9	10 days
11	Relatório Final	16 days

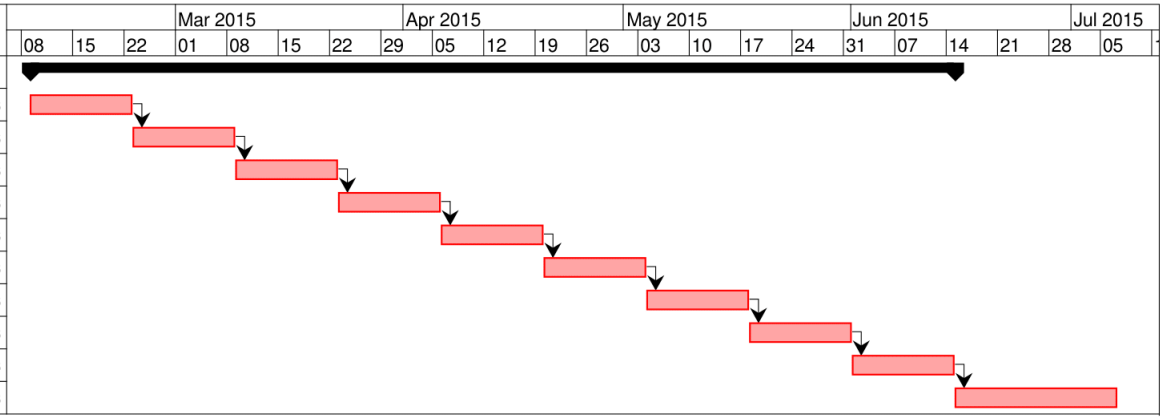

Figura 5.2: Diagrama de *Gantt* para o segundo semestre.

Capítulo 6

Implementação

Tal como descrito anteriormente, este trabalho tem como objetivo o desenvolvimento de diversas ferramentas que permitam, para além de extrair informações das redes sociais, fazer análises a opiniões escritas tanto na Língua Inglesa como na Língua Portuguesa. Essa análise permite extrair a polaridade da opinião de uma forma geral, bem como os quíntuplos associados a essa opinião (informações sobre data, autor, entidade, aspeto e polaridade).

Todas as ferramentas planeadas foram desenvolvidas. No entanto, no decorrer deste projeto, foi necessário criar novos recursos e ferramentas que não estavam planeadas no início do projeto.

De seguida, são apresentados detalhes sobre a implementação e abordagem seguida em cada uma das ferramentas implementadas.

6.1 Conector do *Facebook*

O conector do *Facebook* tem como objetivo extrair da rede social *Facebook* todas as informações públicas, acessíveis através de uma API, e guardar numa base de dados todo o conteúdo recolhido. Uma vez que o *Facebook* não disponibiliza oficialmente uma implementação cliente da sua API para Java, foi usada uma implementação existente, a *RestFB*[1], que permite a integração do nosso sistema com o *Facebook* usando a linguagem Java.

6.1.1 Extração de Informação

Os dados no Facebook estão associados a páginas ou a perfis, por isso para extrair informação é necessário saber quais são as páginas que se pretendem analisar. Essas páginas são obtidas a partir de um utilizador que as administra. Como tal, o ponto central da extração de informação são as páginas do *Facebook* que um determinado utilizador autenticado[2] administra.

Como se pode ver na Figura 6.1, a partir das páginas extraídas são recolhidas todas as outras informações como o *Feed*, os *Insights*, as mensagens e os eventos. entre outras. Nos eventos são extraídas informações sobre as respostas aos convites feitos e ainda o *feed* de publicações. Na tabela 6.1 descrevem-se em mais detalhe todos os dados que se consegue extrair.

[1] Mais informações em `http://restfb.com/`

[2] O utilizador que se autentica no Facebook e autoriza a nossa aplicação a aceder aos seus dados privados.

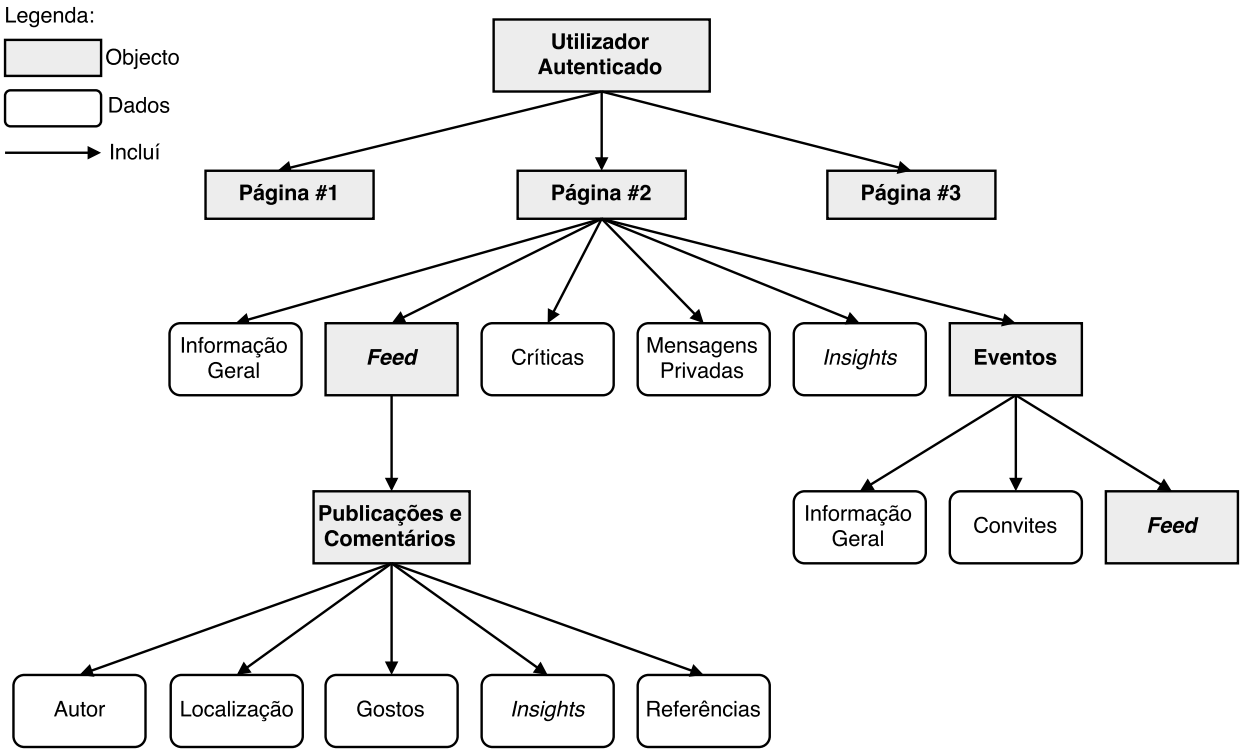

Figura 6.1: Estrutura dos dados extraídos do *Facebook*.

Uma das maiores limitações que se encontra quando se construi um conector para o *Facebook* é a quantidade de restrições que este impõe. Como forma de limitar o acesso a informação privada, o *Facebook* obriga a que cada aplicação explicite, ao utilizador, que informação pretende extrair. No caso do nosso sistema, são necessárias as seguintes permissões:

- **Permissão *Manage_Pages*** Permite extrair o código de acesso das páginas que um determinado utilizador administra.
- **Permissão *Read_Page_Mailboxes*** Permite extrair as mensagens privadas de uma página. Não permite extrair as mensagens dos administradores das páginas.
- **Permissão *Read_Insights*** Permite extrair as estatísticas de uma página.

Naturalmente, o sistema de permissões que o *Facebook* implementa acrescentou grandes limitações à quantidade e tipo de dados que podemos extrair. Por exemplo, para extrair determinadas informações (como, por exemplo, a data de nascimento) de um utilizador que comentou na nossa página, esse mesmo utilizador teria que autorizar a nossa aplicação a recolher esses dados. Por isso, em relação a utilizadores exteriores e desconhecidos apenas a informação pública é recolhida.

[3]Uma página no Facebook, geralmente está associada a uma empresa ou organização, marca ou produto, figuras públicas, artistas ou comunidades.

[4]Os *insights* correspondem às estatísticas de utilização de uma página ou da popularidade da publicação.

[5]As referências ou tags podem ser feitas numa publicação, a pessoas, páginas e eventos.

Tipo de Objecto	Informação	Descrição
Páginas[3]	Informação da página	Informações sobre a categoria a que pertence, data de fundação, nome, descrição, telefone, número de Gostos, url para website, localização, entre outras.
	Insights[4]	Informações temporais sobre o número de fãs por país, o número de utilizadores a falarem sobre a página por país, entre outras.
	Feed	Listagem de todos as publicações e comentários publicados na página.
	Eventos	Listagem de todos os eventos criados e administrados pela página.
	Críticas	Listagem de criticas e avaliações feitas pelas utilizadores.
	Mensagens Privadas	Mensagens privadas enviadas à página, incluindo o autor da mensagem e a data de envio.
Publicações e Comentários	Autor	Informações sobre autor da publicação/comentário, como por exemplo, o primeiro e último nome e o url para o perfil do utilizador.
	Localização	Informação geográfica sobre o local onde a publicação foi feita. Inclui informações como o país, a cidade, latitude e longitude, rua, código postal, entre outras.
	Gostos	Listagem de todos os utilizadores que fizeram Gosto à publicação ou comentário.
	Insights	Informações estatísticas sobre a publicação, como por exemplo o número de utilizadores que viram a publicação, o número de utilizadores que clicaram, o número de ações negativas à publicação, entre outras. Apenas está disponível para publicações feitas pela página.
	Referências[5]	Listagem das referências feitas na publicação. São extraídas as informações de acordo com o tipo de referência que pode ser a uma página, a um eventos ou a uma pessoa.
Eventos	Informação do evento	Informações sobre o evento como a sua localização, o nome, a data e descrição.
	Convites	Listagem de utilizadores que foram convidados e as suas respostas. Cada utilizador pode pertencer a um dos seguintes estados: "Sem Resposta", "Aceite", "Não Aceite"ou "Talvez".
	Feed	Listagem de todos as publicações e comentários publicados na página do evento.

Tabela 6.1: Informação detalhada sobre os dados recolhidos pelo conector do *Facebook*.

6.1.2 Atualização de Dados

De forma a não extrair sempre as mesmas publicações do *Facebook* é necessário perceber quais são os novos dados e quais são os dados que já foram obtidos. Uma das soluções básicas seria usar a data da última extração, de forma a recolher apenas as publicações que foram criadas após essa data. No entanto, com esta solução podia-se perder demasiada informação, uma vez que as informações de cada publicação vão alterando ao longo do tempo, como por exemplo, o número de *Gostos* ou o número de comentários associados à publicação. Como tal, a solução encontrada foi calcular o espaço temporal desde a data da publicação até à data do último comentário. Basicamente, na primeira extração para cada publicação esse espaço temporal é calculado e no fim guarda-se o máximo de todas as publicações. Assim, nas próximas extrações, usando a data da última publicação recolhida e o espaço temporal calculado, consegue-se limitar as publicações que queremos. Por exemplo, na primeira extração foi calculado o espaço temporal máximo. Ou seja, sempre que se extraia uma publicação, por exemplo publicada em 1/3/2015, era encontrado o comentário mais recente como resposta a essa publicação, por exemplo de 8/3/2015 (a última resposta à publicação foi em 8 de Março). A diferença temporal é de 7 dias. Para todos as publicações é feito o mesmo cálculo e o máximo de diferença temporal é a tida em conta. Por exemplo, na primeira extração foi calculado um espaço temporal máximo de 10 dias e a data da última publicação recolhida foi dia 12/05/2015. Na segunda extração, sabemos que apenas queremos as publicações com data superior à de 12/5/2015 menos 10

dias, ou seja 2/5/2015.

Com esta solução consegue-se reduzir a extração de informação desnecessária tentado, ao mesmo tempo, reduzir a possibilidade de perder informação relevante. Esta otimização foi aplicada ao *feed* de uma página, ao *feed* de um evento e às mensagens privadas.

6.2 Conector do *Twitter*

Tal como o conector do *Facebook*, o conector do *Twiter* tem como objetivo recolher do *Twitter* todos os dados disponíveis, usando também uma API. O conector deve permitir que através de uma lista de nomes, por exemplo "MEO"ou "M40", seja possível recolher todos os *tweets* que incluam esses mesmos nomes. Esses tweets e as suas informações relacionadas devem ser posteriormente guardados numa base de dados. Uma vez que o *Twitter* não disponibiliza oficialmente uma implementação cliente da sua API para a linguagem Java, foi usada uma implementação existente *Twitter4J*[6], que permite a integração do nosso sistema com o *Twitter*.

6.2.1 Extração de Informação

Como se pode ver na Figura 6.2, a extração de informação pode ser feita partindo de duas vias: extrair *tweets* das listas que um utilizador autenticado administra OU a pesquisa de *tweets* que incluem determinadas palavras-chave.

Através das listas é possível extrair informações sobre os membros[7], subscritores[8] e recolhidos todos os *tweets* publicados dentro da lista. Sempre que se recolhe um *tweet* extraem-se várias informações, incluindo as *hashtags*, menções (referências a utilizadores), *retweets* e respostas ao *tweet* (que por si só são *tweets*), entre outras.

Na Tabela 6.2 está descrito em detalhe todas as informações que o conector extrai do *Twitter*.

Limitações

Uma das limitações encontradas na fase de extração de informação foi a limitação do números de pedidos[9]. Uma vez que esta limitação é imposta pelo próprio *Twitter*, não há uma solução que contorne o problema. No entanto, foi preciso assegurar que esse limite não era ultrapassado, uma vez que a aplicação podia ficar bloqueada deixando assim de poder fazer qualquer tipo de pedido por tempo indeterminado.

Uma vez que o *Twitter* disponibiliza um método que permite perceber se a aplicação está perto de alcançar esse limite, cada vez que o sistema se prepara para fazer um pedido, é primeiro verificado se esse pedido pode ser feito sem ultrapassar o limite. No caso desse pedido ultrapassar o limite o processo de extração termina.

Devido às limitações impostas sobre o número de pedidos, decidiu-se que é mais importante ter os *tweets* mais populares do que todos os *tweets*, por isso na pesquisa foi usado um parâmetro, o *result_ type*, que permite especificar que tipo de resultados esperamos: populares, recentes ou mistos.

[6]Mais informações em `http://twitter4j.org/en/index.html`

[7]Membros de uma lista são adicionados pelo administrador e podem publicar *tweets*.

[8]Subscritores de uma lista apenas podem visualizar o conteúdo publicado na lista. Cada utilizador pode-se auto-subscrever.

[9]Ver em `https://dev.twitter.com/rest/public/rate-limits`

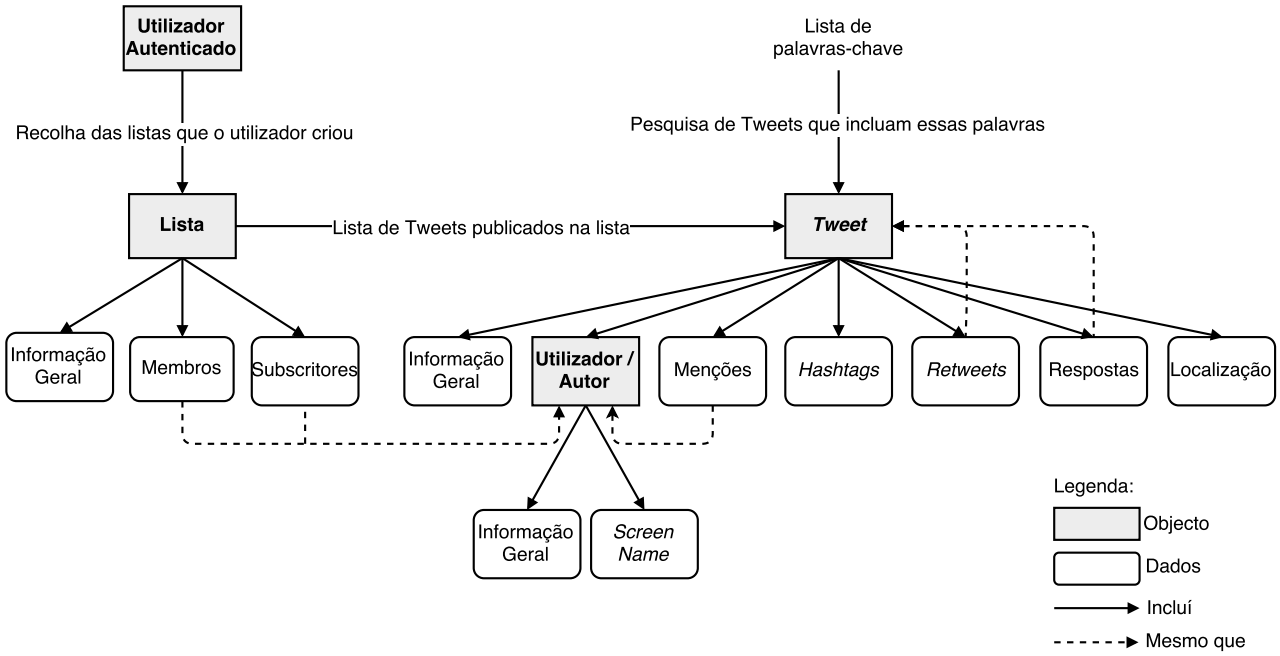

Figura 6.2: Esquema que representa a extração dos dados do *Twitter*.

Atualização de Dados

Como já referido, o *Twitter* tem grandes limites do número de pedidos que se pode fazer, por isso a fase da atualização de dados tornou-se bastante importante de forma a reduzir o pedido de dados repetidos. Na fase da pesquisa, apesar de ser feito um filtro por palavras-chaves, o que já reduz muito o número de resultados possíveis, foi necessário acrescentar um filtro temporal, usando o parâmetro *since_id*. Esse parâmetro é igualado ao id do *tweet* mais recente que temos na base de dados, e assim apenas os *tweets* com id superior, ou seja uma data mais recente que o *tweet* já extraído, são abrangidos.

6.3 Dicionários para a Língua Inglesa

No decorrer do estágio, foram criados diversos dicionários que foram necessários para o desenvolvimento das ferramentas implementadas. Para cada dicionário foi criada uma biblioteca que permite o acesso ao conteúdo do mesmo.

À exceção do léxico de polaridade, a construção dos dicionários passou pelas seguintes fases:

- **Aglomeração de Informação**: Levantamento em vários websites de listas já préfeitas, sendo estas depois aglomeradas num mesmo dicionário.
- **Revisão** De forma a garantir a qualidade dos dicionários, cada um foi revisto e corrigido manualmente. Sempre que necessário foi acrescentada mais informação.

Nesta secção são descritos os diversos dicionários criados para a língua inglesa.

Dicionário de Expressões Idiomáticas

Uma expressão idiomática caracteriza-se essencialmente por uma expressão utilizada vulgarmente numa língua específica (idioma) e cujo significado não é o sentido literal das

Tipo de Objecto	Informação	Descrição
Tweet	Informação geral	Informações sobre o texto, a data de publicação, o número de favoritos, o número de *retweets*, entre outras.
	Autor/Utilizador	Informações sobre o autor da publicação.
	Hashtags	Listagem de todas as *hashtags* utilizadas no *Tweet*.
	Menções	Listagem de todos os utilizadores que são referenciados no *Tweet*.
	Retweets	Listagem de todos os *retweets* cuja origem é o *Tweet* original. A cada um desses *retweets* é feita uma recolha de todas as informação tal como se faz para um *tweet*.
	Respostas	Listagem de todas as respostas ao *tweet* original. Uma vez que uma resposta é também um *tweet*, são também recolhidas todas as informações relativas aos *tweets*.
	Localização	Informações sobre o lugar referenciado pelo *tweet*, como por exemplo, a sua latitude e longitude, o país, o nome a rua, entre outras.
Utilizador	Informação Geral	Informações sobre um utilizador, como por exemplo, a data de criação do perfil, a descrição, o número de seguidores, favoritos e amigos, o número de *tweets* publicados, entre outras.
	Screen Name	O nome que o utilizador detém. Esse nome é usado na identificação ou referência ao utilizador no Twitter.
Lista	Informação Geral	Informações sobre a data de criação da lista, a sua descrição, o nome e o nível de privacidade.
	Membros	Listagem de utilizadores que foram convidados a participar na lista.
	Subscritores	Listagem de utilizadores que se auto-subscreveram na lista e que não podem publicar.

Tabela 6.2: Informação detalhada sobre os dados recolhidos pelo conector do *Twitter*.

palavras que inclui. Por exemplo, em inglês a expressão "*break the ice.*"[10] significa tornar a situação mais confortável quando se está com uma pessoa que ainda não se conhece.

Cada entrada no dicionário de expressões, tem a expressão idiomática, o seu significado e por fim a polaridade associada a essa expressão. O dicionário é composto por cerca de 583 expressões, tal como indicado na tabela 6.3.

Polaridade	Número de Entradas	Exemplos
Positiva	103	"*best of both worlds*" "*be on cloud nine*" "*over the moon*"
Negativa	282	"*in the red*" "*in hot water*" "*costs an arm and a leg*"
Neutra	198	"*sleep on it*" "*a penny for your thoughts*" "*plain as day*"
Total	**583**	

Tabela 6.3: Distribuição das expressões idiomáticas por polaridade.

Dicionário de Calão e Abreviaturas

Tanto os calões[11] como as abreviaturas são muitas vezes encontrados em texto de redes sociais e por isso a construção de um dicionário é muito importante quando se pretende

[10] A tradução literal em português é "quebrar o gelo".

[11] Em inglês, *slang*.

fazer análises a esse tipo de texto. Tal como o dicionário anterior, cada entrada no dicionário tem três aspetos: o calão ou abreviatura, a sua tradução e a polaridade. Na tabela 6.4 estão presentes alguns exemplos de calões e abreviaturas por polaridade.

Polaridade	Número de Entradas	Exemplos
Positiva	63	*gr8* (*great*) *bff* (*best friend forever*) *lol* (*laughing out loud*)
Negativa	81	*adih* (*another day in hell*) *bsod* (*blue screen of death*) *eol* (*end of life*)
Neutra	277	*asap* (*as soon as possible*) *b4* (*before*) *brb* (*be right back*)
Total	**421**	

Tabela 6.4: Distribuição de calões por polaridade e alguns exemplos.

Léxico de Polaridade

Para a ferramenta de extração de polaridade foi criado um léxico de polaridade de forma automática. O conceito base em que a construção deste léxico se baseia é que se uma palavra aparece muitas vezes em contextos negativos é possivelmente uma palavra com um sentimento negativo (Mohammad et al., 2013).

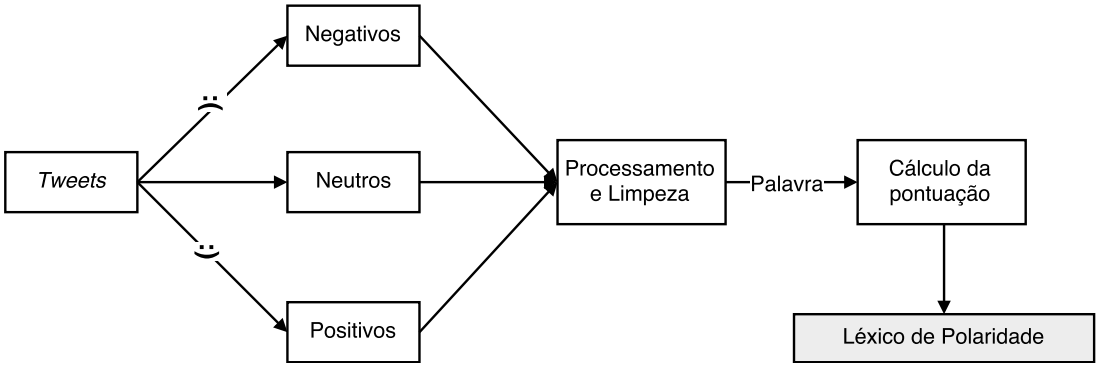

Figura 6.3: Fases de construção do léxico de polaridade.

A construção do léxico, representada pela Figura 6.3, começou pela extração de cerca de 10 milhões de *tweets*, que serviram como base para o léxico. Esses *tweets* foram divididos em três classes: positiva, negativa e neutra. A divisão pelas classes foi também feita de forma automática, e assumiu que sempre que um tweet possui o *smile* "*:)*" é um *tweet* positivo, e sempre que tiver um *smile* "*:(*" é um *tweet* negativo. Um *tweet* que não pertencesse a nenhuma dessas duas classes era processado, e caso não possuísse nenhuma palavra que estivesse presente nos léxicos de polaridade já existentes era considerado neutro. No fim deste processamento obteve-se um corpus com a distribuição descrita na tabela 6.5.

Cada um dos *tweets* obtidos foi processado e todas as palavras consideradas irrelevantes, como as *stop words*, *urls*, números entre outras, foram removidas. Para cada uma das palavras restantes calculou-se uma pontuação que simboliza a ligação entre essa palavra e um determinado sentimento. Essa pontuação baseia-se na frequência que uma palavra tem em *tweets* positivos e negativos. A fórmula usada foi baseada no *PMI* (Liu, 2010), como se pode ver na equação 6.2.

Nome	Total	Positivo	Negativo	Neutro
Dataset Twitter	10.647.936	3.665.713	3.652.822	3.329.401

Tabela 6.5: Estatísticas do corpus construído.

Polaridade Positiva			Polaridade Negativa		
Termo	**Frequência**	**Pontuação**	**Termo**	**Frequência**	**Pontuação**
#TopFollowers	16667	8.2	tymBng	12390	-8
@Harry_Styles	6934	7.8	LouisHow	2191	-7.3
#TopMembers	6262	7.7	@JalenMcMillan	2347	-7.3
#teamdreway	1613	7.2	PeshawarAttack	1286	-7.1
#WelcomeTweet	1113	7.1	gothacked	1565	-7.1
happy	85	5.6	sad	235	-5.3
good	143	5.1	bad	68	-5.1
awesome	87	5.2	horrified	28	-5.4
beauty	83	5.7	ugly	38	-5.3

Tabela 6.6: Dados de cada dicionário temático criado.

$$\text{Pontuação} = \text{PMI(termo, positiva)} - \text{PMI(termo, negativa)} \tag{6.1}$$

$$\text{PMI(termo, positiva)} = \log_2 \frac{\text{Frequência positiva do termo} \times \#\text{Tweets Negativos}}{\text{Frequência negativa do termo} \times \#\text{Tweets Positivos}} \tag{6.2}$$

Uma pontuação positiva indica que a palavra tem uma associação com um sentimento positivo e a sua magnitude indica o grau de associação (Mohammad et al., 2013). Todas as palavras e as respetivas pontuações foram adicionadas ao léxico.

A tabela 6.6 apresenta alguns exemplos de palavras que fazem parte do léxico. No início da tabela é possível ver as palavras cujo grau de associação a cada uma das polaridades é o maior. A maior parte dessas palavras são *hashtags* ou menções, uma vez que são as palavras mais frequentes no corpus. Por exemplo, no topo das palavras de polaridade positiva, a menção *@Harry_Styles* pertence a um membro de uma banda musical popular e na altura da extração dos *tweets* a informação que circulava pelo *Twitter* tinha uma conexão muito positiva, como se pode concluir com a pontuação obtida. Na segunda parte da tabela (as últimas quatro linhas), podem ver-se algumas palavras que geralmente estão presentes nos léxicos de polaridade, no entanto com uma pontuação mais baixa mas ainda assim bastante significativa.

É de notar que o objetivo deste léxico não era extrair as palavras de opinião já conhecidas (como *good* e *bad*) mas sim descobrir que outras palavras tem associações positivas ou negativas que com uma análise superficial não são detetadas.

Dicionário de Referências Temporais

Outro dicionário construído contém uma lista de palavras que são referências temporais. Este dicionário tem no total 79 palavras e está essencialmente dividido em três secções:

- **Referências a dias**: Contém todos os dias da semana, como por exemplo *monday* e *tuesday*.

- **Referências a meses**: Contém todos os meses do ano, como por exemplo *january* e *february*.
- **Referências temporais gerais**: Contém palavras que se referem a dados temporais, como por exemplo *annual*, *evening*, *yesterday*, *minute* entre outras.

Dicionário de Modificadores de Polaridade

Para a ferramenta de extração de polaridade foi imperativo a construção de um dicionário de palavras que modificam a polaridade. Alguns exemplos dessas palavras são: *but*, *no*, *never*, *neither*.

Dicionário Temático

Por último, foram criados vários dicionários que contêm palavras associadas a determinado tema, como por exemplo férias. Na totalidade foram criados 13 temas diferentes, como se pode observar na tabela 6.7. Estes dicionários foram usados tanto para a ferramenta de extração de polaridade, como para a ferramenta de extração de aspetos e entidades.

Tema	Número de Entradas	Exemplos
Aniversário	46	*cake*, *candy*, *gift*
Negócio	234	*ads*, *capital*, *company*
Feriados	50	*Easter*, *Thanksgiving*, *Labor Day*
Emprego	298	*banker*, *astronaut*, *actor*
Liderança	150	*captain*, *administrator*, *chief*
Saúde	188	*bacteria*, *bruise*, *concussion*
Locais	245	*Iceland*, *Finland*, *Croatia*
Política	111	*congress*, *election*, *government*
Restaurante	126	*cafeteria*, *chef*, *dessert*
Escola	85	*arithmetic*, *geography*, *homework*
Desporto	408	*athlete*, *biking*, *champion*
Lojas	150	*florist*, *gallery*, *hardware store*
Verão	86	*bikini*, *holiday*, *park*

Tabela 6.7: Dados de cada dicionário temático criado.

6.4 Dicionários para a Língua Portuguesa

Tal como para a língua inglesa, para o desenvolvimento de algumas ferramentas foi necessário a criação de diversos dicionários. Nesta secção são descritos os dicionários criados para a língua portuguesa. Tanto os dicionários temáticos (à exceção de dois temas), como o dicionário de referências temporais foram construídos seguindo as seguintes fases:

- **Tradução** Foi feita uma tradução automática dos dicionários já criados para inglês.
- **Revisão** Uma vez que da tradução automática podem surgir alguns erros foi feita uma revisão e correção manual a cada um dos dicionários. Sempre que necessário foram adicionadas novas entradas.

Lista de referências Numéricas

Foi construído um dicionário com cerca de 161 palavras que se referem a números. Por exemplo, "último", "vigésimo", "mil", "centésimo", entre outras.

Dicionário de Calão

Tal como para a língua inglesa foi importante criar um dicionário de calões, uma vez que um dos requesitos é permitir a análise de textos provenientes de redes sociais. Cada entrada no dicionário tem três características: o calão, a sua tradução e a polaridade. Na tabela 6.8 estão presentes alguns exemplos de calões por polaridade.

Polaridade	**Número de Entradas**	**Exemplos**
Positiva	16	*baum* (bom) *amour* (amor) *amgo* (amigo)
Negativa	8	*keixa* (queixa) *cvard* (covarde) *infelix* (infeliz)
Neutra	93	*eskeçe* (esquece) *dp* (depois) *fzr* (fazer)
Total	**117**	

Tabela 6.8: Distribuição de calões por polaridade e alguns exemplos.

Dicionário de referências Temporais

Um outro dicionário criado contém palavras que são referências temporais, ou seja, à semelhança do que foi descrito na secção 6.3 para língua inglesa, contém palavras como: "segunda", "março", "ontem", "manhã", "noite", entre outras.

Dicionário de Modificadores de Polaridade

Mais uma vez, para a ferramenta de extração de polaridade foi necessário a construção de uma lista de palavras que modificam a polaridade. Alguns exemplos são: "não", "nunca", "nem", "senão".

Dicionário Temático

Tal como para a língua inglesa, foi criado um conjunto de dicionários temáticos. Para além dos temas selecionados para a Língua Inglesa, descritos na tabela 6.7 da secção 6.3, outros dois dicionários foram acrescentados (ver tabela 6.9): lista de todas as cidades, distritos, freguesias e vilas de Portugal e lista com nomes e apelidos de pessoas.

Tema	**Número de Entradas**	**Exemplos**
Cidades, Distritos, Freguesias e Vilas	3578	Aveiro, Oliveirinha, São Bernardo
Nomes de pessoas	780	Alcino, Diogo, Laura

Tabela 6.9: Dados de cada dicionário temático criado.

6.5 *Parser* de Dependências para a Língua Portuguesa

Tal como explicado na secção 2.1.1, um *parser* de dependências permite a construção de árvores que explicitam as relações entre as palavras numa frase. Esta ferramenta tornou-se particularmente importante no desenvolvimento da extração de aspetos e entidades. Ao contrário da língua inglesa, para o português não existe nenhum *parser* de dependências que possa ser usado em âmbito comercial, e como tal foi necessário o seu desenvolvimento. Na Figura 6.4, estão representadas as diferentes fases, tanto de criação do modelo como a sua execução.

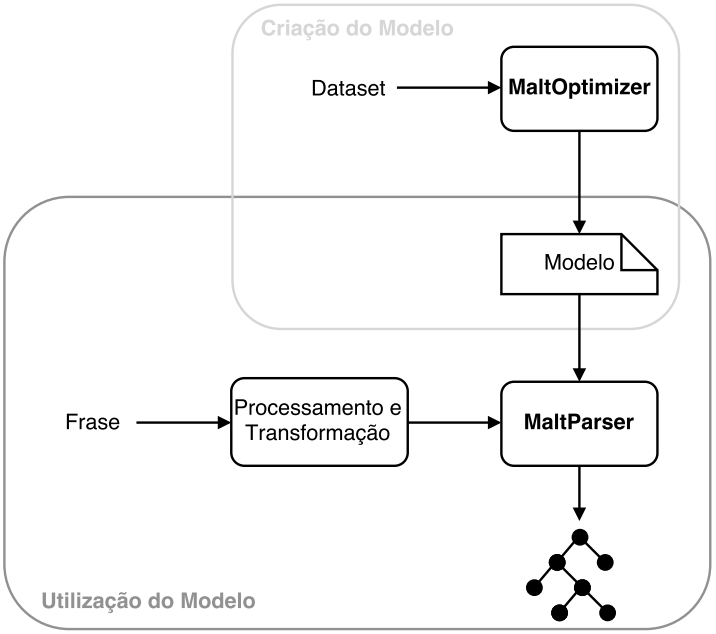

Figura 6.4: Processo de criação de um *parser* de dependências para o Português.

Corpus

O corpus de treino usado foi o Bosque[12](Afonso et al., 2002), descrito na secção 2.5.1. O corpus é composto por 9.071 frases e 206.678 palavras. Cada palavra no corpus é composta por várias características, como descrito na tabela 6.10.

Uma vez que a Wizdee não possui ferramentas para extrair todas as informações das palavras, como por exemplo saber o modo e tempo verbal de um verbo, alguns dados presentes na coluna de *FEATS* tiveram de ser removidos do corpus, ficando apenas o seu género e número.

O facto de este ser um corpus composto essencialmente por texto estruturado, a sua qualidade pode ser afetada quando aplicado a texto não estruturado. No entanto, a possibilidade de criação de um corpus com qualidade e especializado em texto não estruturado foi descartada, devido aos limites temporais existentes.

[12]Disponível em `http://www.linguateca.pt/floresta/corpus.html`.

ID	FORM	LEMA	CPOSTAG	POSTAG	FEATS
1	Há	haver	v	v-fin	S
2	,	,	punc	punc	_
3	em	em	prp	prp	_
4	o	o	art	art	M|S
5	ar	ar	n	n	M|S
6	,	,	punc	punc	_
7	uma	um	art	art	F|S
8	certa	certo	pron	pron-det	F|S
9	ideia	ideia	n	n	F|S
10	de	de	prp	prp	_
11	invasão	invasão	n	n	F|S
12	.	.	punc	punc	_

Tabela 6.10: Exemplo do formato CoNLL para a frase "*Há, no ar, uma certa ideia de invasão.*"

MaltOptimizer

Uma ferramenta muitas vezes usada pela sua facilidade de utilização na criação de *parser* de dependências é o *MaltParser*[13]. O *MaltParser* permite, a partir de um corpus anotado, induzir um modelo de *parsing*, bem como, a partir de um modelo, extrair a árvore de dependências dada uma frase.

Apesar da sua facilidade de utilização, a criação de um *parser* envolve muitas experiências e otimizações de parâmetros, de forma a encontrar o modelo que melhor se adapta aos dados. A ferramenta *MaltOptimizer*[14], que usa o sistema *MaltParser*, permite fazer isso tudo de forma mais abstrata, fazendo todas essas otimizações internamente. Como tal, de forma a facilitar a criação de modelo, o *MaltOptimizer* foi usado.

MaltOptimizer começa por fazer uma análise ao corpus de treino, como por exemplo extrair o número de frases, identificar as classes gramaticais, entre outras. Essas estatísticas são usadas na segunda fase para a escolha de algoritmos que melhor se adaptam. Na segunda fase, cada um dos algoritmos escolhidos são usados e os seus parâmetros otimizados e no fim é escolhido o algoritmo que melhor resultados apresenta. Na terceira fase, *MaltOptimizer* faz algumas experiências com as diferentes combinações de informações presentes na coluna *FEATS*, tentando perceber qual o conjunto de características que melhor resultados obtém. Por fim, na última fase, o modelo final é criado usando as definições que nas fases anteriores melhor resultado apresentaram.

Processamento e Transformação

Após a criação do modelo, este é usado para a extração da árvore de dependências de novas frases. Para cada frase é feito um processamento que consiste na separação de *tokens*, identificação das classes gramaticais, número, género e lematização. Por fim, a frase é transformada para formato CoNLL[15] usando a mesma simbologia que o dataset[16],

[13] Ver mais informações em `http://www.maltparser.org/`

[14] Ver mais informações em `http://nil.fdi.ucm.es/maltoptimizer/`

[15] Ver mais informações sobre o formato CoNLL `http://ilk.uvt.nl/conll/#dataformat`

[16] Ver simbologia do Bosque em `http://beta.visl.sdu.dk/visl/pt/symbolset-floresta.html`.

como já apresentado na tabela 6.10.

MaltParser

Depois de obter a frase em formato *CoNLL*, estes dados são enviados ao *MaltParser*, que juntamente com o modelo criado devolve uma árvore de dependências.

6.6 Ferramenta de Extração de Opiniões para a Língua Inglesa

Nesta secção são apresentados os detalhes do desenvolvimento da ferramenta de extração de opiniões (ver secção 2.3) para a Língua Inglesa. O que se pretende com esta ferramenta é conseguir obter as seguintes informações relacionados com opiniões:

- **Autor e data da opinião:** A extração do autor da opinião e a data da sua publicação. No âmbito deste trabalho o autor é assumido como sendo o autor da publicação, uma vez que numa perspetiva comercial é mais prioritário perceber o que cada um fala do que fazer a ligação entre a opinião e o autor dela.
- **Entidade:** Uma vez que esta ferramenta tem como finalidade o uso comercial, foi decidido que uma entidade pode representar uma marca, produto ou serviço.
- **Aspetos:** Por outro lado, um aspeto é uma característica dessa marca, ou seja uma característica da entidade.
- **Polaridade da relação Entidade-Aspeto:** Usando as entidades e aspetos extraídos, as relações entre eles são analisadas, de forma a perceber qual a entidade a que um determinado aspeto se refere. Após a análise dessa relação é extraída a polaridade. Por exemplo, na frase em baixo, a entidade é "**My Vodafone**", o aspeto é o "**registo**" e a polaridade é negativa, uma vez que o utilizador queixa-se de não conseguir efetuar um registo.

  *"Alguém me diz porque não consigo fazer **registo** no **My Vodafone**?"*

- **Polaridade de toda a frase:** Para além da polaridade associada a um aspeto ou entidade, também é possível extrair a polaridade de toda a opinião no seu geral.

De seguida, cada uma das ferramentas desenvolvidas são analisadas e descritas em detalhe.

6.6.1 Extração de Polaridade

A ferramenta de extração de polaridade tem como objetivo classificar o texto proveniente das redes sociais, tendo em conta a sua polaridade, que pode ser positiva, negativa ou neutra. Nesta secção apenas o texto escrito na Língua Inglesa é tido em conta.

A abordagem explorada é baseada num algoritmo de aprendizagem automática, as Máquinas de Vetores de Suporte, apresentadas na secção 2.2.2, uma vez que é uma das abordagens mais populares e que tem obtido bons resultados (Nakov et al., 2013).

Na Figura 6.5 é possível visualizar as diferentes fases que compõem esta ferramenta. Uma vez que abordagem escolhida envolve as Máquinas de Vetores de Suporte foi usada a biblioteca Scikit-Learn[17] do *Python*, que já possui a implementação desse algoritmo. Como

[17]Disponível em `http://scikit-learn.org/stable/`

a plataforma *Wizdee* está maioritariamente desenvolvida em Java, esta ferramenta divide-se em dois blocos: uma parte em Java, que processa os textos e cria as características que são depois enviadas por uma API *REST* para um serviço web. A parte implementada em *Python*, permite o treino do modelo que depois é guardado. Sempre que se quer classificar um texto consoante a sua polaridade, as suas características são extraídas e enviadas para o *Python*, que usa o modelo guardado para extrair a polaridade que depois é retornada ao Java. A *Wizdee* já possuí uma implementação da ferramenta que permite a comunicação entre *Java* e *Python*, por isso não foi necessário criar esse sistema de comunicação.

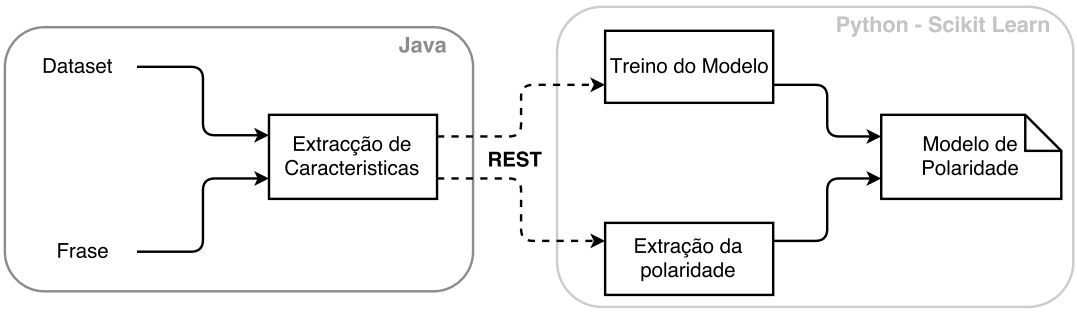

Figura 6.5: Diferentes fases do treino e extração de polaridade para o Inglês.

Corpus

O corpus de treino usado foi criado pelos organizadores do *Workshop on Semantic Evaluation* (SemEval-2014[18]). O corpus contém textos extraídos da rede social *Twitter* que foram anotados usando uma aplicação de *crowdsourcing* (Nakov et al., 2013). O corpus está anotado nas três classes pretendidas: positivo, negativo e neutro.

Na Figura 6.11 é possível observar a distribuição do corpus entre as diferentes classes. O corpus de treino contém 9.619 casos, sendo que 18% desses casos possuem polaridade negativa, 42% são casos neutros e 40% são casos positivos.

Nome	Total	Positivo	Negativo	Neutro	Licença
Corpus de Treino	10861	4306	1985	4569	CC-BY v3

Tabela 6.11: Estatísticas do corpus de treino.

Extração de Características

Para podermos classificar texto em opiniões positivas, negativas ou neutras, é preciso inicialmente transformar esse texto num vetor de características. Foram extraídos dois tipos de características: linguísticas e não supervisionadas. As características linguísticas englobam informações extraídas a partir de léxicos e tarefas de PLN, enquanto que as automáticas são extraídas usando algoritmos de *word embeddings*. Nesta secção, são descritos em detalhe os processos de extração para cada um desses tipos.

Características Linguísticas Usando algumas ferramentas e léxicos, foram criadas cerca de 445 características que se podem dividir essencialmente em dois tipos: características de conteúdo e de léxico. Todas as características linguistas desenvolvidas estão descritas no Anexo A.

[18]Mais informações em `http://alt.qcri.org/semeval2014/`

Word Embeddings Uma outra forma de representação de texto é através dos chamados *Word Embeddings* (WE). Os WE (Bengio et al., 2013) permitem representar palavras de um dicionário em vetores com uma dimensionalidade reduzida. Na prática, o objetivo é conseguir mapear as palavras em vetores, como por exemplo: $W(Portugal) = (0.3, 0.2, 0.8, ...)$ e $W(Espanha) = (0.36, 0.25, 0.78, ...)$. A ideia base é permitir que duas palavras semelhantes tenham vetores de representação semelhantes. Ou seja, no exemplo anterior, Portugal e Espanha são ambos países europeus, ou seja o contexto semântico e sintático em que são usados é potencialmente semelhante, por isso os *word embeddings* devem conseguir transmitir essa semelhança.

Na Figura 6.6 é possível visualizar um caso de exemplo onde cada país está representado por um vetor, neste caso de duas dimensões. Os *word embeddings* permitem através da distancia de cada vetor perceber se o nome de determinado país é usado em contextos semelhantes. Neste caso, os *word embeddings* permitiram a separação dos países por dois continentes, a Europa (a verde) e a Ásia (a cinzento). Um outro exemplo de semelhança é o facto de Portugal, Grécia e Alemanha estarem próximos um dos outros.

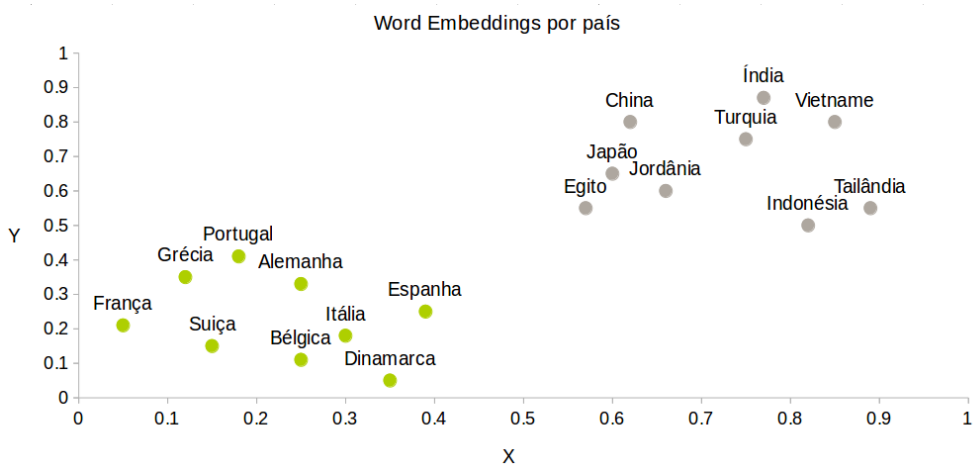

Figura 6.6: Exemplo de *word embeddings* que representam diferentes países.

Para a criação desses vetores foi usada a biblioteca *Word2vec* como base. A biblioteca *Word2vec* implementa dois algoritmos diferentes para a criação dos vetores: *continuous bag-of-words* (Mikolov et al., 2013) e *continuous skip-gram* (Mikolov et al., 2013). Basicamente, através de um conjunto de documentos como entrada, é construído um vocabulário de todas as palavras existentes para depois criar os vetores de representação.

Tendo como objetivo o uso destas representações na resolução de um problema de classificação de polaridade, foi decidido construir três tipos de representações: vetores que captam apenas semelhanças de contexto (daqui em diante são referenciados como **Word Embeddings Simples**), vetores que têm em conta a polaridade (daqui em diante são referenciados como **Word Embeddings de Polaridade**) e vetores de semelhança que são construídos a partir dos *Word Embeddings*.

Word Embeddings Simples A ferramenta *Word2vec* já possui uma implementação que permite capturar as semelhanças entre palavras tendo em conta o seu contexto. Na realidade, *Word2vec* disponibiliza também um modelo já treinado usando um corpus extraído a partir da *Google News*[19], no entanto, foi entendido que seria benéfico treinar um modelo usando texto extraído diretamente das redes sociais.

[19]Disponível em `https://news.google.pt/`

Como tal, para este tipo de *Word Embeddings* foi construído um corpus com cerca de oito milhões de *tweets* extraídos do *Twitter*. O *Twitter* foi escolhido como fonte de dados tendo em conta a facilidade de extração de *tweets*, uma vez que permite extrair textos em tempo real sem obrigar a restringir por utilizador ou página, como no caso do *Facebook*.

Para treino, foi usado o algoritmo *continuous bag-of-words*, que transformou cada palavra num vetor de 20 dimensões.

Word Embeddings de Polaridade Para este tipo de *word embeddings* foi necessário modificar a implementação da ferramenta *Word2vec* que não considerava as informações de polaridade. Tal como o anterior, foi necessário a construção de um corpus de treino cujas distribuições estão apresentadas na Tabela 6.12.

Nome	Total	Positivo	Negativo	Neutro
Corpus do Twitter	10.647.936	3.665.713	3.652.822	3.329.401

Tabela 6.12: Estatística do corpus construído para as *Word Embeddings* de Polaridade.

Na totalidade, foram extraídos cerca de 10 milhões de *tweets*, sendo que se teve o cuidado se ter uma distribuição semelhante de positivos, negativos e neutros. A obtenção de *tweets* positivos e negativos foi feita assumindo que sempre que um *tweet* possui o *smile* "*:)*" ou o *smile* "*:(*" é um *tweet* positivo ou negativo, respetivamente. Para a obtenção de *tweets* neutros, fez-se uma análise a cada *tweet*, sendo que um *tweet* só seria considerado neutro se não apresenta-se nenhuma palavra que estivesse presente nos dicionários de sentimento. É de notar que a polaridade está associada a um *tweet* e não a uma palavra. Ou seja, o que se pretende é perceber se uma determinada palavra é geralmente referida num contexto mais negativo, mais positivo ou neutro.

Como anteriormente mencionado, foi usado o algoritmo *continuous bag-of-words*, no entanto com algumas alterações. Uma das formas encontradas para introduzir informação da polaridade no algoritmo foi a alteração do gradiente descendente tendo em conta a polaridade. A introdução desta informação podia resultar em ajustes demasiado bruscos, por isso para além do gradiente descente estar dependente da taxa de aprendizagem, ou em inglês *learning rate*, foi acrescentada uma constante que obriga a ajustes mais pequenos, tornando assim a aprendizagem um pouco mais lenta. Cada palavra foi transformada num vetor de 10 dimensões.

Vetor de semelhança Uma outra utilização possível dos *Word Embeddings* é a criação de um vetor de semelhança. Basicamente, é feito uma comparação de cada palavra a um conjunto pré-definido de palavras que são naturalmente associadas a uma determinada polaridade, percebendo se estas estão ou não em espaços opostos.

Para esta representação foi definido o seguinte conjunto de palavras: *Words = ["good", "nice", "love", "great", "awesome", ":)"]*. Por cada palavra de entrada é construído o seu vetor de *Word Embedding* com polaridade, que depois é usado para o cálculo da distância angular entre esse vetor de representação com cada uma das palavras do conjunto *Words*. Se a distância for pequena, quer dizer que a palavra em questão é geralmente usada num contexto positivo.

A distância entre as duas palavras é calculada usando a fórmula da distância angular entre dois vetores A e B (ver equação 6.3).

$$\text{Semelhança}_{A,B} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{k}(A_i \times B_i)}{\sqrt{\sum_{i=1}^{k}(A_i)^2} \times \sqrt{\sum_{i=1}^{k}(B_i)^2}} \tag{6.3}$$

6.6.2 Extração de Aspetos

Como descrito anteriormente, um aspeto representa uma característica de uma marca, produto ou serviço, como por exemplo no produto "*Samsung Galaxy S6*", podemos ter algumas características como: "*screen*", "*sound*", "*camera*", "*price*", entre outras.

Na Figura 6.7 estão representadas todas as fases da abordagem linguística desenvolvida para esta ferramenta.

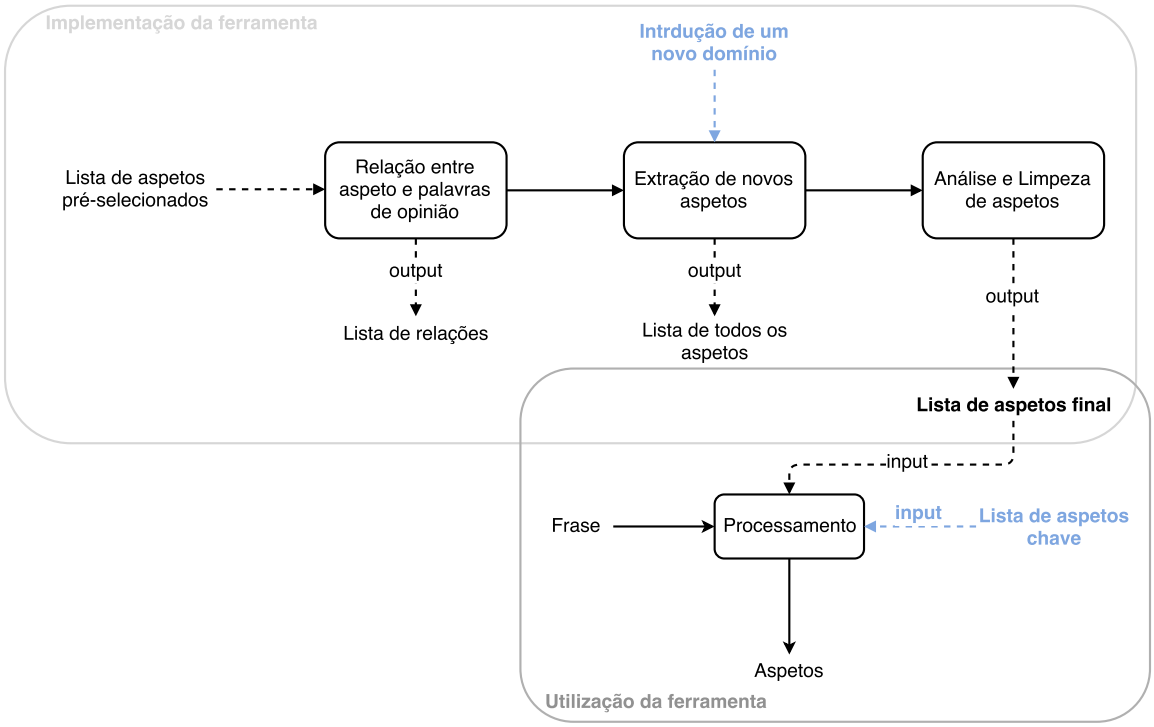

Figura 6.7: Representação das várias fases de extração de aspetos.

Lista de Aspetos Pré-Selecionados

De diversos conjuntos de textos de diferentes domínios, foram inicialmente extraídos todos os substantivos que, posteriormente, foram guardados em ficheiro ordenados pela sua frequência. De seguida, foi feita uma seleção manual de 38 aspetos. Essa seleção teve em conta a frequência e o nível de certeza de que o substantivo pudesse realmente ser um aspeto. Alguns dos substantivos escolhidos foram: "*battery*", "*quality*", "*price*", "*signal*", "*capacity*", entre outros. Esta lista é importante para as próximas fases, uma vez que é a partir dela que se vão tentar descobrir todos os outros aspetos menos frequentes.

Relações entre Aspetos e Palavras de Opinião

Para as ferramentas de extração de aspetos não podemos assumir que todos os substantivos são aspetos, como por exemplo na frase em baixo, onde os substantivos estão assinalados a negrito e onde apenas o substantivo "*quality*" é um aspeto.

"*Took me half an* ***hour*** *to understand why the* ***quality*** *was bad.*"

Como tal, foi preciso perceber quais eram os substantivos que realmente podiam ser aspetos. Usando a lista de aspetos conhecidos, criada na fase anterior, foram estudadas as relações entre os aspetos já conhecidos e as palavras de opinião. Por exemplo, na Figura 6.8, "*quality*" faz parte da lista de aspetos conhecidos, por isso tenta-se encontrar a relação desse aspeto com a palavra de opinião (neste caso "*bad*"). A relação extraída é a relação **NSUBJ** que representa uma relção de sujeito. Todas as relações encontradas são guardadas e usadas para ajudar a encontrar aspetos ainda não conhecidos.

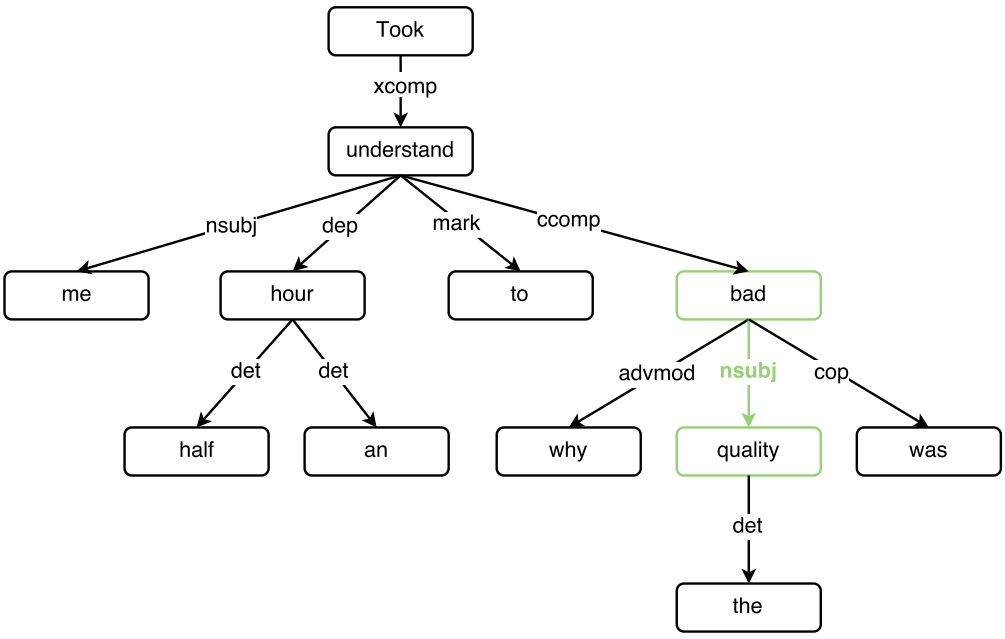

Figura 6.8: Exemplo de relação entre aspeto conhecido e uma palavra de opinião.

Extração dos Novos Aspetos

Usando as relações extraídas na fase anterior novos aspetos foram extraídos. Para cada frase foi construída a sua árvore de dependências e encontradas todas as expressões de polaridade. A partir dessas expressões de polaridade, verifica-se se têm alguma relação extraída na fase anterior que a relacione com um substantivo. Caso isso se verifique, quer dizer que foi encontrado um novo possível aspeto.

Análise e Limpeza de Aspetos

O facto de a relação entre substantivo e palavra de opinião se verificar não é garantia de que esse substantivo seja mesmo um aspeto, por isso foi necessário adicionar uma fase de análise e limpeza dos aspetos falso positivos.

Cada aspeto candidato foi analisado e excluído caso se mostrasse positivo para qualquer uma das seguintes regras:

- É uma palavra de opinião
- É uma *stopword*
- É uma palavra de referência temporal (ver dicionário em 6.3)
- É uma palavra comum
- É um verbo comum
- É um adjetivo comum
- É uma palavra composta que contém símbolos ou números
- Não é considerado um substantivo quando classificado individualmente pelo identificador de classes gramaticais

Se o aspeto não pertencer a nenhuma das regras é considerado um aspeto correto. Todos os aspetos que passam esta fase são inseridos na lista de aspetos final referida na Figura 6.7 que é usada para extrair aspetos de novas frases.

Processamento

Para a fase de execução da ferramenta a partir de uma frase é feito um processamento que tem em conta a lista de aspetos final que foi criada na fase anterior. O processo passa pelas seguintes etapas:

- A frase passa por um processo de identificação de palavras, identificação de classes gramaticais, *lematização* e *chunking*.
- Para todas as palavras da frase é procurado o seu lema na lista de aspetos.
- De forma a mitigar os erros, no caso da palavra se encontrar na lista esta é procurada nos *chunks* do tipo NP. Caso esteja presente num *chunk*, esta é finalmente considerada um aspeto.

Uma opção acrescentada foi a possibilidade de cada cliente definir uma lista de aspetos específica que quer analisar e por isso, para além da lista de aspetos construída automaticamente, é possível também ter uma lista de aspetos-chave específica de um cliente.

Adaptação a Novos Domínios

Uma grande preocupação que existe quando se desenvolve uma ferramenta de extração de aspetos é a sua capacidade de adaptação a novos domínios. Uma vez que a qualidade desta ferramenta depende da lista de aspetos final que foi conseguida na fase de desenvolvimento, foi necessário criar um processo automatizado e de fácil utilização para que a lista de aspetos seja adaptada a um novo domínio.

Como se pode ver na Figura 6.7, se quisermos adaptar, por exemplo, ao domínio das companhias aéreas, basta recolher diversos textos de opinião desse mesmo domínio, introduzir na fase de Extração de novos aspetos e a partir daí o sistema já conhece o domínio.

Neste momento a ferramenta está adaptada para os seguintes domínios:

- Hotelaria
- Telemóveis

- Cinema
- Automóveis
- Livros
- Tecnologia (telemóveis, câmaras digitais, *routers*, portáteis, televisões, etc)

6.6.3 Extração de Entidades

Depois da extração de aspetos, as entidades são extraídas. Como mencionado anteriormente, uma entidade é uma marca, produto ou serviço. Tal como para a extração de aspetos, foi optado por uma abordagem linguística. No entanto, ao contrário dos aspetos, que geralmente são fixos uma vez que dentro do mesmo domínio os aspetos não sofrem grandes alterações, nas entidades isso já não acontece, uma vez que há sempre novas marcas ou produtos. Sendo assim o processo não pode depender de uma lista fixa como a ferramenta de extração de aspetos.

Na Figura 6.9 é possível visualizar todas fases de extração de entidades que são detalhadas nesta secção. Inicialmente, é feita uma pré-análise que permite a criação de um recurso (lista de relações) necessário durante a execução da ferramenta, ou seja, quando se quer a partir de uma frase extrair todas as suas entidades.

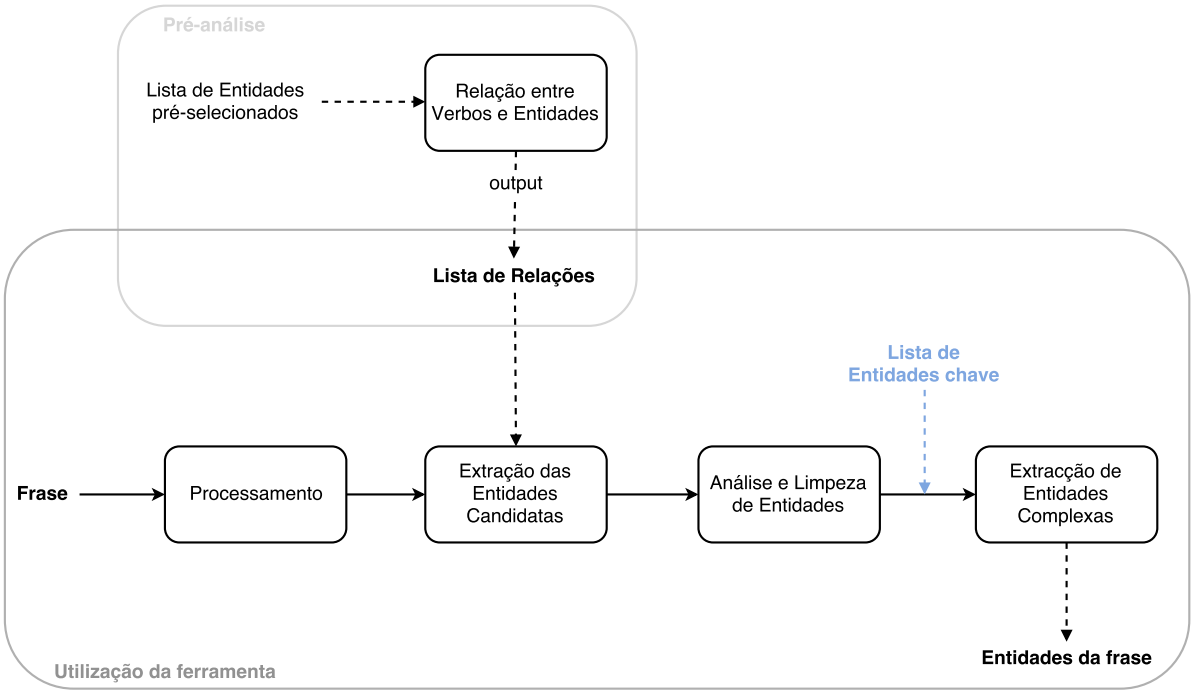

Figura 6.9: Representação das várias fases de extração de entidades para o Inglês.

Lista de Entidades Pré-Selecionados

Numa primeira fase foi necessário a criação de uma pequena lista de entidades frequentes. Essa lista foi criada de forma manual a partir de um conjunto de frases extraídas da página da *Samsung* do *Facebook*. A lista é composta por 15 entidades, como por exemplo: "*samsung*", "*galaxy*", "*s5*", "*lollipop*", "*vodafone*", "*s6*", entre outras

Extração de Relações entre Verbos e Entidades

Para cada uma das entidades pré-definidas anteriormente são estudadas as suas relações com os verbo. Basicamente para cada frase é construída a sua árvore de dependências e identificados todos os nós que representam verbos e os nós que são entidades pré-definidas. Para cada nó entidade é encontrado o nó verbo mais perto, ou seja cujo grau de dependência é maior (menor distância).

Na Figura 6.10 é possível visualizar árvore de dependências da frase "*My S5 stops working properly when it was fine before the update.*" em que foram detetados 3 verbos e uma entidade. A relação relevante é a relação entre a entidade "*S5*" e o verbo "*stops*" porque é a relação com menos passos (apenas tem um passo de distância). Cada uma dessas relações extraídas é guardada para posterior utilização.

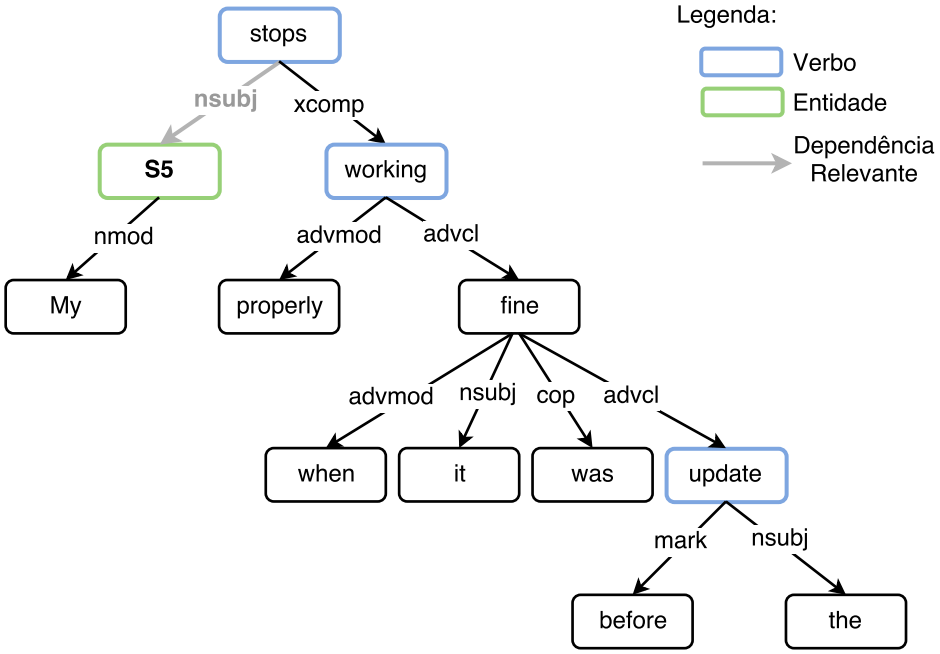

Figura 6.10: Representação das várias fases de extração de entidades para o Inglês.

Uma observação importante é que usar apenas as relações para extrair as entidades não é suficiente. Por exemplo, ainda na Figura 6.10, ao extrairmos a relação "*nsubj*", que exprime uma relação de sujeito, para obter todas as entidades que se relacionem com um verbo através dessa dependência, quer dizer que também extraímos a palavra "*the*" que está nas mesmas condições que a entidade "*S5*". Por isso é necessário a fase de análise e limpeza de entidades que é mencionada na Figura 6.9.

Processamento

Quando a partir de uma frase se quer extrair todas as suas entidades, começa-se pela fase de processamento. Tal como na ferramenta de extração de aspetos, cada frase passa por um processo de identificação dos *tokens*, identificação de classes gramaticais, *lematização* e *chunking*. De forma a facilitar o resto do processo são removidos *tokens* cuja informação é considerada pouco importante para esta tarefa, como por exemplo os *urls*, *smiles* e menções.

Extração de Entidades Candidatas

Nesta fase são extraídas todas as entidades candidatas da frase usando as relações de dependência extraídas na pré-análise. O processo é um pouco semelhante ao referido na ferramenta de extração de aspetos, onde se começa por extrair todos os nós que são verbos, usando a árvore de dependências da frase. Por cada um desses nós verbo, tenta-se encontrar novos nós que obedecem a qualquer uma das relações extraídas na pré-análise. Cada um desses novos nós é uma entidade candidata. Tal como mencionado anteriormente, usar apenas as relações não é suficiente pois vão existir várias entidades candidatas que na realidade não o são e por isso todos essas entidades candidatas passam pela seguinte fase de análise e limpeza.

Análise e Limpeza de Entidades

Uma das fases importantes é a de análise e limpeza de entidades candidatas. É nesta fase que se decide quais são realmente as entidades. Para tal, cada entidade candidata que se enquadra nos seguintes pontos é excluída da lista de entidades:

- **Não é substantivo** Todas as entidades devem ser nomes ou nomes próprios. Todas as outras classes gramaticais são excluídas.
- **É um aspeto** Durante a análise da frase são extraídos todos os aspetos. Se uma determinada palavra for um aspeto é excluída como uma possível entidade.
- **Não está presente nos *chunks* do tipo NP** Como explicado na fase de Processamento, usa-se a ferramenta *chunker* para extrair todos os seus *chunks*, no entanto apenas *chunks* do tipo NP são considerados. Se a entidade candidata não estiver representada nesse tipo de *chunks* é excluída.
- **É uma palavra de opinião** Tal como na extração de aspetos, uma entidade não pode ser uma palavra de opinião conhecida.
- **É uma *stopword*** A entidade candidata não pode ser uma *stopword*.
- **É uma referência temporal** A entidade não pode estar presente nos dicionários de referência temporal descritos em 6.3
- **É um substantivo, verbo ou adjetivo comum** Se a entidade candidata pertencer a um conjunto das palavras inglesas mais comuns é excluída.

Para além das entidades extraídas de forma automática por todo o processo descrito em cima, é possível cada cliente inserir uma lista de entidades-chave. Ou seja, para além das entidades extraídas automáticas as palavras incluídas nessa lista são também consideradas entidades sem passarem pelo processo de análise e limpeza.

Extração de Entidades Complexas

Nesta última fase pretende-se perceber que entidades são complexas. Uma entidade complexa é uma entidade que é composta por mais do que uma palavra, como por exemplo "*Samsung Galaxy S6*".

As entidades complexas são extraídas usando os *chunks* da frase. Ou seja, as palavras que compõem o *chunk* onde a entidade está inserida é considerada a entidade complexa. De forma a excluir algumas palavras que possam estar no *chunk* e que na realidade não fazem parte da entidade, todas as palavras que não sejam substantivos ou números são

removidas, como se pode ver no exemplo da Figura 6.11. A decisão de manter também os números deveu-se ao facto da existência de entidades do tipo "*Iphone 4*" em que é importante manter o número pois, neste caso, a alteração do número indica um produto diferente.

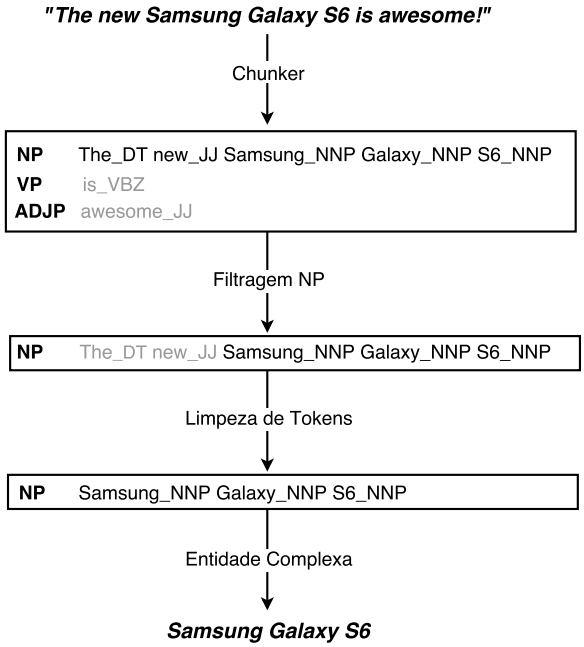

Figura 6.11: Exemplo de Extração de Entidades Complexas usando o *Chunker*.

6.6.4 Extração de Quíntuplos

Nesta secção é descrita em detalhe o desenvolvimento da ferramenta de extração de quíntuplos. Tal como explicado anteriormente um quíntuplo é composto pelos cinco seguintes componentes: autor, data, entidade, aspeto e polaridade. Esta ferramenta permite fazer a ligação entre cada um desses componentes, usando algumas das ferramentas desenvolvidas e descritas nas secções anteriores.

Na Figura 6.12 é possível visualizar as diferentes fases necessárias à extração de quíntuplos. De forma a extrair todas as informações necessárias a ferramenta necessita, inicialmente de receber o seguintes dados:

- **Texto:** O texto do qual se pretende extrair os quíntuplos existentes.
- **Autor e data da opinião:** Tanto o autor como a data são informações que fazem parte do quíntuplo. Neste trabalho, o autor é assumido como sendo o autor da publicação. Todos estes dados são extraídos no momento da extração do texto das redes sociais, por isso não é necessário nenhuma análise posterior.
- **Hierarquia:** Permite-se que no caso de não existirem aspetos, o quíntuplo se refira ao aspeto "GERAL". No entanto, o mesmo não deve acontecer com as entidades, ou seja o texto deve ter uma entidade associada. Caso o texto analisado não contenha nenhuma entidade essa informação deve ser extraída usando a hierarquia de textos. Por exemplo, no contexto das redes sociais, cada comentário/publicação segue uma hierarquia, ou seja, um comentário pode ser a resposta a outro comentário (a que se chama de pai), ou a resposta a uma publicação.

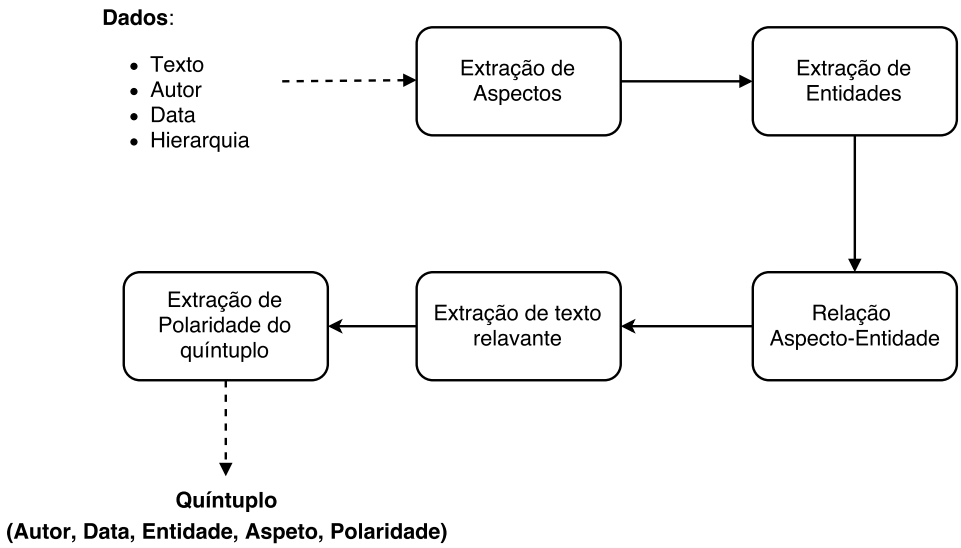

Figura 6.12: Representação das várias fases de extração de quíntuplos.

Usando alguns dados recebidos, são extraídos todos os aspectos e entidades do texto usando as ferramentas descritas em 6.6.2 e em 6.6.3.

Relação Aspeto-Entidade

Após extrair todos os aspectos e entidades é preciso perceber quais são as relações entre eles, ou seja, por exemplo, tendo mais que uma entidade, quais são os aspectos que se relacionam com a entidade A e os aspectos que estão relacionados com a entidade B.

Na tabela 6.13 é possível observar as diferentes situações que podem acontecer nesta fase. Quando a frase não apresenta nenhuma entidade, recorre-se à informação hierárquica, ou seja, se a publicação tiver como origem uma outra publicação, a entidade dessa publicação de origem é considerada também a entidade da nova publicação. No contexto da rede social do *Facebook* é assegurado que um texto nunca tem quíntuplos cuja entidade é desconhecida, uma vez que em último caso a entidade é considerada a página em que a publicação foi feita. Já no *Twitter*, isto não é assegurado, uma vez que os *tweets* não são publicados em páginas específicas. No caso de nenhum texto de hierarquia superior ao texto analisado tenha uma entidade, o quíntuplo assume a entidade "Desconhecida".

Se existir apenas uma entidade na frase toda e apenas um aspeto, esses dois componentes são considerados relacionados, ou seja o aspeto é considerado como uma caraterística da entidade identificada. O mesmo acontece se existir mais que um aspeto, em que todos os aspetos são considerados parte da única entidade identificada. Um outro ponto que também é relevante é quando o número de aspetos é zero. Sempre que isso acontece é introduzido no triplo o aspeto "GERAL" que representa que o texto está a falar na entidade no seu geral e não numa característica específica.

Por fim, se no texto for identificada mais que uma entidade é necessário perceber quais são os aspetos que pertencem a quais entidades. Para tal, foi desenvolvido um sistema que, usando as árvores de dependências, percebe qual a dependência mais forte entre o aspeto e todas as entidades identificadas. Ou seja, a partir de um aspeto, analisa-se as dependências com todas as entidades encontradas e a dependência mais forte (a relação com menor número de passos) representa a entidade que o aspeto está relacionada, como se pode ver na Figura 6.13, que representa a árvore de dependências para a frase "*The camera of my S5 is not working and Samsung customer service doesn't do anything.*", existem duas

Nº de Entidades	Nº de Aspetos	Exemplo
0	0	Entidade Pai = A Relação = (A, "GERAL")
	1	Entidade Pai = A Aspeto = Z Relação = (A, Z)
	>1	Entidade Pai = A Aspeto = X, Z Relações = (A, X); (A,Z)
1	0	Entidade = A Relação = (A, "GERAL")
	1	Entidade = A Aspeto = Z Relação = (A, Z)
	>1	Entidade = A Aspeto = X, Z Relações = (A, X); (A,Z)
>1	0	Entidade = A, B Relações = (A, "GERAL"); (B, "GERAL")
	1	Entidade = A,B Aspeto = Z Relação = (A, Z); (B, "GERAL")
	>1	Entidade = A,B Aspeto = X, Z Relações = (A, X); (B,Z)

Tabela 6.13: Todas as diferentes possibilidades de relações entre entidades e aspetos.

entidades, "*S5*" e "*Samsung customer service*", e o aspeto "*camera*". O aspeto é confrontado com todas as entidades existentes e, no final, concluí-se que o aspeto refere-se à entidade "*S5*", uma vez que o nível de dependência é maior (apenas uma relação de distância - *nmod*).

No mesmo exemplo considerado na Figura 6.13, desta fase obtém-se os seguintes quíntuplos:

Quíntuplo #1 = (Sara, 18/6/2015, *Iphone*, *camera*, ?)
Quíntuplo #2 = (Sara, 18/6/2015, *Samsung customer service*, GERAL, ?)

Extração do Texto Relevante

Depois de obter todos os quíntuplos do texto com informações sobre as entidades e aspetos, falta perceber qual a polaridade desse quíntuplo. Como tal é necessário perceber que parte da frase está relacionada com esse quíntuplo.

Para a divisão de texto existem quatro tipos de regras:

- **Orações**: Usando a árvore de constituência a frase é dividida em duas se existir um nó que divide a frase em duas orações em que de um lado está uma entidade/aspeto e do outro lado está outra. Esta divisão só é feita se permitir a separação de dois componentes, como por exemplo separar dois aspetos. Existem diferentes etiquetas que permitem esta separação, como por exemplo S, SBAR, SBARQ, SINV e SQ.

- **Conjunções**: Caso as regras anteriores ainda não consigam separar todos os aspetos e entidades, experimenta-se separar usando conjunções coordenativas, como por exemplo "*and*", "*or*" ou "*but*".

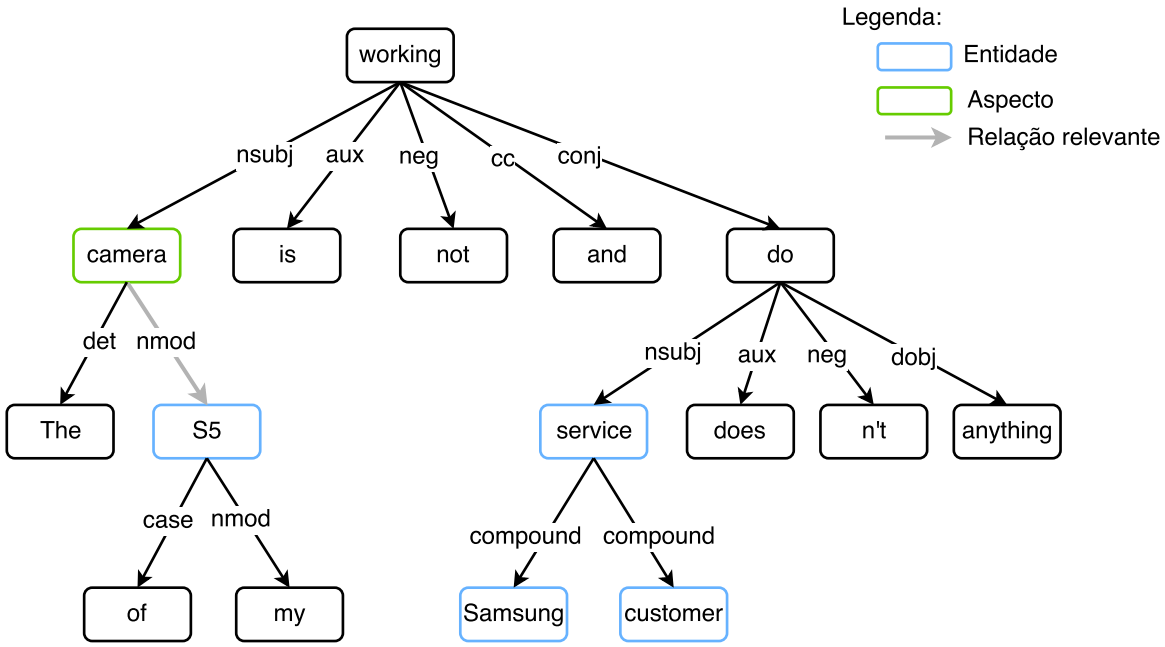

Figura 6.13: Exemplo de extração de relação entre entidades e aspetos.

- **Símbolos**: Se ambas as regras falharam, recorre-se aos símbolos. Por exemplo, na existência de uma vírgula que divida dois aspetos esta regra é aplicada.
- **Complexo**: Por fim, se todas as regras em cima falharem, tenta-se perceber se o aspeto/entidade pode ser do tipo complexa. Neste contexto, considera-se complexo se as entidades ou aspetos estiverem divididos por uma preposição ou conjunção subordinativa como por exemplo, "*on*", "*of*", "*in*" entre outras.

Se todas as regras descritas em cima falharem, assume-se que a separação de texto não é possível por isso considera-se todo o texto como relevante.

Um exemplo da aplicação dessas regras está presente na Figura 6.14, onde se tem duas entidades e um aspeto e por isso dois quíntuplos e pretende-se perceber qual o texto relevante para cada quíntuplo. Começando a aplicar a primeira regra, regra das orações, consegue-se dividir a frase em duas separando assim ambas as entidades ficando com: "*The camera of my Iphone is not working*" e "*and Samsung customer service does not do anything.*". Depois desta separação já não existe no mesmo bloco de texto diferentes entidades ou aspetos.

No entanto, se não existisse a oração, também era possível separar a frase usando a conjunção coordenativa (CC), cujo resultado seria o mesmo.

Sendo assim desta fase e usando o exemplo em cima conseguimos extrair a seguinte informação:

Quíntuplo #1 = (Sara, 18/6/2015, *Iphone*, *camera*, ?)
Texto Relevante = "*The camera of my Iphone is not working*"

Quíntuplo #2 = (Sara, 18/6/2015, *Samsung customer service*, GERAL, ?)
Texto Relevante = "*and Samsung customer service does not do anything.*"

Extração da Polaridade do Quíntuplo

Finalmente, para completar a informação do quíntuplo é necessário extrair a polaridade de cada quíntuplo. É através do texto relevante de cada quíntuplo extraído na fase anterior

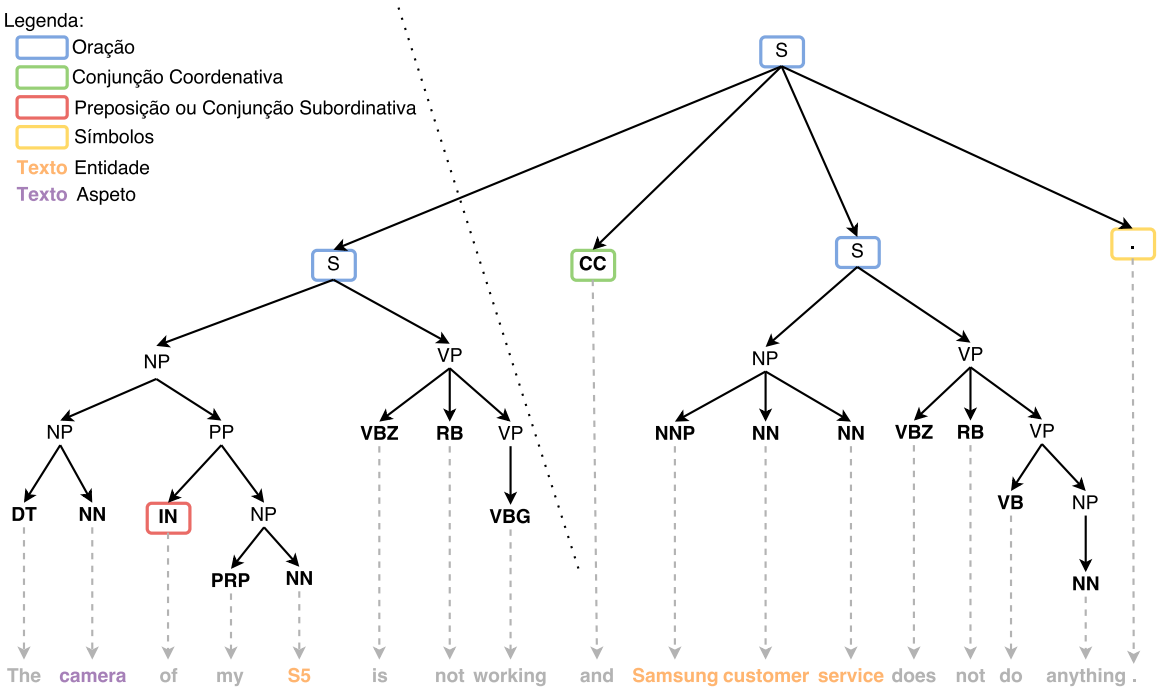

Figura 6.14: Exemplo da aplicação das regras para extrair o texto relevante a cada quíntuplo.

que a polaridade é calculado. Para tal é usada a ferramenta de extração de polaridade desenvolvida e descrita na secção 6.6.1. A essa ferramenta é alimentado o texto relevante de cada quíntuplo e esta devolve a polaridade correspondente. No final obtêm-se toda a informação dos quíntuplos como por exemplo:

Quíntuplo #1 = (Sara, 18/6/2015, *Iphone*, *camera*, Polaridade Negativa)

Quíntuplo #2 = (Sara, 18/6/2015, *Samsung customer service*, GERAL, Polaridade Negativa)

6.7 Ferramenta de Extração de Opiniões para o Português

Um outro requisito necessário para este projeto era a construção de uma ferramenta de extração de opiniões para a Língua Portuguesa. Tal como explicado, para a mesma ferramenta associada à Língua Inglesa, pretende-se extrair informações como o autor, data, aspetos, entidades e polaridade associada ao texto e aos quíntuplos.

Nesta secção são apresentados os detalhes do desenvolvimento da ferramenta de extração de opiniões, que é dividida em quatro ferramentas principais: ferramenta de extração de polaridade, de extração de aspetos e entidades e ferramenta de extração de quíntuplos.

É de salientar que as abordagens desenvolvidas assemelham-se às abordagens implementadas para a ferramenta de extração de opiniões para a Língua Inglesa, como tal, esta secção apenas irá descrever de forma sucinta, as principais diferenças entre ambas as ferramentas.

6.7.1 Extração de Polaridade

A ferramenta de extração de polaridade pretende classificar texto escrito em Português e extraído das redes sociais, como sendo de opinião positiva, negativa ou neutra.

Tal como para a mesma ferramenta para a Língua Inglesa a abordagem usada é supervisionada, usando as Máquinas de Vetores de Suporte. A Figura 6.5, já apresentada na secção 6.6.1, também se aplica a esta ferramenta. Ou seja, inicialmente foi criado um corpus que serviu de treino para o modelo criado. Para cada instância desse corpus são extraídas um conjunto de características que descrevem os textos. Todas esses dados são enviados por API *REST* para a biblioteca *Scikit-Learn*, que treina o modelo de polaridade usando as Máquinas de Vetores de Suporte.

Nesta secção é descrito com algum detalhe o desenvolvimento desta ferramenta, começando pela descrição do corpus de treino usado.

Corpus

O corpus de treino foi desenvolvido por vários elementos da equipa da Wizdee. De forma a uniformizar as anotações foi elaborado um conjunto de regras e exemplos do que se pretendia, para que todos os elementos pudessem anotar a polaridade seguindo o mesmo raciocínio.

Na tabela 6.14 é possível observar a distribuição do corpus entre as diferentes classes. O corpus de treino contém 9169 casos, sendo que 13% desses casos possuem polaridade positiva, 46% são casos neutros e 41% são casos negativos.

Nome	Total	Positivo	Negativo	Neutro
Corpus de Treino	9169	1259	3628	4282

Tabela 6.14: Distribuição do corpus de treino pelas três classes de polaridade existentes.

Extração de Características

De forma a classificar o texto nas diferentes classes de polaridade é necessário uma fase que faz o processamento do texto e o transforma num vetor de características. Cada frase é transformada num vetor de 352 características.

Todas as características desenvolvidas foram criadas usando várias ferramentas e léxicos de polaridade e, tal como para a língua inglesa, podem-se dividir em dois grupos: características de conteúdo e de léxico.

As características de conteúdo, uma vez que são as mais básicas, são muito semelhantes às já desenvolvidas e descritas para a ferramenta de extração de polaridade para a língua inglesa, e por isso não vão ser detalhadas nesta secção, uma vez que já foram referidas na secção A

Para as características de léxico, ou seja características que são dependentes dos léxicos de polaridade, foram usados três diferentes léxicos: *SentiLex-PT*, *Léxico ReLi* e *Lista de Polaridades*. Cada um destes léxicos está descrito em detalhe na secção 2.5.2.

Tal como para a Língua Inglesa, para a extração das características de léxico o texto foi processado de três diferentes formas, como já descrito na secção A. No entanto, esta ferramenta apresenta uma diferença no processamento que pretende obter o texto da forma mais correta possível (sem calões, *urls*, etc). Ou seja, para além de todo o processamento feito, o texto passa pelo corretor ortográfico, onde são corrigidas todas as palavras que não estão escritas de forma correta. Esta necessidade deveu-se ao facto de se observar uma maior presença de erros ortográficos presentes no texto escrito em português.

Para cada um desses léxicos usados foi construído um conjunto de características também muito semelhante à ferramenta de extração de polaridade para o inglês, que por isso não serão de novo detalhadas (ver detalhes em A).

6.7.2 Extração de Aspetos

A ferramenta de extração de aspetos tem como objetivo extrair de um texto características inerentes a uma marca, produto ou serviço, ou seja características associadas a uma entidade. Alguns exemplos dessas características são: "preço", "atendimento", "ecrã", "tarifário", entre outras.

O desenvolvimento desta ferramenta baseou-se na abordagem linguística já usada para o desenvolvimento da mesma ferramenta mas direcionada para a Língua Inglesa, que está representada na Figura 6.7 da secção 6.6.2.

Lista de Aspetos Pré-Selecionados

A primeira fase do desenvolvimento desta ferramenta passa pela criação manual de uma lista de aspetos que é usada nas fases posteriores para extrair outros aspetos menos comuns.

A abordagem foi a mesma que o para Inglês, ou seja inicialmente, são extraídos os substantivos de um conjunto de textos. Cada um desses substantivos é ordenado pelo número de vezes que aparecem nos textos e são revistos manualmente. No final, obteve-se uma lista de 40 aspetos que inclui, por exemplo, "serviço", "loja", "cliente", "preço", "passatempo", "telemóvel", "desconto", "cartão", "música", "fatura", entre outros.

Relações entre Aspetos e Palavras de Opinião

Uma vez que não basta retirar todos os substantivos dos textos para extrair os aspetos, pois nem todos os substantivos caracterizam uma entidade, foi necessário perceber quais seriam os substantivos que tinham um maior potencial em ser aspeto. Para tal analisou-se as relações entre os aspetos já conhecidos (a partir da lista criada na fase anterior) e as palavras de opinião, uma vez que nos interessa substantivos que muitas vezes são mencionados numa opinião.

Sendo assim, e usando o *parser* de dependências desenvolvido (ver secção 6.5), foram extraídas uma lista de relações de dependência entre os aspetos conhecidos e as palavras de opinião, tal como já exemplificado para a ferramenta de extração de aspetos para o inglês (ver Figura 6.8 na secção 6.6.2).

Extração dos Novos Aspetos

Usando a lista de relações de dependência obtidas na fase anterior, extraíram-se novos aspetos. Ou seja, através de todas as palavras de opinião encontradas na frase, procuram-se todos os substantivos que tenham uma dependência conhecida extraída na fase anterior. Todos os substantivos encontrados são assumidos como sendo aspetos candidatos.

Análise e Limpeza de Aspetos

Depois de se obter todos os aspetos candidatos foi observado que existiam muitos aspetos que não deviam ser considerados aspetos e como tal foi adicionada a fase de análise e limpeza. Esta fase tem como objetivo diminuir o número de aspetos candidatos, usando várias regras que se foram verificadas os vão excluindo. Antes do aspeto candidato ser submetido a análise, este é enviado ao corretor ortográfico onde apenas os erros associados à

falta de acentos são corrigidos. Este passo foi importante tendo em conta que foi constatado que muitos textos publicados nas redes sociais carecem da utilização correta dos acentos.

Para que um aspeto candidato seja excluído ele é positivo para pelo menos uma das seguintes regras:

- É uma palavra de opinião
- É uma *stopword*
- Não é uma palavra conhecida pelo Thesaurus Português.
- Não é uma palavra conhecida pelo corretor ortográfico.
- Se for uma palavra inglesa, é considerado um aspeto pela ferramenta de extração de aspetos para o Inglês.
- É uma palavra composta por apenas símbolos
- É uma palavra de referência temporal (ver dicionário em 6.4)
- É uma palavra, verbo ou adjetivo comum
- É uma palavra que contém números ou é uma referência numérica (ver dicionário em 6.4)
- É uma palavra que contém sufixos associados a verbos (como "veste-te") ou possui um modo infinitivo (como "vestir")
- É o nome de uma pessoa, cidade ou país
- Não é considerado um substantivo quando classificado individualmente pelo identificador de classes gramaticais

Algumas das regras adicionada pretendem diminuir alguns erros proporcionados pelas diversas ferramentas de base usadas, como o identificador de classe gramatical. Todos os aspetos candidatos que não pertençam a nenhuma das regras são adicionados à lista de aspetos final que é usada para extrair os aspetos de novas frases.

Processamento

Quando a partir de um frase se pretende extrair todos os aspetos é necessário fazer um pré processamento para que a ferramenta consiga extrair corretamente.

A fase de processamento é muito semelhante à fase de processamento já explicada para a ferramenta em inglês (ver secção 6.6.2), à exceção no início serem analisadas todas as abreviaturas ou calões do texto e substituídas pela sua forma correta. De seguida, é aplicado o processo de identificação de *tokens*, identificação das classes gramaticais, *lematização* e *chunking*. Para cada substantivo do texto, o seu lema é procurado na lista de aspetos extraída na fase anterior.

Adaptação a Novos Domínios

Tal como a ferramenta de extração de aspetos para a Língua Inglesa, esta ferramenta carece dos mesmos processos quando se pretende adaptar a um novo domínio. De forma a facilitar a introdução de um novo domínio, o processo está todo automatizado, sendo preciso apenas fornecer um conjunto de textos do domínio pretendido.

Neste momento, a ferramenta apenas se encontra adaptada ao domínio de telecomunicações e telemóveis.

6.7.3 Extração de Entidades

Para as ferramentas de extração de opiniões é também necessário extrair as entidades que podem ser marcas, produtos ou serviços. A abordagem desenvolvida é semelhante à já apresentada para a ferramenta de extração de entidades para a Língua Inglesa.

Na Figura 6.9 da secção 6.6.3 estão representadas todas as fases de extração de entidades. Tal como para os aspetos, inicialmente é construída manualmente uma lista de entidades que permite, posteriormente, extrair as relações de dependência associadas geralmente a entidades. Essas relações são usadas para extrair entidades candidatas de uma frase. Neste secção, são apresentados alguns detalhes de cada uma destas fases.

Lista de Entidades Pré-Selecionados

Tal como referido anteriormente, a primeira fase tem como objetivo a criação de uma lista de entidades frequentes num domínio. Foi criado um conjunto de textos que foram extraídos de várias páginas do Facebook como a página da Phone House Portugal[20], Samsung Portugal[21], PT Empresas[22] entre outras. A lista foi criada de forma manual, observando um conjunto de textos, e contém 49 entidades das quais, "*YouTube*", "*Fox*", "*RTP*", "*Sony*", "*SportTV*", "*Iphone*", entre outras

Extração de Relações entre Verbos e Entidades

Tal como realizado para a mesma ferramenta em inglês, usando a lista manual de entidades, foram extraídas as suas relações com os verbos, usando as àrvores de dependências. Basicamente, no mesmo conjunto de textos usado na fase anterior, para cada frase é encontrados todos os nós que correspondem a entidades e para cada um deles encontra-se o nó verbo mais perto. Todas as relações obtidas entre entidades e verbos são guardadas e utilizadas nas fases seguintes.

Na Figura 6.10 também apresentada na secção 6.6.3 é possível observar um exemplo da informação que se pretende extrair nesta fase.

Processamento

Para esta fase é aplicado o mesmo tipo de processamento aplicado para a ferramenta em inglês. Ou seja, para cada frase que se pretenda extrair as entidades, passa pelo processo de identificação dos *tokens*, identificação de classes gramaticais, *lematização* e *chunking*. Todos os *smiles*, *urls*, emails, *hashtags* e menções são removidas do texto.

Extração de Entidades Candidatas

Depois de processar o texto, são extraídas todas as possíveis entidades usando as relações de dependência conhecidas entre verbos e entidades.

Basicamente, todos os nós da árvore de dependências do texto que representam verbos são extraídos. A partir de cada um desses nós tenta-se replicar cada relação conhecida, ou seja, usando o nó verbo descobre-se se existe alguma relação conhecida entre esse verbo e um nó substantivo. Cada um desses novos nós são considerados entidades candidatas. Por exemplo, se conhecermos a relação verbo-entidade "*subj*" e se o nó verbo possuir uma ligação com outro nó através dessa dependência, esse nó é considerado uma entidade candidata.

[20]Disponível em `https://www.facebook.com/phonehouse.pt`
[21]Disponível em `https://www.facebook.com/samsungportugal`
[22]Disponível em `https://www.facebook.com/PTEmpresas`

No entanto, as relações não são suficiente para conseguir separar tudo o que é entidade do que não é, por isso é necessário a fase seguinte para excluir as entidades falso positivas.

Análise e Limpeza de Entidades

De forma a reduzir o número de entidades falso positivas, foram acrescentadas um conjunto de regras que caso alguma das entidades candidatas se enquadre em pelo menos uma, ela é excluída da lista de entidades da frase. O conjunto de regras usadas, semelhante à mesma ferramenta para o inglês, é o seguinte:

- Não é substantivo
- É uma palavra de opinião ou *stopword*
- É uma referência temporal
- É um substantivo, verbo ou adjetivo comum
- É considerado um aspeto
- Se for uma palavra inglesa, é considerado um aspeto pela ferramenta de extração de aspetos para o Inglês.
- É uma palavra composta por apenas símbolos
- É uma palavra que contém números ou é uma referência numérica (ver dicionário em 6.4)
- Palavra contém sufixos associados a verbos (como "veste-te") ou é encontrada o modo infinitivo (como "vestir")
- É o nome de uma pessoa, cidade ou país
- Não é considerado um substantivo quando classificado individualmente pelo identificador de classes gramaticais

É de notar que muitas vezes as entidades não só compostas por palavras em português e por isso não faz sentido o uso do corretor ortográfico como foi usado para a extração de aspetos em português.

Um outro ponto importante é que muitas vezes nos textos extraídos das redes sociais existem vários estrangeirismos, como por exemplo a palavra "*zoom*", por isso foi adicionada uma regra que caso a palavra esteja em inglês é comparada os todos os aspetos conhecidos em inglês e removida caso se verifique que é um aspeto.

Por fim, tal como para a ferramenta em inglês, todo o processo permite que cada cliente tenha uma lista de entidades-chave. Ou seja, para além das entidades extraídas automáticas as palavras incluídas nessa lista são também consideradas entidades sem passarem pelo processo de análise e limpeza.

Extração de Entidades Complexas

No fim da fase anterior obtêm-se um conjunto de entidades de um determinado texto, no entanto uma entidade pode não ser composta apenas por uma palavra, como na frase seguinte:

> "Boa tarde, tenho um **vodafone smart 3** com um problema muito irritante."

Neste exemplo, não se pretende apenas extrair a palavra "*vodafone*" ou "*smart*" como entidade individual mas sim a entidade complexa "*vodafone smart 3*".

Uma forma de obter essas entidades complexas seria usando através de um *Chunker*, no entanto a Wizdee apenas possui uma ferramenta de *Chunking* que se baseia na árvore de constituência. Como tal foi necessário construir um mecanismo que aproveita os *chunks* da árvore de constituência e recria novos *chunks*.

Na Figura 6.15 estão representados os vários passos para a extração de *chunks*. Inicialmente o texto é enviado para a ferramenta de *Chunker* que se baseia na Árvore de constituência que devolve um conjunto de *chunks*. Como se pode observar existem muitos chunks que são irrelevantes para esta tarefa, como tal foi construído um conjunto de regras que para além de excluir alguns *chunks* permite agrupar outros.

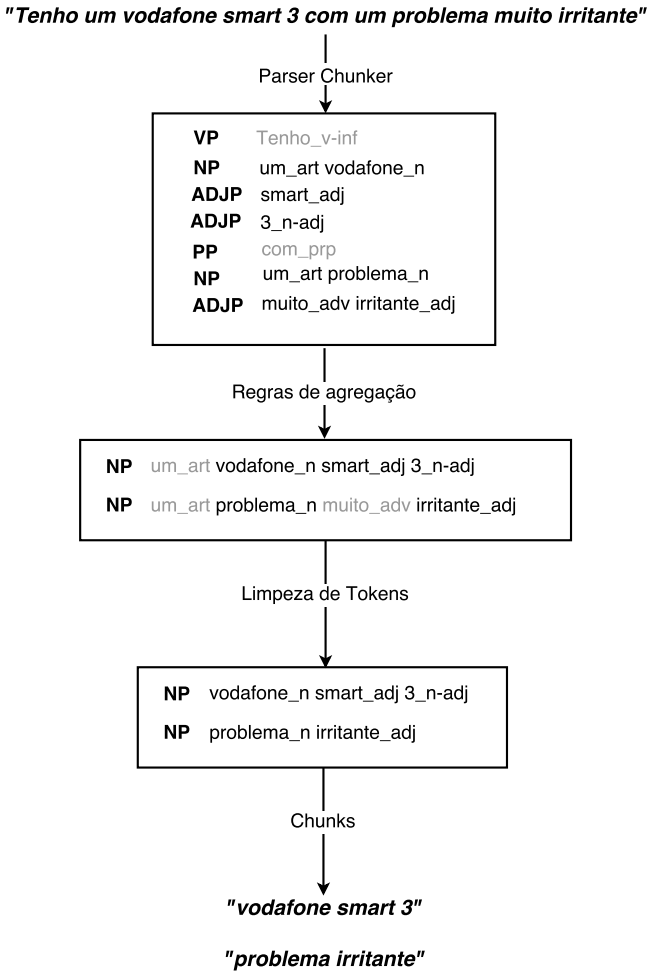

Figura 6.15: Exemplo de Extração de *Chunks* usando o *Parser Chunker*.

O conjunto de regras são as seguintes:

- Todos os *chunks* ADJP consecutivos são agrupados
- Todos os *chunks* NP consecutivos são agrupados
- Todos os *chunks* NP que são seguidos por *chunks* ADJP são agrupados

Todos os *chunks* que não sejam do tipo NP ou ADJP são excluídos. Sendo assim, no exemplo apresentado, excluímos o *chunk* VP e PP e agregamos todos os restantes em

dois. No entanto, como se pode observar na segunda caixa ainda existe palavras que se podem excluir dentro do mesmo *chunk*, por isso é seguido um processo de limpeza onde todos os artigos e advérbios são excluídos. Fica-se, assim, no fim com dois *chunks*. Se a entidade extraída pertencer a um desses chunks quer dizer que é uma entidade complexa. Por exemplo, se extrairmos a entidade "*vodafone*", uma vez que ela pertence a um *chunk* a entidade extraída passa a ser "*vodafone smart 3*".

6.7.4 Extração de Quíntuplos

Nesta secção é descrita em detalhe o desenvolvimento da ferramenta de extração de quíntuplos para a Língua Portuguesa. Tal como já foi referido anteriormente um quíntuplo é composto pelos cinco seguintes componentes: autor, data, entidade, aspeto e polaridade.

A abordagem desenvolvida é semelhante à já descrita para a ferramenta de extração de quíntuplos para a Língua Inglesa (ver diagrama em 6.12). Para relembrar, inicialmente são extraídos todos os aspetos de um dado texto e todas as entidades usando as ferramentas descritas nas secções 6.7.2 e 6.7.3 respetivamente. Depois é extraído a relação entre as entidades e aspetos, ou seja é a fase onde se percebe a que entidade um aspeto se refere. Por fim, e de forma a extrair a polaridade do quíntuplo é extraído o texto relevante, ou seja o texto que se refere a um aspeto ou entidade.

Relação Aspeto-Entidade

Depois de obter os aspetos e entidades é necessário perceber qual a ligação entre eles, de forma a conseguir agrupar os aspetos às entidades correspondentes. Na tabela 6.13 já apresentada para a ferramenta de Língua Inglesa estão representados todas diferentes situações que podem ocorrer nesta fase. Por exemplo, se o texto tiver apenas uma entidade e pelo menos um aspeto estes são considerados relacionados, ficando todos os aspetos como uma característica dessa entidade. Caso não tenha nenhum aspeto, as entidades encontradas são consideradas como estando a ser referidas de uma forma geral e por isso o aspeto passa a ser "GERAL".

No entanto, se o texto avaliado tiver mais do que uma entidade e pelo menos um aspeto é necessário uma análise mais detalhada para perceber que aspetos se referem a quais entidades. Para tal, foi desenvolvido um método que, usando as árvores de dependência, permite perceber quais são as relações mais fortes. Ou seja, para cada aspeto encontra-se qual a entidade cuja relação de dependência é mais forte (a relação tem menor distância) e essa entidade é assumida como sendo a entidade do aspeto em questão. Pode-se encontrar um exemplo desta abordagem na Figura 6.13.

Neste exemplo, no fim desta fase obtêm-se com os seguintes quíntuplos:

Quíntuplo #1 = (Catarina, 20/6/2015, *MEO*, tarifários, ?)
Quíntuplo #2 = (Catarina, 20/6/2015, *Vodafone*, GERAL, ?)

Extração do Texto Relevante

Outra tarefa importante é perceber que parte da frase se refere a cada quíntuplo para que a sua polaridade possa ser extraída. No exemplo anterior, tendo dois quíntuplos é preciso saber que parte da frase está a referir-se ao primeiro e ao segundo de forma a perceber se a opinião do autor é positiva ou negativa em relação à entidade MEO e à entidade Vodafone.

Tal como para a ferramenta em inglês, existem vários tipos de regras que são usadas para a divisão do texto pelos quíntuplos, que se baseiam na árvore de constituência da frase. As regras baseiam-se nas orações (nós do tipo fcl, icl, acl e cu), conjunções (palavras

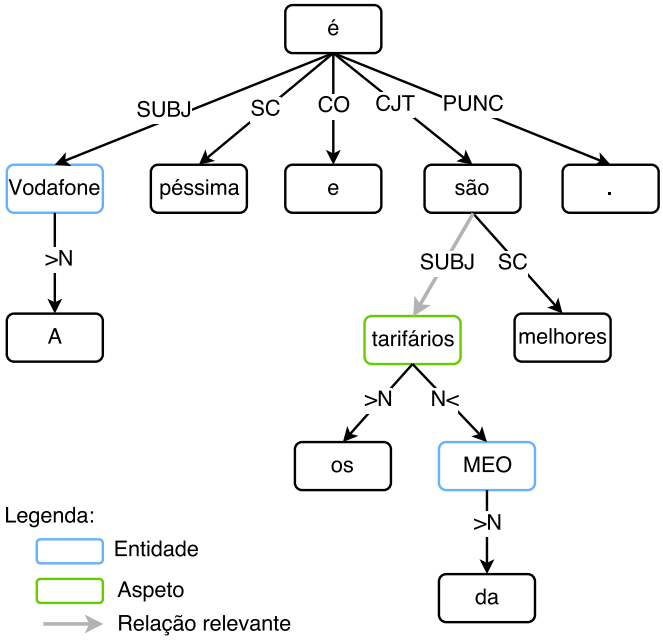

Figura 6.16: Exemplo de extração da relação entre entidades e aspetos para o Português.

como "e", "ou" e "mas") e símbolos. No caso de nenhuma dessas regras conseguir separar os quíntuplos, tenta-se criar um quíntuplo complexo recorrendo às preposições, como por exemplo usando a palavra "de", "neste", "na". No caso da não existência de preposições que satisfaçam é assumido que todo o texto pertence a todos os quíntuplos.

Um exemplo de extração de texto relevante está explicito na Figura 6.17.

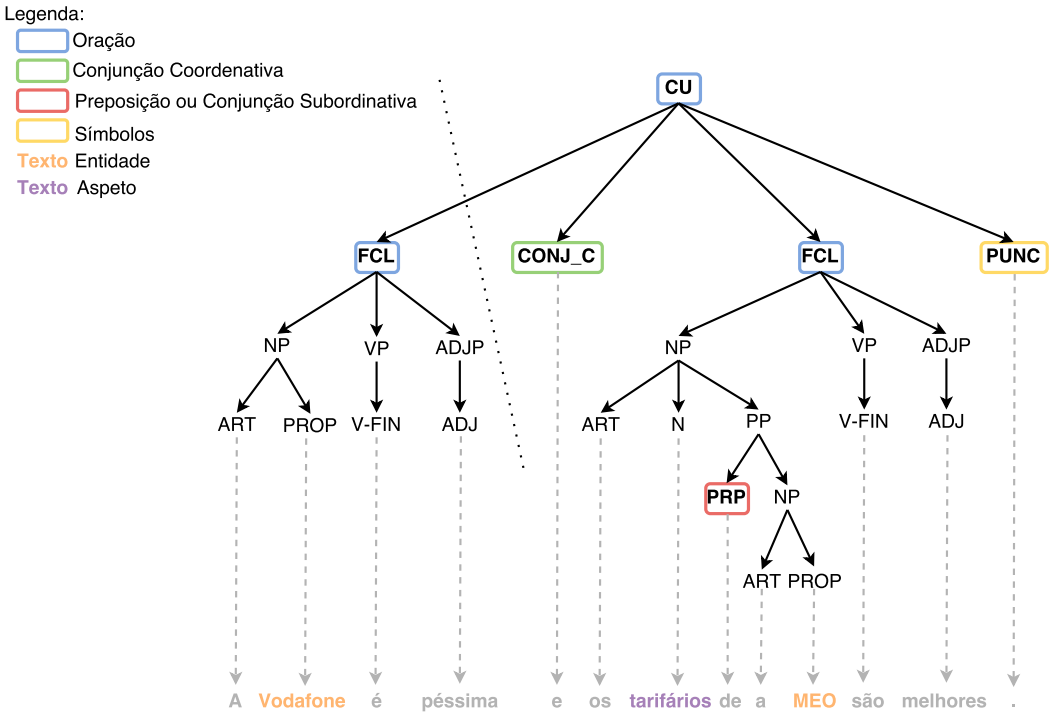

Figura 6.17: Exemplo da aplicação das regras para extrair o texto relevante de cada quíntuplo.

No exemplo, tem-se duas entidades, sendo que uma delas tem um aspeto associado, por isso o objetivo é conseguir isolar as duas entidades em frases diferentes sem perder muita informação. Nesse caso e recorrendo à primeira regra que permite separar a frase por orações, encontra-se três tipos de nós que o permitem. Os dois nós a azul que são do tipo FCL conseguem separar ambas as orações ficando cada uma com uma entidade. Como tal, o primeiro nó encontrado do tipo FCL determina por onde a divisão é feita. No caso de não existir nós de orações, recorreria-se à conjunção presente que neste caso fazia a mesma divisão.

Sendo assim desta fase e usando o exemplo conseguimos extrair a seguinte informação:

Quíntuplo #1 = (Catarina, 20/6/2015, *MEO*, tarifários, ?)
Texto Relevante = "*A Vodafone é péssima*"

Quíntuplo #2 = (Catarina, 20/6/2015, *Vodafone*, GERAL, ?)
Texto Relevante = "*e os tarifários da MEO são melhores.*"

Um outro exemplo está presente na Figura 6.18 que representa a árvore da frase: "Não gostei de não receber o bónus de 20euros no saldo!". Neste caso, são detetados apenas dois aspetos, "bónus" e "saldo" que pertencem cada um a quíntuplo. No entanto, os aspetos não são separáveis usando as três regras principais, por isso recorre-se às preposições que permitem a criação de aspetos complexos. Procuram-se todas as preposições representadas pelo nó PRP, e o nó que as separar é o nó escolhido.

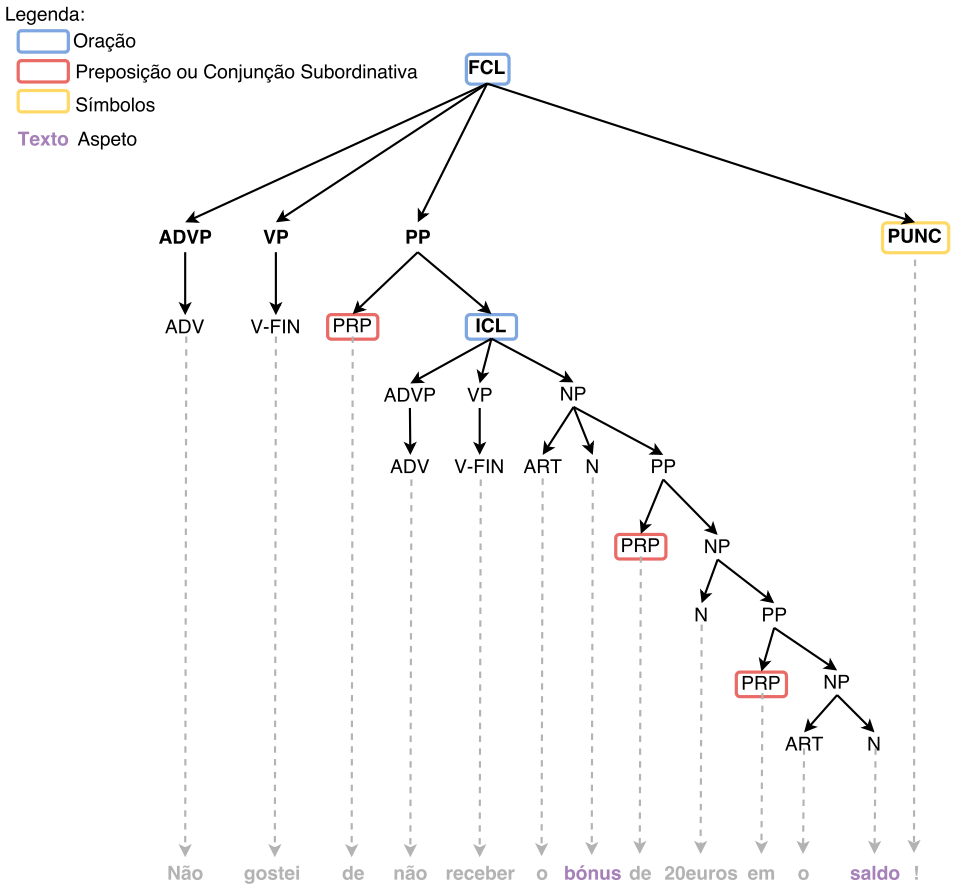

Figura 6.18: Exemplo da aplicação da regra que faz uso das preposições para extrair o texto relevante de cada quíntuplo.

Neste caso, a ferramenta agrupa ambos os quíntuplos e extraí o seguinte:

Quíntuplo #1 = (Silva, 20/6/2015, Desconhecido, "bónus no saldo", ?)
Texto Relevante = "*Não gostei de não receber o bónus de 20euros no saldo!*"

Extração da Polaridade do Quíntuplo

Por fim, usando o texto relevante atribuído a cada quíntuplo é extraída a polaridade do quíntuplo. O calculo da polaridade é feito usando a ferramenta desenvolvida para extração de polaridade da língua portuguesa descrita na secção 6.7.1.

Usando o mesmo exemplo apresentado em cima o resultado esperado é o seguinte:

Quíntuplo #1 = (Catarina, 20/6/2015, *MEO*, tarifários, Polaridade Positiva)

Quíntuplo #2 = (Catarina, 20/6/2015, *Vodafone*, GERAL, Polaridade Negative)

Capítulo 7

Testes

Este capítulo tem como objetivo apresentar e descrever os diferentes testes realizados para cada uma das ferramentas desenvolvidas. Serão apresentados os testes para as seguintes ferramentas:

- Parser de Dependências para o Português;
- Ferramenta de Extração de Aspetos para o Inglês e Português;
- Ferramenta de Extração de Entidades para o Inglês e Português;
- Ferramenta de Extração de Quíntuplos para o Inglês e Português.

É de notar que este tipo de ferramentas apresenta sempre uma dificuldade relacionada com a complexidade de cada língua e que o seu desenvolvimento nunca se dá por terminado, podendo sempre ser melhorado. No entanto, o mais importante é que neste momento a Wizdee já possui um conjunto de ferramentas que conseguem extrair o pretendido e que estão preparadas para serem usadas em projetos futuros.

Para cada uma das ferramentas, é descrito os tipos de testes realizados e as suas motivações, apresentando de seguida os resultados e a análise aos mesmos. Todos os resultados apresentados neste capítulo foram realizados na mesma máquina cujas especificações encontram-se na tabela 7.1.

Especificação	**Máquina de Teste**
Sistema Operativo	Ubuntu 14.04 (64 bits)
CPU	Intel® Core™i7-4770 @ 3.4GHz (8 Cores)
RAM	16GB DDR3
Disco Duro	Seagate Barracuda 1TB @ 7200 rpm

Tabela 7.1: Especificações da máquina de testes.

7.1 *Parser* de Dependências para a Língua Portuguesa

Nesta secção são apresentados todos os testes realizados para avaliação da ferramenta de parser de dependências que permite criar árvores de dependência para o Português descrita na secção 6.5.

7.1.1 Testes de Qualidade

Para a realização dos testes de qualidade foi necessário escolher um corpus anotado de forma a conseguir avaliar a qualidade do modelo criado. Como tal, foi usado o corpus de teste do Bosque[1] que possui 288 frases e 5867 palavras.

As métricas (Nilsson, 2014) usadas para apresentação dos resultados são as seguintes:

- **LAS** Métrica denominada como *BothRight* onde uma palavra é contada como certa se tanto o parâmetro *HEAD*[2] como o parâmetro *DEPREL*[3] forem iguais aos dados de teste.
- **LA** Métrica denominada como *LabelRight* onde para uma palavra ser contada como correta basta apenas o parâmetro *DEPREL* ser o mesmo que os dados de teste.
- **UAS** Métrica denominada como *HeadRight* onde para uma palavra ser contada como correta basta apenas o parâmetro *HEAD* ser o mesmo que os dados de teste.
- **Medida-F** Ver equação 2.7.

Resultados e Análise

Na Tabela 7.2 é possível visualizar os vários resultados obtidos nos 30 testes feitos. É de notar que nos 30 testes feitos o desvio padrão foi 0.0%, ou seja os resultados ao longo dos testes não variaram.

Métrica	**Exatidão**[4]
LA	91.5 %
LAS	83.4 %
UAS	86.8 %

Tabela 7.2: Exatidão do modelo segundo diferentes métricas.

Segundo os resultados obtidos pode-se concluir que o modelo é melhor a detetar qual o tipo de relação (91.5%) do que a hierarquia entre as palavras (86.8%).

Trabalho	**Corpus**	**Exactidão**
LX-DepParser[5]	Corpus próprio	LAS = 91.1%
(Buchholz and Marsi, 2006)	Bosque	LAS = 87.6%
(Zhang et al., 2014)	Bosque	UAS = 92.42%

Tabela 7.3: Resultados do *parsing* de dependências para cada um dos tipos de dependências.

Comparando os resultados obtidos com os resultados de outros sistemas de *parsing* de dependências (ver Tabela 7.3), o sistema criado apresenta uma exatidão mais baixa. Uma diferença significativa é observada quando o corpus de treino não é o mesmo como no caso

[1] A Linguateca disponibiliza tanto um corpus de treino com um de teste do Bosque.

[2] HEAD é um parâmetro que representa o id da palavra que é o pai da palavra em questão na árvore de dependências.

[3] DEPREL é um parâmetro que representa a relação de dependência como por exemplo a relação de sujeito.

[4] Em inglês, *Accuracy*.

do LX-DepParser. A diferença entre os resultados dos trabalhos que usam o mesmo corpus é explicável pelo facto de que ao corpus de treino ter sido removidas algumas características das palavras que podem ser úteis para a qualidade do modelo. O terceiro trabalho embora use o mesmo corpus, usa uma ferramenta para treino diferente: o *MSTParser*[6].

Na Tabela 7.4 é possível visualizar os vários resultados obtidos para cada tipo de dependência[7]. Tal como na tabela anterior, os resultados não variaram ao longo dos testes, sendo o desvio padrão 0.0%. Em vermelho estão representadas as dependências que no treino estavam pouco presentes (menos de 0.5%) com menos de 1028 casos. Nas primeiras duas colunas é possível observar a distribuição do número de casos presentes tanto no treino como no teste.

Apesar da média ser 71.7%, existem vários tipos de dependência que apresentam uma Medida-F bastante baixa. No entanto, maior parte destes casos deve-se à pouca presença desses tipos de dependência no corpus de treino. Na tabela estão marcados a vermelho todos os tipos de dependência que tem menos de 0.5% de presença no corpus, o que é um valor muito baixo para conseguir obter depois bons resultados. Se ignoramos todas essas situações fica-se com uma média de 86.7% e desvio padrão de 10.3% sendo que o pior resultado passa a ser o tipo *PIV* com cerca de 67.5% de Medida-F.

A distribuição dos tipos no corpus de treino não é equilibrada, como se pode ver pela média e desvio padrão (tendo um desvio padrão maior que a própria média). Por exemplo, tanto a dependência *>N* e a *PUNC* são tipos que estão muito presentes no corpus de treino e tem muitos casos no teste apresentam uma Medida-F maior do que 98%.

7.1.2 Teste de Performance

Foram feitos dois tipos de teste de performance. O primeiro contabiliza o tempo que demora a criação do modelo. De forma a obter resultados estatisticamente relevantes o teste foi repetido 30 vezes.

O segundo teste contabiliza o tempo necessário para a execução desse modelo. Para este teste foi organizado um conjunto de 4660 frases de tamanhos variáveis que foram alimentadas uma a uma ao *parser*. Esse corpus é essencialmente constituído por frases extraídas da rede social *Facebook*, mais propriamente, da página da Vodafone Portugal[8].

Resultados e Análise

Na Tabela 7.5 estão explícitos os resultados obtidos tanto na parte de otimização de parâmetros, como na fase de criação do modelo final que envolve o treino com os parâmetros encontrados na fase anterior.

Como se pode verificar a fase que mais tempo consome é a de otimização de parâmetros. Essa fase dependente tanto do número de características associadas a cada palavra e também do tamanho do corpus. Já a fase de treino do modelo, ou seja depois de já saber quais as melhores definições, demora apenas 1.4 minutos, em média, o que é um tempo aceitável tendo em conta que o modelo não é criado em tempo real, ou seja criasse uma primeira vez e não precisa de ser retreinado sempre que for necessário usá-lo. A Figura 7.1 mostra a relação entre o tamanho da frase (número de palavras) e o tempo que demora a criação da sua árvore de dependências.

Um dos pontos mais relevantes em termos de performance é perceber quanto tempo demora a aplicação do modelo a uma determinada frase. Segundo os resultados obtidos é

[6]Disponível em `http://www.seas.upenn.edu/~strctlrn/MSTParser/MSTParser.html`

[7]Ver listagem e significado de cada tipo de dependência em `http://beta.visl.sdu.dk/visl/pt/symbolset-floresta.html`

[8]Página em `https://www.facebook.com/vodafonePT`.

Dependência	# Treino	# Teste	Precisão (%)	Abrangência (%)	Medida-F (%)
>A	1690	42	65.9	69.0	67.4
>N	37868	1043	98.9	98.9	98.9
>P	103	10	50.0	20.0	28.6
A<	1248	29	67.6	86.2	75.8
ACC	11077	362	88.6	91.1	89.9
ADVL	15669	453	77.2	79.6	78.4
ADVO	159	8	50.0	12.5	20.0
ADVS	805	20	40.9	45.0	42.9
APP	763	25	69.2	72.0	70.6
CJT	6430	169	82.9	78.9	80.9
CO	5055	145	100.0	98.4	99.2
DAT	227	5	66.7	80.0	72.7
FOC	201	4	100.0	75.0	85.7
KOMP<	249	3	100.0	33.3	50.0
MV	3008	85	90.9	96.4	93.6
N<	26151	712	92.2	93.4	92.8
N<PRED	3709	139	72.0	68.3	70.1
OC	344	13	33.3	37.5	35.3
P<	30894	887	97.8	98.3	98.0
PASS	717	19	89.5	100	94.4
PIV	2661	87	73.9	62.2	67.5
PRED	292	150	85.7	54.5	66.7
PRT-AUX<	620	10	75.0	90.0	81.8
PUNC	29097	858	100.0	99.9	99.9
QUE	176	6	50.0	16.7	25.0
S<	91	5	33.3	20.0	25.0
SC	3218	77	73.3	81.8	77.3
STA	7844	251	92.0	92.4	92.2
SUB	2021	50	90.6	96.0	93.2
SUBJ	12089	379	87.8	85.8	86.8
UTT	1174	41	75.0	80.5	77.6
Média	**6633.8**	**196.3**	**76.6**	**70.9**	**71.7**
Desvio Padrão	**10239.3**	**288.8**	**19.5**	**27.1**	**23.8**

Tabela 7.4: Resultados do *parsing* de dependências para cada um dos tipos de dependências.

possível verificar que, como esperado, quanto maior a frase mais tempo é necessário para a criação da árvore, uma vez que envolve mais processamento. No entanto, uma frase com cerca de 160 a 180 palavras apenas demora 20 milissegundos o que é um número bastante baixo. De notar que no corpus usado, cujas frases não foram pré-selecionadas, o tipo de frase mais frequente tem cerca 1 a 20 palavras (63% de presença no corpus).

Fase	Média (minutos)	Desvio Padrão (minutos)
Otimização de Parâmetros	56.8	4.7
Treino modelo final	1.4	0.07

Tabela 7.5: Tempo necessário para as diferentes fases de construção do modelo.

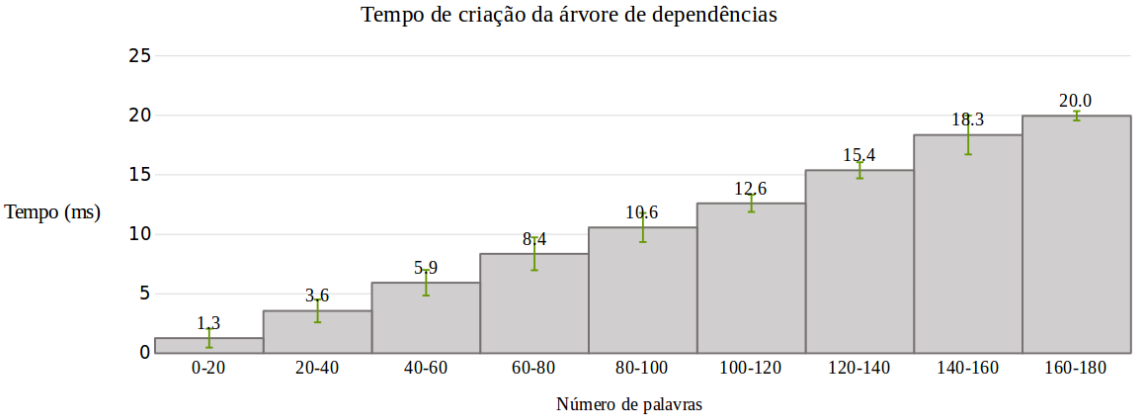

Figura 7.1: Tempo de criação de uma árvore de dependências de acordo com o tamanho da frase.

7.2 Ferramenta de Extração de Opiniões para o Inglês

Nesta secção são descritos todos os testes realizados, e apresentados e analisados os resultados obtidos para as diferentes ferramentas de extração de opiniões para a Língua Inglesa.

7.2.1 Extração de Polaridade

De forma a avaliar a ferramenta de extração de polaridade para a Língua Inglesa foram realizados vários testes que se podem dividir em dois diferentes conjuntos: testes de qualidade e testes de performance. Nesta secção cada um desses grupos de teste são descritos e os seus resultados demonstrados e analisados.

Testes de Qualidade

De forma a analisar a qualidade desta ferramenta foram feitos três testes diferentes. O primeiro teste tem como objetivo perceber que tipo de *kernel* (ver descrição na secção 2.2.2) é mais adequado ao problema que se pretende resolver.

Usando o *kernel* que melhor resultado apresentou foram feitas um conjunto de testes cujo objetivo é perceber quais são as características mais importantes. Neste teste diferentes grupos de características são testadas e confrontadas com os seus resultados. Por fim, o último teste tem como objetivo comparar a ferramenta desenvolvida a outras ferramentas semelhantes já existentes.

A qualidade foi medida através das métricas usadas para este tipo de ferramentas (Nakov et al., 2014) que estão representadas nas equações 7.1, onde VP_{pos} representa as instâncias que foram classificadas de forma correta como sendo de polaridade positiva,

FP_{pos} representa as instâncias que foram classificadas como sendo de polaridade positiva mas na realidade não o são e FN_{pos} representa as instâncias que deviam ter sido classificadas como de polaridade positiva no entanto não foram. RE

$$\text{Medida-F}_{pos} = \frac{2 \times P_{pos} \times R_{pos}}{P_{pos} + R_{pos}} \tag{7.1a}$$

$$P_{pos} = \text{Precisão}_{pos} = \frac{VP_{pos}}{VP_{pos} + FP_{pos}} \tag{7.1b}$$

$$R_{pos} = \text{Abrangência}_{pos} = \frac{VP_{pos}}{VP_{pos} + FN_{pos}} \tag{7.1c}$$

Tanto para a classe positiva e negativa são calculadas tanto a precisão, a abrangência e a Medida-F. O modelo final é avaliado usando as duas Medida-F obtidas tanto na polaridade positiva como negativa, como representado na equação 7.2.

$$\text{Medida-F} = \frac{\text{Medida-F}_{pos} + \text{Medida-F}_{neg}}{2} \tag{7.2}$$

É de notar que durante o treino de todos os modelos é usado a validação de 10-folds descrita em 2.2.3. Uma vez que a Máquina de Vetores de Suporte é um algoritmo que dado o mesmo corpus de treino produz sempre o mesmo modelo, ou seja encontra sempre o máximo global, nesta fase apenas foi necessário repetir os testes uma vez.

Para realizar os diferentes testes foi usado um corpus de teste também disponibilizado pelos organizadores do *Workshop on Semantic Evaluation* (Nakov et al., 2013). Tal como o corpus de treino, este contém textos extraídos da rede social *Twitter*. Na Figura 7.2 é possível observar a distribuição do número de instâncias por cada uma das três classes. O corpus contém exatamente 3196 instâncias em que 15% pertencem à classe de polaridade negativa e 44% pertence à classe de polaridade neutra.

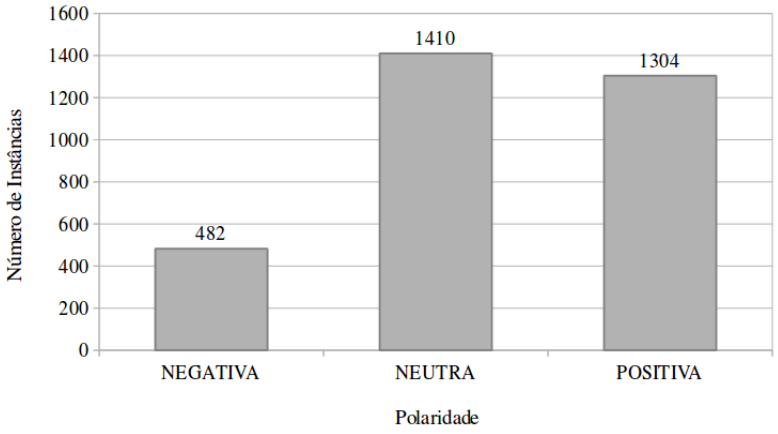

Figura 7.2: Distribuição das instâncias por cada tipo de polaridade no corpus de teste.

Por fim, foi realizado um teste que permite comparar a ferramenta desenvolvida com os resultados obtidos no SemEval-2014. Para esse teste foi usado o corpus de avaliação fornecido também pelos organizadores do *Workshop on Semantic Evaluation*. Esse corpus está dividido em 5 conjuntos de textos, cuja distribuição está representada na Figura 7.3.

Resultados e Análise

Uma das primeiras decisões que se encara quando se escolhe fazer um modelo usando o algoritmo de Máquinas Vetor de Suporte é perceber que *kernel* devemos usar. Na tabela

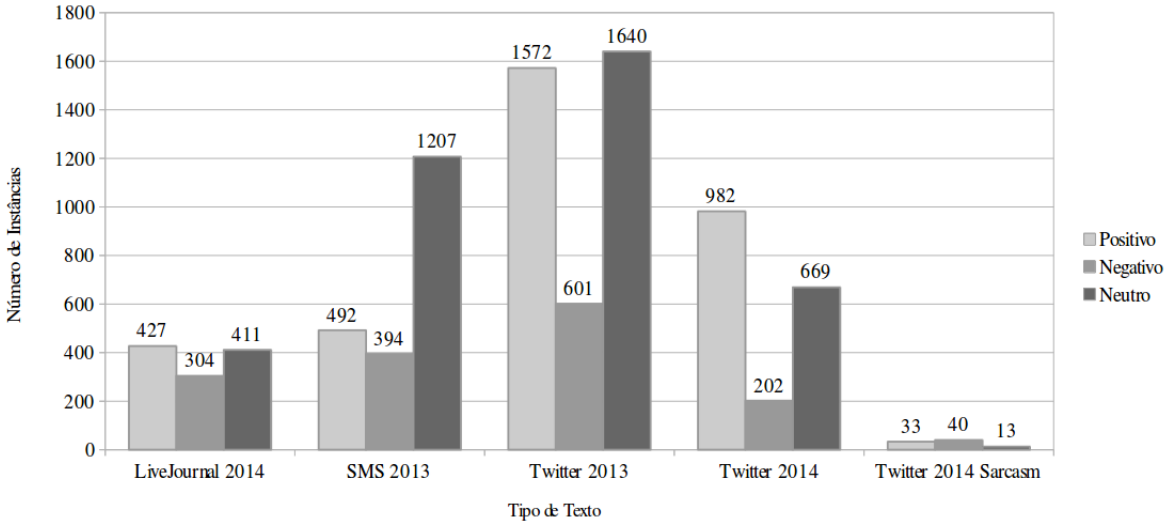

Figura 7.3: Distribuição das instâncias por cada tipo de polaridade nos diferentes tipo de texto do corpus de avaliação do SemEval-2014.

7.6 é possível visualizar os resultados obtidos nos testes usando diferentes *kernels*. Como se pode ver, o *kernel* que melhores resultados obtém é o *RBF*, tanto para a polaridade positiva como negativa. Tanto o polinomial como o linear têm uma diferença de cerca de 2% do melhor resultado obtido. No entanto, o *kernel* com maior consistência na Medida-F é o polinomial que apresenta um desvio padrão de 4.8%. Sendo assim, dado os resultados obtidos na Medida-F, foi escolhido o *kernel RBF*. É de notar que para a obtenção destes resultados foi feito para cada um dos *kernels* uma otimização de parâmetros usando *grid-search*.

Métricas	**Kernel RBF**	**Kernel Linear**	**Kernel Polinomial**
Medida-F Positiva (%)	72.1	70.4	69.6
Medida-F Negativa (%)	61.0	58.2	59.9
Média (%)	**66.5**	**64.3**	**64.7**
Desvio Padrão (%)	**5.5**	**6.0**	**4.8**

Tabela 7.6: Resultados dos testes a diferentes *kernels* das Máquinas Vetor de Suporte.

No segundo teste realizado, tentou-se perceber quais as características que mais influenciavam o sistema. De forma a perceber isso foi realizado o seguinte conjunto de testes relacionados, apenas, com as características de conteúdo, cujos resultados estão apresentados na tabela 7.7 :

- **Teste A** Para este teste apenas características que representem métricas de texto, *stopwords* e negações são tidas em conta.
- **Teste B** Incluindo também as características dos teste anterior, neste teste são acrescentadas as características associadas a palavras maiúsculas e repetição de letras, às *hashtags*, menções e *urls*.
- **Teste C** Neste teste foram adicionadas características associadas a *smiles*.
- **Teste D** Para além de todas as características dos testes anteriores foram adicionadas características associadas às classes gramaticais e pontuação.

- **Teste E** A este teste foram acrescentadas informações relacionadas com os dicionários temáticos.
- **Teste F** Este teste inclui todas as características de conteúdo, ou seja às anteriores foram acrescentadas as características associadas a calões, expressões idiomáticas e referências temporais.

Métricas	Teste A	Teste B	Teste C	Teste D	Teste E	Teste F
Medida-F Positiva (%)	51.2	50.0	49.3	49.4	51.1	61.9
Medida-F Negativa (%)	34.3	34.5	37.2	41.3	42.3	43.9
Média (%)	**42.8**	**42.3**	**43.3**	**45.4**	**46.7**	**52.9**
Desvio Padrão (%)	**8.4**	**7.7**	**6.0**	**4.0**	**4.3**	**8.9**

Tabela 7.7: Testes a diferentes conjuntos de características de conteúdo para a ferramenta de extração de polaridade para a Língua Inglesa.

Como se pode observar, à medida que se acrescenta novas características a ferramenta também melhora, à exceção da Medida-F Positiva. É possível observar que usando apenas características básicas como, *stopwords*, negações ou métricas do texto obtemos desde logo uma Medida-F de 42.8%. A maior melhoria, com uma acréscimo de 6.2 pontos percentuais, apresenta-se entre o teste E e F, onde se acrescenta informações sobre os calões, expressões idiomáticas e referências temporais. As informações sobre as classes gramaticais e informações sobre a pontuação também proporcionam uma melhoria ao sistema de 2.1 pontos percentuais. A diferença mais negativa apresenta-se entre os testes A e B cuja Medida-F diminui 0.5 pontos percentuais, consequente da Medida-F Positiva ter diminuído 1.2 pontos percentuais.

Mais relacionado com as características de léxico, foi realizado um outro conjunto de testes:

- **Teste F** Teste que inclui todas as características de conteúdo.
- **Teste G** Às características de conteúdo foram acrescentadas as características referentes aos *word embeddings*.
- **Teste H** Neste teste, foram as adicionadas as caraterísticas que usam o dicionário de léxico *AFFIN*.
- **Teste I** Neste teste, foram acrescentadas todas as caraterísticas associadas ao dicionário de léxico *Bing Liu*.
- **Teste J** Neste teste, foram acrescentadas as características associadas ao dicionário de léxico *NRC Emotion*.
- **Teste K** Neste teste, foi adicionado o último léxico usado, o léxico criado de forma automática. Neste teste estão incluídas todas as características desenvolvidas.
- **Teste L** Por fim, ao teste anterior foram retiradas todas as características que usam o texto pré-processado (referido em A) que envolveu a remoção de todos as palavras específicas das redes sociais, calões e expressões idiomáticas.

É de salientar que estes testes incluem todas as características de conteúdo. Na tabela 7.8 é possível observar os resultados obtidos.

Métricas	Teste F	Teste G	Teste H	Teste I	Teste J	Teste K	Teste L
Medida-F Positiva (%)	61.9	61.7	70.3	70.9	71.4	72.1	71.8
Medida-F Negativa (%)	43.9	44.5	58.2	59.3	59.5	61.0	60.2
Média (%)	**52.9**	**53.1**	**64.2**	**65.1**	**65.4**	**66.5**	**66.0**
Desvio Padrão (%)	**8.9**	**8.6**	**6.0**	**5.7**	**5.9**	**5.5**	**5.8**

Tabela 7.8: Testes a diferentes conjuntos de características de conteúdo e de léxico para a ferramenta de extração de polaridade para a Língua Inglesa.

Como se pode observar, todos os testes apresentam uma melhoria na ferramenta, sendo que a mais acentuada acontece quando se acrescenta o primeiro dicionário de léxico no teste H, cuja melhoria é de 11.1 pontos percentuais.

Outra situação interessante de salientar é o facto de quando se acrescenta as informações sobre o léxico que foi criado de forma automática obtém-se uma melhoria de 1.1 pontos percentuais que é maior do que quando se acrescenta tanto o léxico de *Bing Liu* e *NRC Emotion*. Isto pode se dever ao facto de todos esses três léxicos (Bing Liu, *NRC Emotion* e *AFFIN*) possuírem informações muito semelhantes o que faz com que o modelo não aprenda nada de novo, ao contrário do dicionário construído de forma automática que não se baseia apenas em palavras de opinião básicas, como "*good*", "*bad*".

Com o teste L pretendeu-se perceber se o processamento e remoção de palavras especiais era importante para a ferramenta, e como se pode observar se retirarmos todas essas características o sistema piora em 0.5 pontos percentuais.

O conjunto de características que proporciona melhores resultados é o simulado no teste K, que junta tanto as características de conteúdo como as de léxico, cujos resultados podem ser visualizados em mais detalhe na tabela 7.9.

Total de Instâncias	3196
Precisão Negativa	55.8%
Precisão Positiva	78.9%
Abrangência Negativa	67.2%
Abrangência Positiva	66.4%
Medida-F Negativa	61.0%
Medida-F Positiva	72.1%
Medida-F Final	66.5% ($\sigma = 5.5\%$)

Tabela 7.9: Resultados do modelo final para a ferramenta de extração de polaridade para Inglês.

É possível observar que o sistema tem mais dificuldades em detetar quais são os textos negativos, ou seja comparando a polaridade negativa com a positiva os textos que são anotados como negativos tem um erro maior do que os textos de polaridade positiva. Esta realidade pode dever-se ao facto de que no corpus de treino a polaridade negativa é a que está menos representada (apenas 18 % dos casos de treino).

Tal como já mencionado anteriormente, esta é uma tarefa com um grau de subjetividade muito grande, ou seja o que para algumas pessoas é positivo para outras pode ser considerado neutro, por exemplo. A subjetividade pode-se encontrar também no corpus usado para teste, como por exemplo na frase em baixo:

"Dear @WickedNemesis, I need you to come to Neb in Feb, so I have someone to go to the outdoors hockey game with me. -Ninja"

Nesta frase o corpus anota-a como uma frase negativa, no entanto a ferramenta desenvolvida anota-a como sendo uma frase neutra, o que se devia considerar como uma resposta admissível.

Este texto é considerado positivo pela ferramenta, no entanto o corpus anota-o como sendo negativo. No entanto, este texto também pode ser considerado positivo.

A polaridade negativa apresenta uma série de desafios, sendo que um deles é a deteção de ironia que muitas vezes é usada para expressar uma opinião negativa. Por exemplo, a frase em baixo pode ser considerada como irónica, no entanto essa definição não é óbvia. Ou seja, algumas pessoas consideram a frase como sendo positiva e outras pessoas interpretam como sendo uma frase irónica e por isso classificam-na como sendo negativa.

"LOOL, Samsung will be launching Galaxy Note II in Canada on 30th, Psy Gangnam will be performing at the launch."

Uma forma de avaliar a qualidade da ferramenta é compará-las com outras também semelhantes. Na tabela 7.10 é possível ver o resultado obtido pela ferramenta desenvolvida e pelo sistema desenvolvido pelo grupo TeamX que ficou em primeiro lugar no SemEval-2014 na tarefa 9 (Análise de Sentimento para o Twitter) (Nakov et al., 2014).

Ferramenta	LiveJournal 2014	SMS 2013	Twitter 2013	Twitter 2014	Twitter 2014 Sarcasm	Média
Team X	69.44 %	57.36 %	72.12 %	70.96 %	56.5 %	65.27% ($\sigma = 7.7\%$)
Wizdee	70.73 %	61.73 %	65.39 %	63.95 %	53.77 %	63.11% ($\sigma = 6.2\%$)
Diferença	**1.29**	**4.37**	**-6.73**	**-7.01**	**-2.73**	**-2.16**

Tabela 7.10: Comparação entre a ferramenta desenvolvida e o melhor sistema no SemEval-2014.

Como é possível observar a ferramenta desenvolvida consegue superar a melhor equipa do SemEval-2014 em dois corpus: LiveJournal 2014 e SMS 2013, sendo que a maior diferença regista-se no última com 4.37 pontos percentuais. Já nos outros três corpus a ferramenta obtém resultados abaixo da equipa TeamX, sendo que os mais acentuados registam-se nos corpus Twitter 2013 e Twitter 2014. No entanto, tendo em conta que existe uma grande restrição nos recursos que se podem usar num contexto comercial, pode-se concluir que o sistema consegue competir com os melhores resultados obtidos recentemente nesta área.

Teste de Performance

De forma a analisar a performance da ferramenta foi feito um teste que calcula o tempo necessário para as várias etapas da ferramenta. O teste foi repetido 30 vezes.

Resultados e Análise

Na tabela 7.11 são apresentados os tempos necessários para as diferentes fases. A extração de características para cada texto demora em média 18.44 segundos. É de notar que a primeira extração feita na ferramenta demora cerca de 6 minutos, uma vez que, necessita de carregar todos os dicionários, e inicializar todas ferramentas necessárias. No entanto,

este processo não é repetido. Já o treino do modelo demora cerca de 7 minutos, no entanto este número não é preocupante uma vez que a fase de treino não é necessária para o uso da ferramenta pois o modelo já está criado.

Adicionando o tempo de extração de características e o tempo necessário para extrair a polaridade usando o modelo fica-se com 18.47 segundos, sendo que a fase que mais consome tempo é a de extração de características.

Fase	**Média (segundos)**	**Desvio Padrão (segundos)**
Extração de Características	18.44	1.92
Treino do modelo final	443.47	9.41
Teste do modelo (1 instância)	0.034	0.004

Tabela 7.11: Resultados do teste de performance realizado para a ferramenta de extração de polaridade.

7.2.2 Extração de Aspetos

Para a avaliar a ferramenta de extração de aspetos para a Língua Inglesa foram realizados diversos testes que se podem dividir em dois grupos: testes de qualidade e performance. Nesta secção é descrito a especificação de cada teste e são apresentados os resultados e a sua análise.

Testes de Qualidade

De forma a analisar a qualidade da ferramenta desenvolvida foram feitos dois testes de qualidade. O primeiro teste pretende perceber qual as melhores relações de dependência que devem ser usadas para a extração de aspetos (fase descrita em 6.6.2). Ou seja, nessa fase é recolhida uma lista de relações que vão ser usadas na extração de aspetos e esse teste analisa quais são as relações que melhores resultados apresentam. As melhores relações são usadas nas fases seguintes. O segundo teste pretende perceber qual o melhor conjunto de restrições que se deve usar na fase de limpeza de aspetos descrita na secção 6.6.2. Ambos os testes foram repetidos 30 vezes.

A qualidade foi medida usando diferentes métricas comuns neste tipo de ferramentas como a Medida-F (Nakov et al., 2014) (ver equações 7.3), onde S representa os aspetos extraídos e G os aspetos anotados no corpus. Também foi calculado o Erro de Não reconhecimento[9] (ver equação 7.4) e o Erro de Demasiado Reconhecimento[10] (ver equação 7.5) (V. and A., 2015), onde VP representa verdadeiro positivos, FN é falso negativos e FP é falso positivos.

$$\text{Medida-F} = \frac{2 \times P \times R}{P + R} \tag{7.3a}$$

$$P = \text{Precisão} = \frac{|S \bigcap G|}{|S|} \tag{7.3b}$$

$$R = \text{Abrangência} = \frac{|S \bigcap G|}{|G|} \tag{7.3c}$$

[9]Em inglês, *miss rate* ou *false negative rate*

[10]Em inglês, *false discovery rate*

$$\text{Erro de Não Reconhecimento} = \frac{VP}{VP + FN} \tag{7.4}$$

$$\text{Erro de Não Reconhecimento} = \frac{FP}{VP + FP} \tag{7.5}$$

Para a realização do teste foi desenvolvido um corpus com frases extraídas do *Facebook da Samsung Mobile*[11]. O corpus é composto por 1038 frases. Por cada frase a média do número de aspetos anotados são 1.33 aspetos, sendo que a frase com mais aspetos apresenta 8 aspetos e a com menos aspetos tem 0, como se pode ver na Figura 7.4.

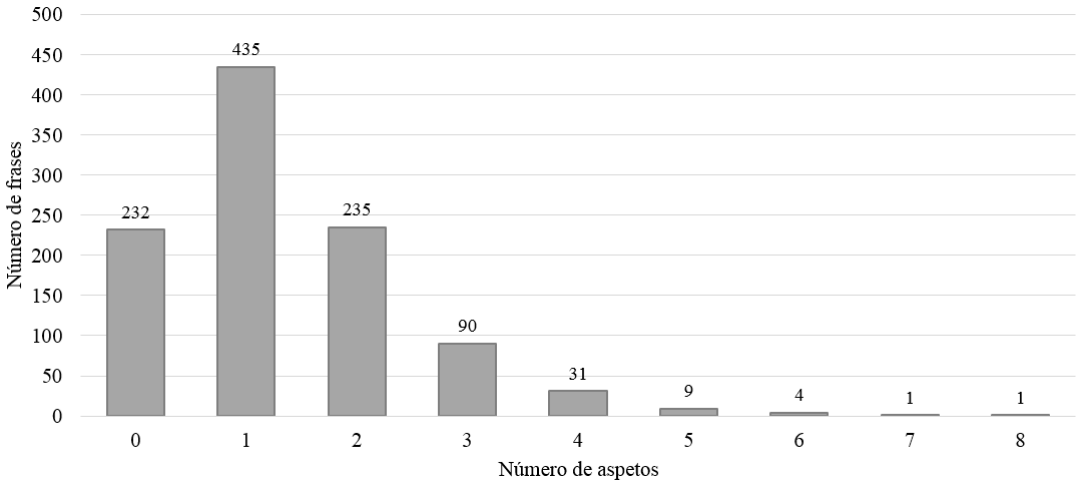

Figura 7.4: Distribuição dos aspetos no corpus de teste da extração de aspetos para o Inglês.

Resultados e Análise

Na Figura 7.5 é possível visualizar a evolução da qualidade da ferramenta à medida que se usam mais relações de dependência para extrair novos aspetos. Idealmente pretende-se obter uma Medida-F alta e o número de falsos positivos e falsos negativos baixos. É de notar que em todos as repetições de teste os resultados eram sempre iguais o que simboliza um desvio padrão de 0%.

Por exemplo, se usarmos a relação com maior frequência (ou seja, a relação que usando a lista inicial pré-definida, mais aspetos permitiu extrair), a Medida-F que obtemos é 62.1% (primeiro ponto na figura). Na Figura 7.6 está representado a evolução da média de aspetos considerados falso positivos, ou seja aspetos que foram extraídos que não deviam ter sido considerados aspetos.

À medida que mais relações são usadas maior é a Medida-F e menor é o número de falsos negativos, no entanto o número de falsos positivos também aumenta. O mesmo acontece com o número de falsos positivos que também aumenta com o número de relações de dependência usadas. Ou seja, à medida que acrescentamos mais relações de dependências acertamos mais aspetos, no entanto também aumentamos o número de aspetos que a ferramenta erra. No entanto, como podemos ver no gráfico, as últimas relações (as relações menos frequentes) já pouca influência causam à ferramenta, uma vez que os resultados estabilizam (como se pode ver a partir das 9 relações mais frequentes).

Naturalmente usar apenas a relação mais frequente não aparenta grande vantagens uma vez que o erro de não reconhecimento é bastante elevado (46%), mesmo que o número de falsos positivos seja mais baixo (erro de 26.1%). Considerou-se o número de relações

[11] Disponível em `https://www.facebook.com/SamsungMobileUK`

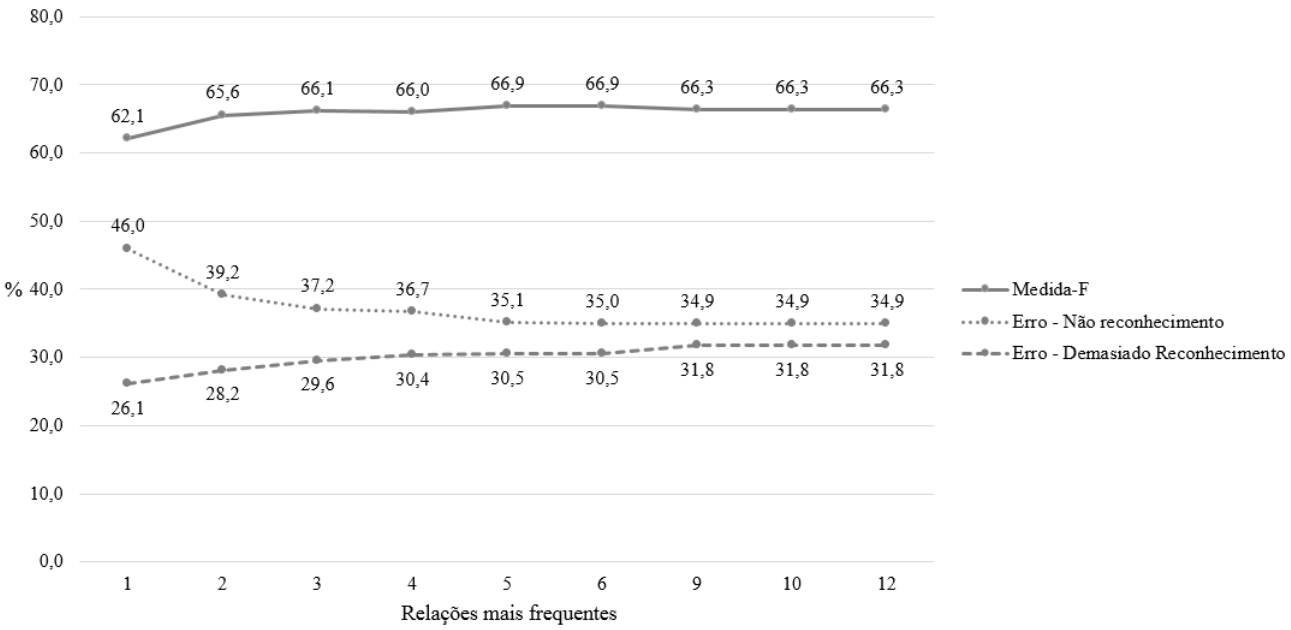

Figura 7.5: Resultados obtidos na extração de aspetos para o Inglês.

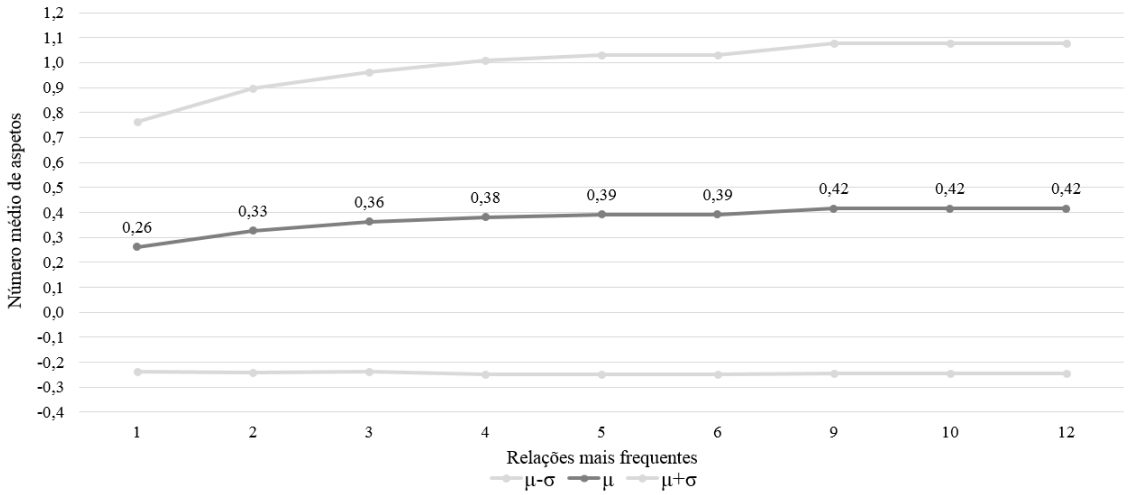

Figura 7.6: Média de número de aspetos falso positivos.

ideal as seis mais frequentes, uma vez que em termos de Medida-F foi o máximo obtido no teste. Em termos do erro de não reconhecimento fica apenas a 0.1 pontos percentuais do erro mais baixo obtido pelo teste. Já o erro de demasiado reconhecimento tem uma subida de dois pontos percentuais logo na opção a seguir (as nove relações mais frequentes). Embora tenha uma diferença de quase 4 pontos percentuais do mínimo obtido, achou-se que era a melhor relação de falsos positivos e falsos negativos sem prejudicar muito o erro de não reconhecimento.

Um outro teste realizado foi tentar perceber quais eram os melhores tipo de análise e limpeza de aspetos que se poderia usar. De forma a perceber isso foram realizados os seguintes testes, cujos resultados estão descritos na tabela 7.12:

- **Teste A** Este teste simula a opção de não existir qualquer tipo de limpeza aos aspetos.

- **Teste B** Neste teste apenas é usado os dicionários de palavras comuns, ou seja se o

aspeto candidato for uma palavra comum (substantivo, adjetivo o verbo) é removido da lista de aspetos possíveis.

- **Teste C** Neste teste para além do uso dos dicionários das palavras comuns, é introduzido o dicionário de referências temporais, ou seja, se o aspeto candidato for um palavra de referência temporal deixa de ser considerado um aspeto.

- **Teste D** No teste D é introduzido os dicionários de *stopwords* e palavras de opinião. Ou seja, para além dos dicionários de palavras comuns e referências temporais, se o aspeto candidato for uma *stopword* ou uma palavra de opinião é descartado.

- **Teste E** Este teste acrescenta a funcionalidade de descartar aspetos se estes incluírem números. Todas as funcionalidades dos testes em cima são também tidas em conta neste teste.

- **Teste F** Neste teste, para além de todos os dicionários e funcionalidades dos testes anteriores, é acrescentado a limitação de que o aspeto tem de ser uma palavra reconhecida pelo corretor automático de inglês.

- **Teste G** Neste teste, que também inclui todas as funcionalidades e dicionários dos testes anteriores, analisa-se, de novo e de forma isolada, a palavra e passa-se pelo identificador de classe gramatical que se não identificar o aspeto como um substantivo este é descartado.

- **Teste H** Por fim, este teste engloba todas as funcionalidades e dicionários anteriores à exceção da limitação acrescentado no Teste F. Ou seja, o aspeto candidato não é removido se não for reconhecido pelo corretor ortográfico.

Métricas	Teste A	Teste B	Teste C	Teste D	Teste E	Teste F	Teste G	Teste H
Medida-F (%)	53.6	62.5	63.9	63.9	66.6	66.4	66.9	67.4
Erro - Não Reconhecimento (%)	21.1	32.4	32.5	32.5	32.53	34.4	35.0	33.1
Erro - Demasiado Reconhecimento (%)	59.2	41.3	38.8	38.8	33.57	32.04	30.5	31.4

Tabela 7.12: Testes ao tipo de limpeza a usar na extração de aspetos.

Como se pode observar, há medida que se acrescenta novas restrições à fase de limpeza de aspetos o sistema vai melhorando à exceção de duas situações. Quando no Teste D se introduz os dicionários de *stopwords* e palavras de opinião o sistema mantém-se igual, ou seja não apresentou qualquer melhoria. Isto deve-se ao facto de geralmente as palavras de opinião e as *stopwords* não serem identificadas como substantivos por isso nem sequer serem candidatas a aspetos. Já no Teste F, onde se adiciona a restrição de que o aspeto candidato tem de ser uma palavra reconhecida pelo corretor ortográfico como sendo uma palavra correta, os resultados da ferramenta descem uma pequena percentagem. Isso também se vê nos Testes G e H em que o primeiro usa essa restrição e o segundo não, e o segundo este apresenta melhores resultados.

Dado estes resultados escolheu-se o conjunto de restrições simulados no Teste H cujos resultados estão apresentados em pormenor na tabela 7.13.

Total de Aspetos	1383
Precisão	68.5%
Abrangência	66.3%
Medida-F	67.4%
Erro - Não Reconhecimento	33.1%
Erro - Demasiado Reconhecimento	31.4%
Média de Aspetos Falso Positivos por frase	0.42 ($\sigma = 0.57$)

Tabela 7.13: Resultados para a ferramenta de extração de aspetos para o Inglês.

Teste de Performance

De forma a analisar a performance da ferramenta foi feito um teste que contabiliza o tempo necessário para extrair os aspetos dado uma determinada frase. Para este teste foi usado o mesmo corpus usado para os testes de qualidade já descrito em cima.

Resultados e Análise

Na Figura 7.7 está representado o tempo necessário (medido em milissegundos) para a extração de aspetos a partir de uma frase. Uma frase é representada pelo número de palavras que contém. Como se pode observar, quanto mais palavras a frase contém, mais tempo a ferramenta demora, o que se deve ao facto da ferramenta ter mais nós para avaliar. No entanto, mesmo com uma frase com mais de 320 palavras o tempo de extração é apenas de 8.2 milissegundos o que é bastante baixo.

É de notar, no entanto, que no corpus cerca de 50% das frases tem entre 40 a 120 palavras que pode representar que as frases mais frequentes são desse tamanho o que precisa de apenas 2 milissegundos para a extração.

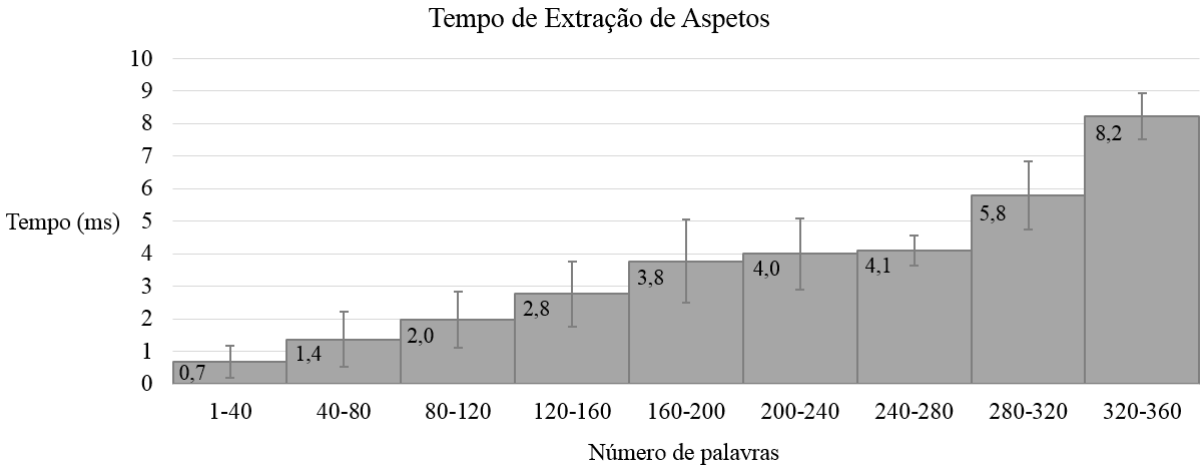

Figura 7.7: Tempo de extração de aspetos consoante o tamanho da frase.

7.2.3 Extração de Entidades

Tal como a ferramenta anterior de extração de aspetos foram realizados alguns testes para avaliar a ferramenta de extração de entidades para a Língua Inglesa. Os testes realizados tanto foram de avaliação da qualidade como de performance e cada um deles é descrito em detalhe nas seguintes secções.

Testes de Qualidade

O teste de qualidade realizado pretende perceber qual é o conjunto de relações de dependência que melhores resultados apresenta. Os testes de qualidade foram repetidos 30 vezes de forma a produzir resultados estatísticamente relevantes

Os resultados extraídos do teste usam diferentes métricas como a Medida-F, o Erro de Não Reconhecimento e o Erro de Demasiado Reconhecimento, já descritas na secção anterior em 7.2.2. Para a realização do teste foi desenvolvido um corpus com frases extraídas do Facebook da Samsung Mobile[12]. O corpus é composto por 998 frases. Cada instância ou frase do corpus contém em média 1.48 entidades, sendo que no máximo temos 7 entidades e no mínimo temos 0 entidades por frase, como se pode ver na Figura 7.8.

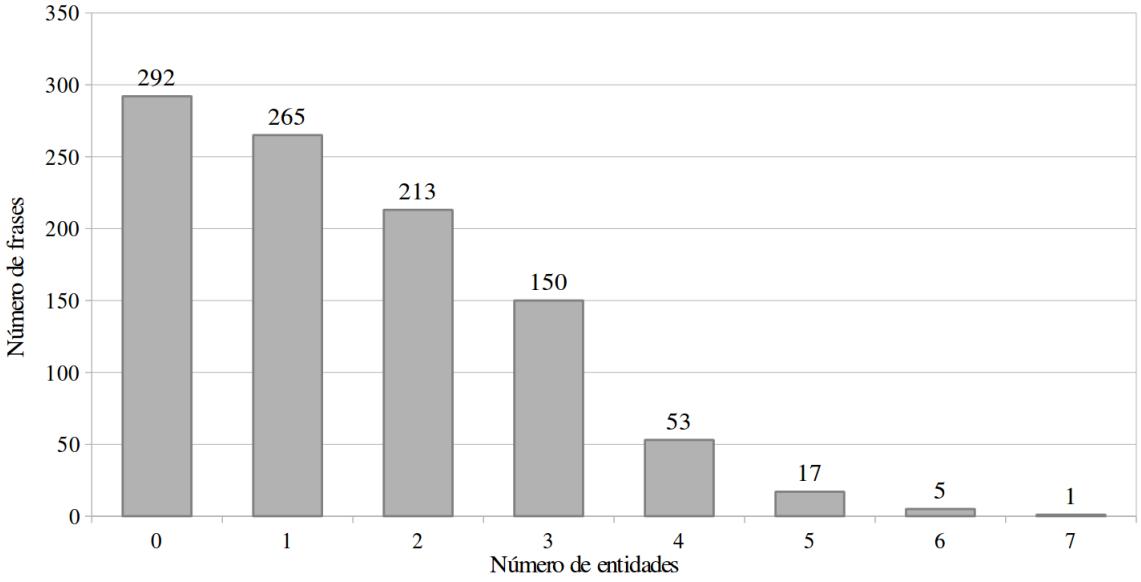

Figura 7.8: Distribuição dos aspetos no corpus de teste da extração de entidades para o Inglês.

Resultados e Análise

Como explicado na secção 6.6.3, da fase de extração de relações, que servem como método para extrair as entidades de uma qualquer fase, é recolhida uma lista de dependências entre verbos e entidades conhecidas usando a árvore de dependências. No entanto, é necessário perceber desse conjunto de regras quais são as que melhores resultados produzem. Por isso, neste primeiro teste foram analisados os resultados consoante diferentes conjuntos de dependências. Esses conjuntos foram construidos através da frequência que cada relação apresentava quando foi extraída de forma automática. Ou seja, por exemplo, um conjunto tem as 5 relações mais frequentes, e outro as 10 relações mais frequentes.

Na Figura 7.9 é possível visualizar os resultados obtidos nos diferentes conjuntos. Este teste foi repetido 30 vezes e os resultados foram sempre iguais, ou seja o desvio padrão dos resultados é de 0%. Tal como para a extração de aspetos, nesta ferramenta pretende-se conseguir uma Medida-F alta e o número de falsos positivos e negativos baixos. No gráfico é possível observar se usarmos conjuntos de relações muito pequenos o os erros são muito elevados e a Medida-F bastante baixa chegando mesmo a alcançar um mínimo de 9% quando usa apenas uma relação de dependência que é a mais frequente. Há medida que o conjunto de dependências aumenta a Medida-F também aumenta. Já o erro de demasiado

[12]Disponível em `https://www.facebook.com/SamsungMobileUK`

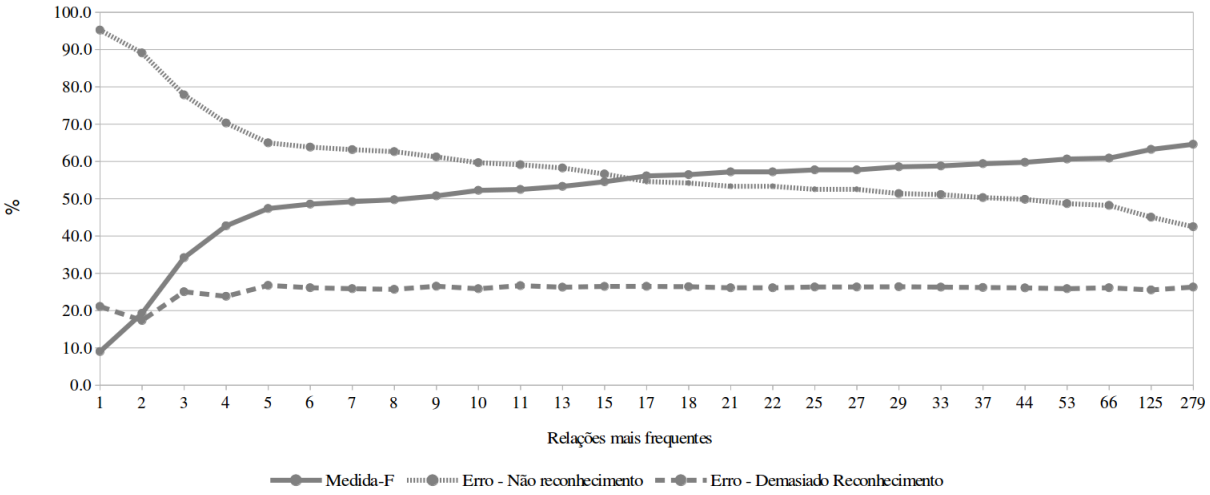

Figura 7.9: Resultados obtidos na extração de entidades para o Inglês.

reconhecimento mantém-se estável nos diferentes testes nunca ultrapassando os 27 %. Isso permite concluir que mesmo usando um conjunto de relações muito limitado o número de falsos positivos não varia, o que pode indicar que a fase de análise e limpeza das entidades candidatas, descrita em 6.6.3 é uma fase importante. Já o erro de não reconhecimento que permite perceber a quantidade de falsos negativos, baixa consoante o número de relações usadas. Inicialmente apresenta um erro bastante alto, cerca de 95.2% usando apenas uma relação, no entanto vai diminuindo apresentando um mínimo de 42.5% quando se usam todas as relações extraídas.

Uma questão importante de analisar é os falsos positivos, ou seja entidades que são extraídas pela ferramenta que não deveriam ter sido consideradas como entidades. Na Figura 7.10 está representado a média e o desvio padrão do número de falsos positivos por frase nos diferentes testes. Tal como no gráfico anterior, o número de falsos positivos mantém-se estável nos diferentes testes. Obtém um mínimo de 0.02 entidades por frase ao usar apenas uma relação e um máximo de 0.31 entidades se usar todas as relações de dependências. No entanto, a diferença entre o mínimo e máximo é de apenas 0.29. Observando o desvio padrão é possível perceber que este é sempre alto comparando com a média. Por exemplo, quando usamos todas apenas as 10 relações mais frequentes existem frases em que são extraídas 5 entidades falso positivas embora a média seja baixa de apenas 0.21 entidades.

Dado os resultados obtidos, optou-se por usar todas as relações extraídas, ou seja, as 279 relações de dependência. Essa decisão prende-se pelo facto de que o número de falso positivos não aumenta de acordo com o número de relações usadas, à exceção do conjunto das duas regras mais frequentes que tem o menor erro de 17%. No entanto nenhum desses conjunto iniciais são interessantes uma vez que tanto o erro de não reconhecimento e a Medida-F apresentar valores bastante fracos. Uma vez que tanto o erro de não reconhecimento e Medida-F vão melhorando com o aumento do tamanho do conjunto de relações escolheu-se as 279 relações.

Por fim, outro teste realizado foi tentar perceber quais eram os melhores tipo de restrições de entidades que se poderia usar. De forma a perceber isso foram realizados os seguintes testes, cujos resultados estão descritos na tabela 7.14:

- **Teste A** Este teste simula a opção de não existir qualquer tipo de restrição nas entidades, ou seja se uma entidade é uma candidata não lhe é imposto qualquer restrição passando logo a entidade final.

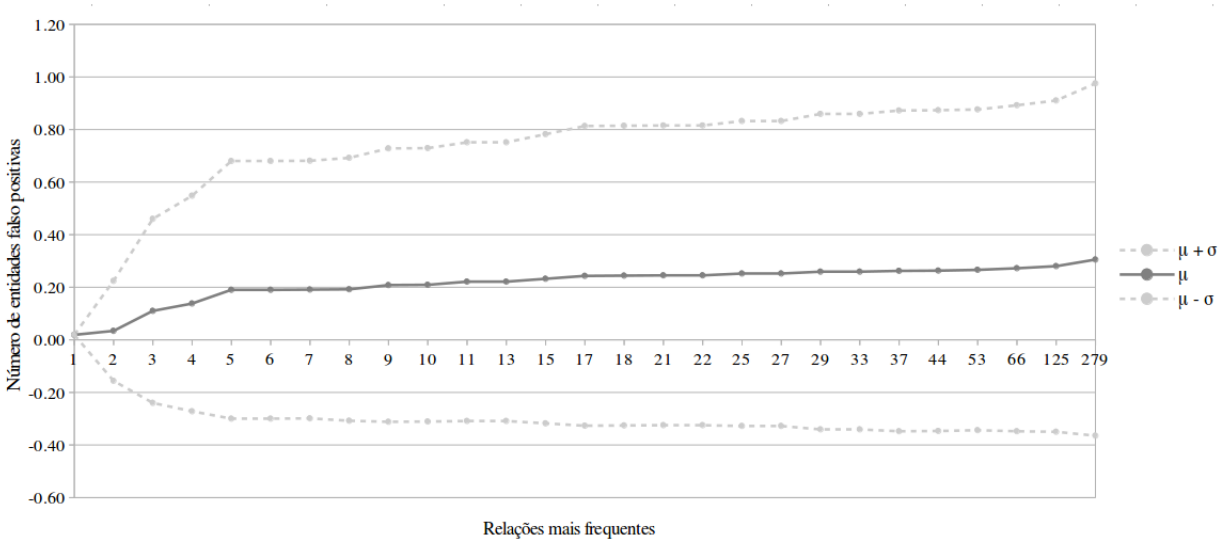

Figura 7.10: Média de número de entidades falso positivos.

- **Teste B** Para este teste foi acrescentado a restrição de que a entidade candidata não pode pertencer a nenhum dos seguintes tipos de palavras: *urls*, emails, menções, *hashtags* e símbolos.
- **Teste C** Neste teste para além da restrição anterior todas as entidades que sejam palavras de opinião ou *stopwords* são excluídas das lista de candidatas.
- **Teste D** Para além das restrições simuladas no teste B e C, todas as palavras que são referências temporais são também excluídas.
- **Teste E** Para este teste são usadas também todas as restrições dos testes anteriores e acrescenta-se o uso dos dicionários das palavras frequentes, ou seja se uma entidade candidata for um substantivo, adjetivo ou verbo comum esta é excluída.
- **Teste F** Neste teste, para além de todas as restrições descritas nos outros testes todas as entidade candidatas que sejam compostas apenas por números são excluídas.
- **Teste G** Neste teste, para além de todas as restrições dos testes anteriores, a entidade candidata não pode ser considerada um aspeto.
- **Teste H** Por fim, este teste engloba todas as restrições anteriores e também a restrição que dita que se a entidade quando corrigida pelo corretor ortográfico for considerada um aspeto esta é excluída da lista de entidades candidatas.

Métricas	Teste A	Teste B	Teste C	Teste D	Teste E	Teste F	Teste G	Teste H
Medida-F (%)	46.7	46.8	52.6	55.5	63.7	64.0	65.1	64.6
Erro - Não Reconhecimento (%)	33.1	33.1	33.8	33.8	36.9	36.9	41.6	42.4
Erro - Demasiado Reconhecimento (%)	64.1	63.9	56.1	52.1	35.5	34.8	26.42	26.31

Tabela 7.14: Testes ao tipo de limpeza a usar na extração de entidades.

Como se pode observar pelos resultados, sempre que se acrescenta uma nova restrição à fase de limpeza de entidades a ferramenta alcança melhor Medida-F, à exceção do último teste. É possível observar que a maior subida foi conseguida entre o teste D e E, quando se acrescentou o uso dos dicionários das palavras frequentes. Embora o erro de não reconhecimento que é relativo ao falsos negativos tenha subido 3.1%, o erro de demasiado reconhecimento desceu mais de 15%.

Dado estes resultados escolheu-se o conjunto de restrições simulados no Teste G cujos resultados estão apresentados em pormenor na tabela 7.15.

Total de Entidades	1485
Precisão	73.5%
Abrangência	58.3%
Medida-F	65.1%
Erro - Não Reconhecimento	41.6%
Erro - Demasiado Reconhecimento	26.4%
Média de Entidades Falso Positivos por frase	0.31 ($\sigma = 0.67$)

Tabela 7.15: Resultados para a ferramenta de extração de entidades para Inglês.

Embora o erro de demasiado reconhecimento seja baixo, o erro de não reconhecimento apresenta valores altos. Existem várias explicações para esse resultado. Uma delas é que o corpus usado para extração dessas relações pode não ser suficiente grande e variado para conseguir extrair todas as relações e por isso existirem relações de dependência que não foram conhecidas e por isso algumas entidades nem chegarem a ser analisadas. Outro problema encontrado é a qualidade de ferramentas base necessárias para extrair entidades, como por exemplo o Identificador de classes gramaticais. Na frase em baixo, é considerado uma entidade a expressão: "*samsung*".

"*I did contact with samsung customer repair centre and I have been told , there is no warranty for that even if that is obviously fabric issue* "

Tal como descrito na secção 6.6.3 apenas os substantivos são candidatos a entidades. No entanto, neste caso a entidade é identificada como um adjetivo e por isso não é identificada pela ferramenta.

Um outro exemplo é também na frase seguinte:

"*i had my samsung Galaxy 5 for 11 months, and charger cover has snapped off....*"

Neste exemplo a entidade considerada como correta é a expressão "*samsung Galaxy 5*". No entanto, apenas a expressão "*Galaxy 5*" é considerada como entidade uma vez que a palavra "*samsung*" é identificada, mais uma vez, como um adjetivo.

Por fim, um outro problema é identificado na ferramenta de *chunking*. Por exemplo na frase em baixo, a entidade pretendida seria "*samsung galaxy edge s6*".

"*why aren't three getting emerald green samsung galaxy edge s6?????*"

Apesar da ferramenta de identificação de classes gramaticais identificar todas as palavras da entidade como substantivos, a ferramenta de *chunking* não as considerou como fazendo parte do mesmo *chunk* e por isso apenas a expressão "*samsung galaxy edge*" foi considerada uma entidade.

Teste de Performance

Outro aspeto importante é a performance da ferramenta. De forma a avaliar a performance temporal da ferramenta foi realizado um teste que contabiliza o tempo preciso para extrair entidades de uma frase. Para a realização do teste foi usado o mesmo corpus descrito anteriormente nos testes de qualidade.

Resultados e Análise

Na Figura 7.11 é possível visualizar o tempo necessário para a extração de entidades. Como é possível observar quanto maior é a frase, ou seja quantas mais palavras a frase contém mais tempo é necessário. Esta relação é fácil de perceber tendo em conta que quanto mais palavras a frase tiver mais demorada é a construção da árvore de dependências e mais nós são necessários analisar.

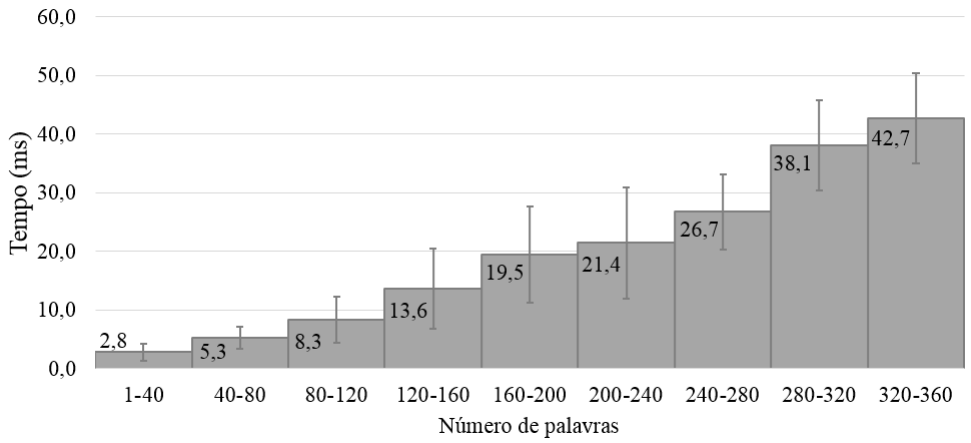

Figura 7.11: Tempo de extração de entidades consoante o tamanho da frase.

No entanto, os tempos alcançados são aceitáveis, tendo conta que estão representados em milissegundos, as frases mais longas demoram apenas 40 milissegundos. No entanto, é importante referir que geralmente as frases não são tão longas, sendo que as mais frequentes no corpus, tem um tamanho entre 40 a 120 palavras (cerca de 61% do corpus), o que necessita apenas de 2 milissegundos.

7.2.4 Extração de Quíntuplos

De forma a avaliar a ferramenta de extração de quíntuplos foram realizados dois tipos de teste: qualidade e performance. Nesta secção cada um deles é descrito em detalhe e são apresentados e analisados os resultados de cada um.

Testes de Qualidade

Para avaliar a qualidade foi realizado um teste, que permite perceber se a ferramenta é capaz de dado as entidades e os aspetos relacioná-los de forma correta. Para tal foi desenvolvido um corpus com cerca de 700 instâncias como se pode ver na Figura 7.12. Esse corpus foi construído usando diversos textos extraídos da página do *Facebook da Samsung Mobile*, e removidos todos os textos que não tinham quíntuplos associados. Pode-se verificar que quase 70% do corpus é composto por instâncias com apenas um quíntuplos, e cerca de 15% possui dois quíntuplos. A métrica usada para avaliar a ferramenta é a abrangência já mencionada anteriormente na equação 2.6.

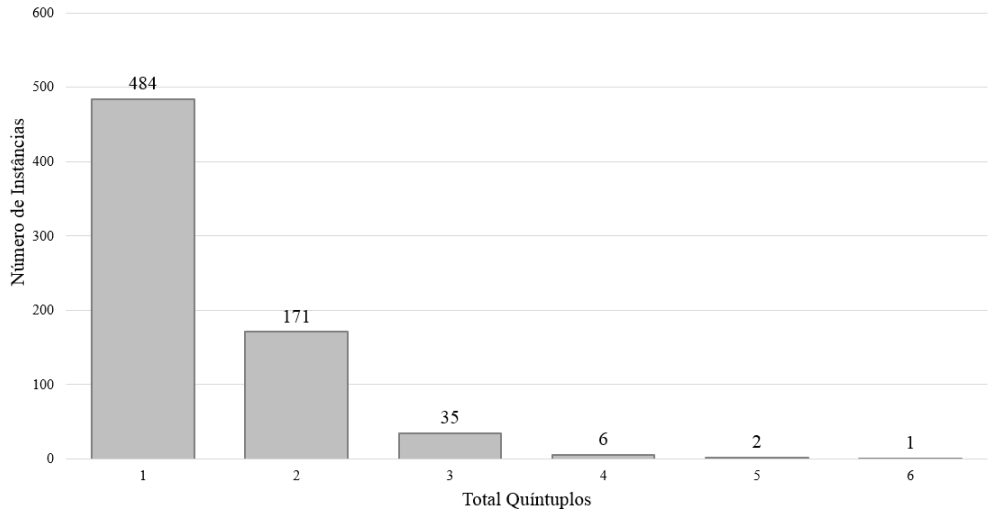

Figura 7.12: Distribuição dos quíntuplos no corpus de teste da extração de quíntuplos para o Inglês.

Resultados e Análise

Na tabela 7.16 é possível observar o resultado obtido no teste efetuado. De todos os quíntuplos, a ferramenta é capaz de os identificar de forma correta cerca de 86% das vezes.

Total de Quíntuplos	971
Abrangência	86.4%

Tabela 7.16: Resultados para a ferramenta de extração de quíntuplos para o Inglês.

Como esperado, a ferramenta apresenta melhores resultados em instâncias que produzem apenas um quíntuplo, conseguindo acertar cerca de 90% desses casos. Há medida que o número de quíntuplos por instância aumenta a abrangência diminui. Por exemplo em instâncias com 3 quíntuplos, a ferramenta consegue acertar 80% quíntuplos. No caso da instância que tem 6 quíntuplos, a ferramenta apenas consegue acertar 33% desses quíntuplos. No entanto é de notar que nessa instância o número de triplos é elevado uma vez que a ferramenta de identificador de frases falhou em separar diferentes frases, ficando assim com uma instância composta por diferentes frases mas que são vistas como uma, o que dificulta as análises posteriores, como por exemplo no seguinte texto:

"*I'm still waiting to hear when the s6 edge will receive an update to sort the multiple flaws in its software including dropped WiFi signal and saying I'm using all 3gb of ram at all times....please respond as the rest of the world have already started receiving their updates....*"

É de salientar que para este teste apenas foi testado a relação entre entidades e aspetos, ou seja, para cada uma das instâncias de teste foram fornecidas à ferramenta quais eram as entidades e os aspetos e a ferramenta construída os quíntuplos usando apenas essa informação.

De forma a testar a usabilidade da ferramenta, foram aglomeradas mais cerca de 7.000 frases extraídas do Facebook da Samsung, e foram extraídos os quíntuplos de cada frase. Na tabela 7.17 estão representadas as 3 entidades mais frequentes e os seus aspetos mais frequentes associados a uma polaridade média.

Entidade Samsung		Entidade Galaxy S4		Entidade Galaxy S5	
Aspeto	**Polaridade**	**Aspeto**	**Polaridade**	**Aspeto**	**Polaridade**
General	-0.2	General	-0.8	General	-0.7
Update	-0.9	Update	-0.9	Screen	-0.5
Phone	0.2	SD Card	-0.1	Camera	-0.4
Device	-0.1	Phone	-0.7	Features	0.3
Service	-0.6	Lollipop	-0.9	Update	-0.8
Handset	0.1	Battery	-0.9	Battery	-0.7
Manufacturer	-0.5	Notification	-0.3	Handset	0.2
Contract	-0.7	Factory	-0.8	Zoom	0.1
Staff	-0.8	Camera	-0.3	Messages	-0.1
Upgrade	-0.8	Screen	0.1	Power	-0.7
Store	-0.4	Sound	-0.2	Memory	-0.1

Tabela 7.17: Resultados produzidos pela ferramenta de extração de quíntuplos.

Usando apenas as três entidades com maior frequência obtemos resultados que podem ser interessantes. Por exemplo, em todas as entidades o aspeto "*update*" parece gerar um consenso que gera algo negativo e é muito frequentemente referido. Outro exemplo, falando das dois últimas duas entidades, é possível observar que a bateria gera muitas queixas. Uma vez que todos os comentários foram extraídos da mesma página e geralmente as pessoas publicam nessa página com o efeito de se queixarem, as polaridades médias estão muitas vezes a tender para a polaridade negativa. No entanto, se analisarmos para além das entidades mais frequentes, encontramos alguns problemas, como por exemplo, a ferramenta para além de extrair a entidade "*Galaxy S5*" também extrai a entidade "*S5*", e assume que são entidades diferentes. Mas no contexto da *Samsung* ambas as entidades podem-se referir ao mesmo produto. Ou seja, a mesma entidade é muitas vezes referida de diferentes formas o que dificulta este tipo de análise.

Teste de Performance

De forma a analisar a ferramenta foi realizado um teste de performance que permite calcular o tempo necessário para extração de quíntuplos desde a extração de entidades e aspetos até ao cálculo da polaridade de cada quíntuplo.

Resultados e Análise

Como se pode observar na Figura 7.13, a fase que necessita de mais tempo é a de extração de polaridade, que vai aumentando à medida que o tamanho da frase aumenta. A Extração do texto relevante para cada quíntuplo, embora seja a fase menos demorada, também acresce à medida que a frase aumenta.

Em média para uma frase mais pequena é necessário cerca de 6 segundos por extrair os quíntuplos completos. Já numa frase com um número maior de palavras necessita de cerca de 12 segundos.

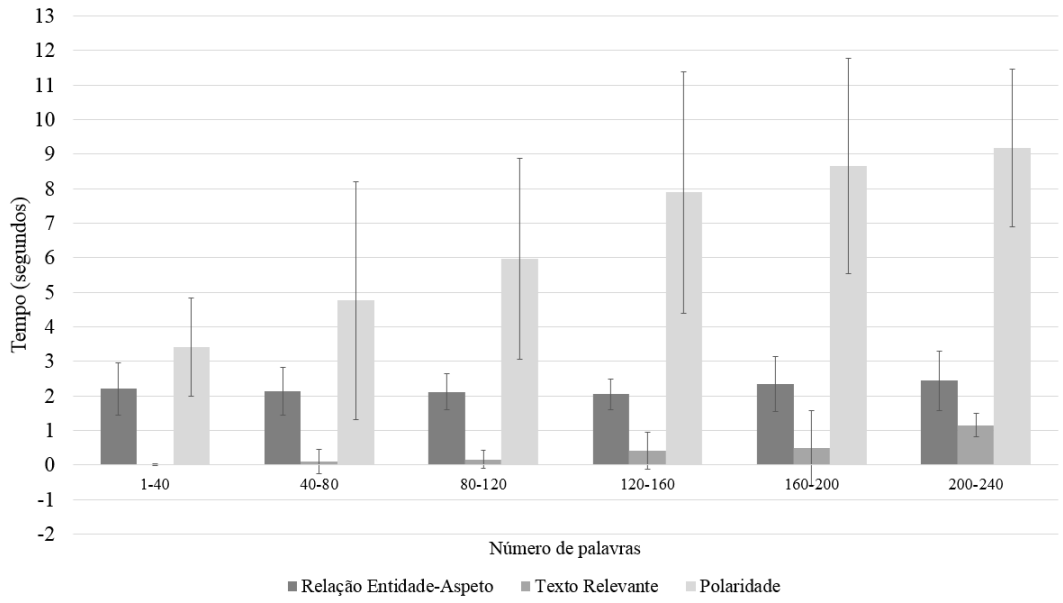

Figura 7.13: Tempo médio para cada fase por tamanho da frase.

7.3 Ferramenta de Extração de Opiniões para o Português

Nesta secção são descritos todos os testes realizados para as diferentes ferramentas de extração de opiniões para a Língua Portuguesa. Por cada teste serão apresentados e analisados os respetivos resultados.

7.3.1 Extração de Polaridade

Para avaliar a ferramenta de extração de polaridade para a Língua Portuguesa foram realizados dois tipos de testes testes: testes de qualidade e testes de performance. Nesta secção cada um desses grupos de teste são descritos e os seus resultados apresentados e analisados.

Testes de Qualidade

Tal como para a mesma ferramenta de extração de polaridade para a Língua Inglesa foram realizados três diferentes testes. O primeiro teste tem como objetivo de perceber qual o *kernel* que mais se adequa ao problema em questão. Para cada *kernel* foi necessário fazer um *grid-search* de forma a otimizar os parâmetros usando validação 10-*folds* para comparação dos modelos. Os resultados apresentados representam os modelos com os melhores parâmetros encontradas por cada *kernel*. O segundo teste tem como objetivo perceber quais são as características que mais influenciam o resultado, obtendo assim o melhor conjunto de características que produz o modelo com melhores resultados.

Tal como para a mesma ferramenta em Inglês, foram usadas as seguintes métricas: Precisão e Abrangência, Medida-F para cada uma das classes (ver equações em 7.1) e a Medida-F global (ver equação 7.2) Uma vez que a Máquina de Vetores de Suporte é um algoritmo que dado o mesmo corpus de treino produz sempre o mesmo modelo, ou seja encontra sempre o máximo global, apenas foi necessário repetir os testes uma vez.

Para realizar os diferentes testes foi desenvolvido um corpus de teste. Tal como o corpus de treino este contém textos extraídos da rede social *Facebook*. Na Figura 7.14 está

representada a distribuição das instâncias por cada tipo de polaridade. O corpus possui 1275 instâncias sendo que cerca de 50% pertence à classe negativa, 35% à classe neutra e 15% à classe positiva.

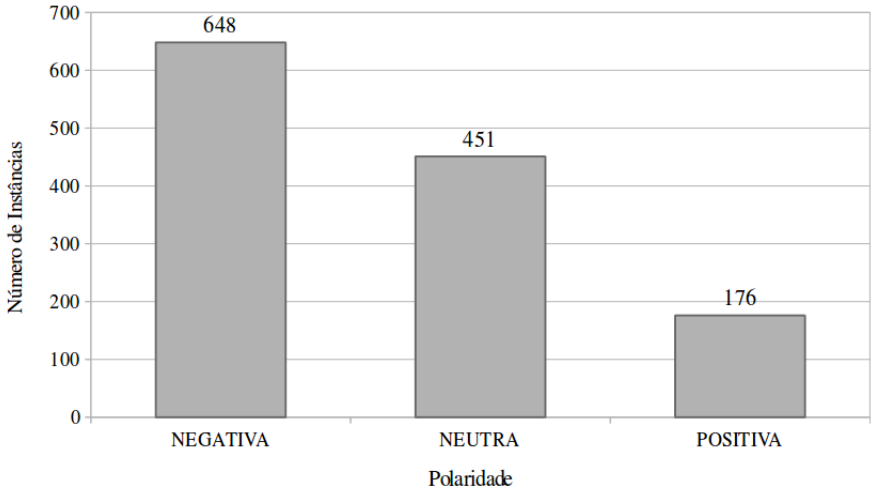

Figura 7.14: Distribuição das instâncias por cada tipo de polaridade no corpus de teste.

Resultados e Análise

O primeiro teste efetuado pretende perceber qual o *kernel* das Máquinas Vetor de Suporte (descrito na secção 2.2.2) que melhor se adapta aos dados usados. No contexto das Máquinas Vetor de Suporte, o *kernel* é a função que permite separar as diferentes classes.

Na tabela 7.18 estão apresentados os resultados para três diferentes *kernels*.

Métricas	***Kernel RBF***	***Kernel* Linear**	***Kernel* Polinomial**
Medida-F Positiva (%)	43.2	42.3	42.9
Medida-F Negativa (%)	73.0	71.0	71.4
Média (%)	**58.1**	**56.6**	**57.2**
Desvio Padrão (%)	**14.9**	**14.3**	**14.3**

Tabela 7.18: Resultados dos testes a diferentes *kernels* das Máquinas Vetor de Suporte.

O *Kernel* com melhor resultado é o *RBF* com uma melhoria de quase 1% na Medida-F média. Tanto para a Medida-F Positiva como para a Negativa o *kernel* apresenta melhores resultados. No entanto, é de notar que a diferença entre *kernels* não é muito grande, tendo obtido uma diferença de menos de 2% na Medida-F para o melhor e pior resultado. Mesmo assim, e dado os resultados obtidos na Medida-F, foi escolhido o *kernel RBF*.

O segundo teste realizado, tinha como objetivo perceber perceber quais as características que mais influenciavam o sistema. Para tal foram realizados os seguintes conjuntos de testes, cujos resultados estão apresentados na tabela 7.19 :

- **Teste A** Para o primeiro teste apenas foram consideradas as características que incluem algumas métricas do texto, como número de palavras, tamanho do texto, entre outras.
- **Teste B** Neste teste, para além das características anteriores, foram também adicionadas as relativas às *stopwords* e negações.

- **Teste C** Neste teste, foram adicionadas as características relacionadas a menções, *urls*, *hashtags* e palavras em maiúscula.
- **Teste D** Neste teste, foram adicionadas as características relacionadas com os *smiles*.
- **Teste E** Para este teste, para além de todas as características anteriores as características relacionadas com classes gramaticais e pontuação foram adicionadas.
- **Teste F** Neste teste, as informações relacionadas com os dicionários temáticos foram acrescentados.
- **Teste G** Por fim, neste teste estão incluídas todas as caraterísticas de conteúdo, ou seja foram adicionadas as caraterísticas relacionadas com calões, expressões ou referências temporais.

Métricas	Teste A	Teste B	Teste C	Teste D	Teste E	Teste F	Teste G
Medida-F Positiva (%)	30.1	29.8	31.5	30.9	36.8	37.1	38.0
Medida-F Negativa (%)	47.7	54.6	62.3	63.7	64.3	64.0	64.8
Média (%)	**38.9**	**42.2**	**46.9**	**47.3**	**50.5**	**50.6**	**51.4**
Desvio Padrão (%)	**8.8**	**12.3**	**15.3**	**16.4**	**13.7**	**13.4**	**13.3**

Tabela 7.19: Testes a diferentes conjuntos de características de conteúdo para a ferramenta de extração de polaridade para a Língua Portuguesa.

Como se pode observar, à medida que novas informações são adicionadas, melhor o sistema se comporta. Usando apenas as métricas do texto (Teste A), consegue-se alcançar uma Medida-F média de quase 39%.

As características que apresentam uma melhor melhoria para a ferramenta são as características relacionadas com *urls*, *hashtags*, palavras em maiúsculas e outras. Essa melhoria regista-se entre o Teste B e C com cerca de quase 5% de diferença. É de notar que deste o primeiro teste, a ferramenta apresenta uma deficiência na Medida-F da classe positiva, o que pode ser explicado pelo facto de que o corpus de treino ser composto por apenas 13% de instâncias de classe positiva o que pode não ser suficiente para treinar essa classe. As características que se mostraram ter menos impacto foram as características relacionadas com os dicionários temáticos, que apenas aumentou a Medida-F em 0.1%, como se pode ver no Teste F.

Para além dos testes anteriores, foram feitos testes relacionados com as características relacionados com léxico. De forma a perceber o impacto dessas características foram realizados os seguintes conjuntos de testes, cujos resultados estão apresentados na tabela 7.20:

- **Teste G** Neste teste, as características de conteúdo são as úncias adicionadas.
- **Teste H** Neste teste, foram adicionadas as características relativas ao léxico de polaridade *SentiLex*.
- **Teste I** Neste teste, foram adicionadas as informações relativas à lista de polaridades da Wizdee.
- **Teste J** Neste teste, foram acrescentadas as informações relativas ao léxico de polaridade *ReLi*.

- **Teste K** Neste teste, ao teste anterior foram retiradas todas as características que usam o texto pré-processado (referido em 6.7.1) que envolveu a remoção de todas as palavras específicas das redes sociais, calões e expressões idiomáticas.
- **Teste L** Neste teste, ao Teste J são removidas todas as características de léxico relacionadas com classes gramaticais.
- **Teste M** Neste teste, ao Teste J são removidas todas as características de léxico relacionadas com lemas e *stems*.

Métricas	**Teste G**	**Teste H**	**Teste I**	**Teste J**	**Teste K**	**Teste L**	**Teste M**
Medida-F Positiva (%)	38.0	43.6	43.2	43.2	42.1	45.4	41.8
Medida-F Negativa (%)	64.8	71.9	73.0	73.0	73.1	73.4	72.1
Média (%)	**51.4**	**57.7**	**58.1**	**58.1**	**57.6**	**59.4**	**56.9**
Desvio Padrão (%)	**13.3**	**14.1**	**14.8**	**14.8**	**15.5**	**13.9**	**15.1**

Tabela 7.20: Testes a diferentes conjuntos de características de conteúdo e de léxico para a ferramenta de extração de polaridade para a Língua Portuguesa.

É possível observar que, quando se acrescenta o primeiro dicionário de polaridade, no Teste H, a Medida-F melhora consideravelmente, cerca de 6%. No entanto, quando se acrescenta outros léxicos apenas apresenta uma melhoria pequena. Isto pode se dever ao facto de que entre os léxicos poderá existir uma grande interseção de informação. É de notar que o léxico *ReLi*, quando adicionado no Teste J não altera de todo o sistema.

Comparando o teste K com o J , é possível concluir que o pré-processamento do texto tem apenas uma influência de cerca de 0.5%. No entanto, se removermos todas as informações sobre as classes gramaticais a ferramenta melhora mais de 1%. Essa influência é mais notável na classe de polaridade positiva que melhora quase 2%. Já sem as informações relacionadas sobre os lemas e *stems* o sistema perde cerca de 1%.

Dado os resultados obtidos, foi decidido que o melhor conjunto de características é as características simuladas no Teste L. Os resultados finais da ferramenta estão apresentados em detalhe na tabela 7.21.

Total de Instâncias	1275
Precisão Negativa	79.4%
Precisão Positiva	40.9%
Abrangência Negativa	68.2%
Abrangência Positiva	51.1%
Medida-F Negativa	73.4%
Medida-F Positiva	45.4%
Medida-F Final	59.4% ($\sigma = 13.9\%$)

Tabela 7.21: Resultados do modelo final para a ferramenta de extração de polaridade para o Português.

Tal como já referido anteriormente, esta é uma tarefa muito sujeita à subjetividade. Ou seja, nem sempre é fácil ter o consenso entre diferentes pessoas sobre a polaridade no mesmo texto, como por exemplo na seguinte frase:

"*É que, pessoalmente, estou mortinho para ser 'alvo' dessa oferta...*"

Neste exemplo, é difícil perceber que o autor está a ser irónico ou não, o que dificulta a construção de uma ferramenta de extração de polaridade se até para nós, humanos, essa definição não é clara.

Um outro exemplo onde a ferramenta falha é na seguinte frase:

"*Com esta brincadeira ja la vão uns bons euros...*"

Esta frase é considerada uma frase negativa, no entanto a ferramenta considera-a como neutra. A frase em si não descreve nenhuma opinião explícita, no entanto, nós somos capazes de a interpretar como sendo uma frase negativa uma vez que conhecemos o conceito de gastar dinheiro e apresentado daquela maneira apresenta ter uma conexão negativa. No entanto, é difícil transmitir estes conceitos de forma a que uma ferramenta deste tipo consiga detetá-los e analisá-los corretamente.

Um outro problema muito encontrado neste corpus, pode ser observado nas seguintes frases:

"*Volta ZON estas perdoada...* "
"*Já fui MEO e entretanto mudei para a NOS e estou muito melhor* "

Ambas as frases poderiam ser consideradas negativas no entanto, uma vez que estas frases foram extraídas da página da MEO, para a entidade MEO estas deveriam ser consideradas negativas, uma vez que ambas as frases apresentam vontade ou satisfação por regressar a um concorrente. Por exemplo, ambas as frases foram classificadas pela ferramenta como sendo positivas, no entanto foram consideradas erradas uma vez que o corpus as considera negativas numa perspetiva do produto MEO.

Teste de Performance

De forma a analisar a performance da ferramenta foi feito um teste que calcula o tempo necessário para as várias etapas da ferramenta.

Resultados e Análise

Na tabela 7.22 são apresentados os tempos necessários para as diferentes fases. A fase de transformação do texto para um vetor de características que o representam demora em média 15 segundo por cada texto. É de salientar que na primeira extração de características esta fase demora em média 22 segundos, uma vez que a primeira vez obriga a que todos os dicionários, léxicos e ferramentas sejam inicializados. Este procedimento apenas é necessário fazer uma vez. O treino do modelo demora cerca de 5 minutos o que é um valor pouco preocupante, uma vez que esta fase é feita internamente e só feita apenas uma vez. Depois o modelo fica guardado para se usar sempre que se pretender extrair a polaridade de uma nova frase.

O processo completo de extração de polaridade de uma frase totaliza em média 15.59 segundos, que é a soma da fase de extração de características e de teste ao modelo. Como se pode observar a fase de extração de características é a fase que mais tempo consome, o que é esperado uma vez que nessa fase são feitos muitos processamentos diferentes.

7.3.2 Extração de Aspetos

À semelhança da ferramenta de extração de aspetos para a Língua Inglesa, para avaliar esta ferramenta foram realizados vários testes que podem ser divididos em dois grupos: testes de qualidade e performance. Nesta secção é descrito a especificação de cada teste e são apresentados os resultados e a sua análise.

Fase	Média (s)	Desvio Padrão (s)
Extração de Características	15.5	21.7
Treino do modelo final	333.94	7.56
Teste do modelo (1 instância)	0.009	0.0001

Tabela 7.22: Resultados do teste de performance realizado para a ferramenta de extração de polaridade.

Testes de Qualidade

Na abordagem seguida para o desenvolvimento desta ferramenta existem duas fases que tem uma grande influência na qualidade da ferramenta: a fase de extração de relações de dependência que são usadas para a extração de aspetos (descrita em 6.7.2), e a fase de análise e limpeza dos aspetos candidatos (descrita em 6.7.2). Como tal, o primeiro teste realizado pretende perceber quais são as relações de dependência que melhores resultados obtém. O segundo teste tem como objetivo perceber qual o melhor conjunto de regras que se devem usar na fase de limpeza de aspetos.

A qualidade da ferramenta é medida consoante diferentes métricas já referenciadas para a ferramenta semelhante para a língua inglesa. Essas métricas são Medida-F (ver equação 7.3), Erro de Não reconhecimento[13] (ver equação 7.4) e o Erro de Demasiado Reconhecimento[14] (ver equação 7.5).

Para a realização dos testes foi criado um corpus com frases extraídas do Facebook da Vodafone Portugal[15]. O corpus é composto por 999 frases. Por cada frase a média do número de aspetos anotados é 1.7 aspetos, sendo que a frase com mais aspetos apresenta 8 aspetos e a com menos aspetos tem 0, como se pode ver na Figura 7.15. Cerca de 57% das frases do corpus possuem entre um e dois aspetos anotados.

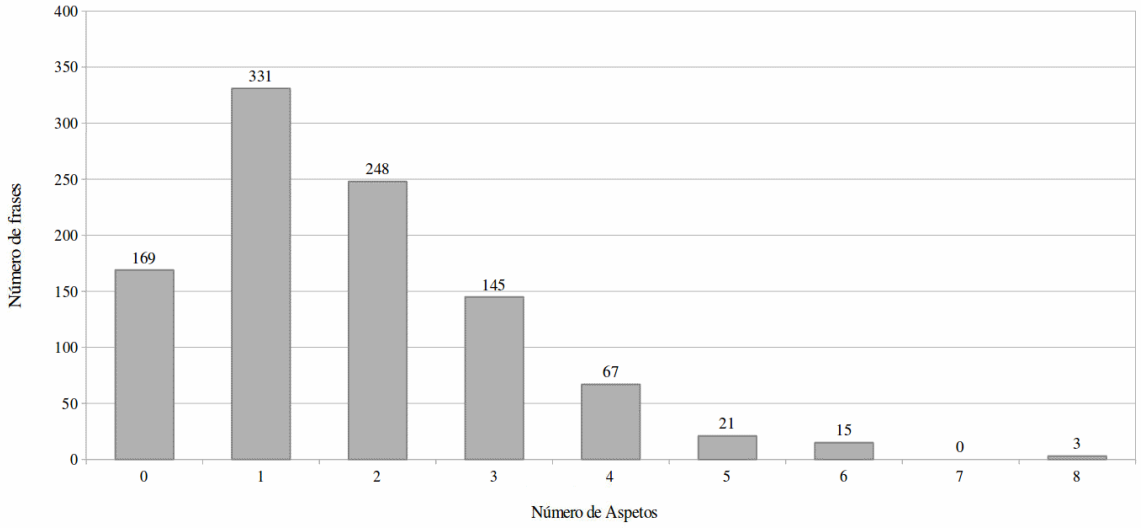

Figura 7.15: Distribuição dos aspetos no corpus de teste da extração de aspetos para o Português.

[13]Em inglês, *miss rate* ou *false negative rate*
[14]Em inglês, *false discovery rate*
[15]Disponível em `https://www.facebook.com/vodafonePT`

Resultados e Análise

Na Figura 7.16 é possível observar a evolução da qualidade da ferramenta à medida que se usam mais relações de dependência para extrair novos aspetos. Na Figura 7.17 está representado a média do número de aspetos falso positivos por cada frase, consoante o número de relações de dependeria usadas. É de notar que o teste foi corrido 30 vezes e que em todos as repetições do teste os resultados eram sempre iguais o que simboliza um desvio padrão de 0.

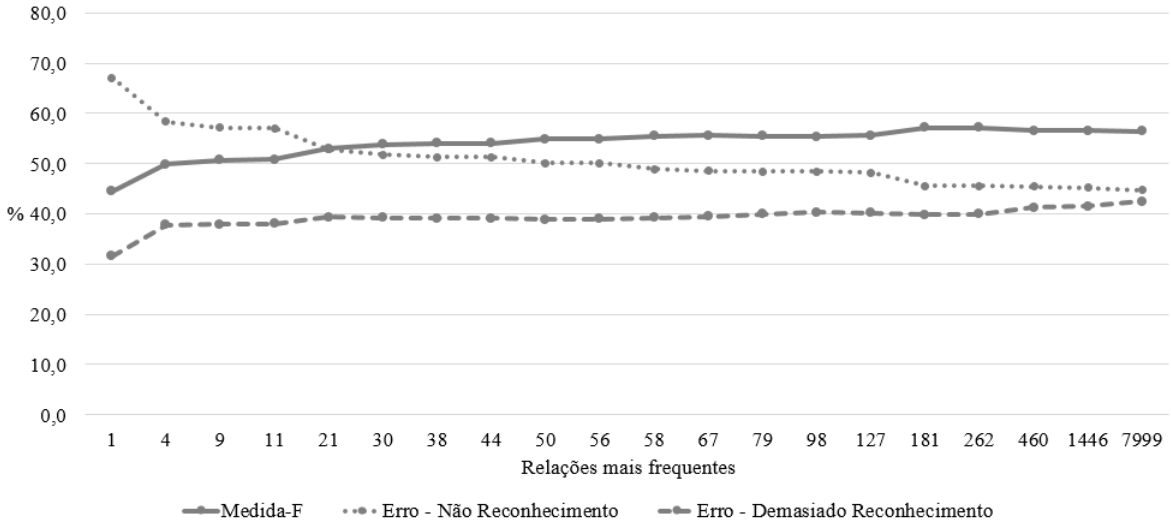

Figura 7.16: Resultados obtidos na extração de aspetos para a Língua Portuguesa.

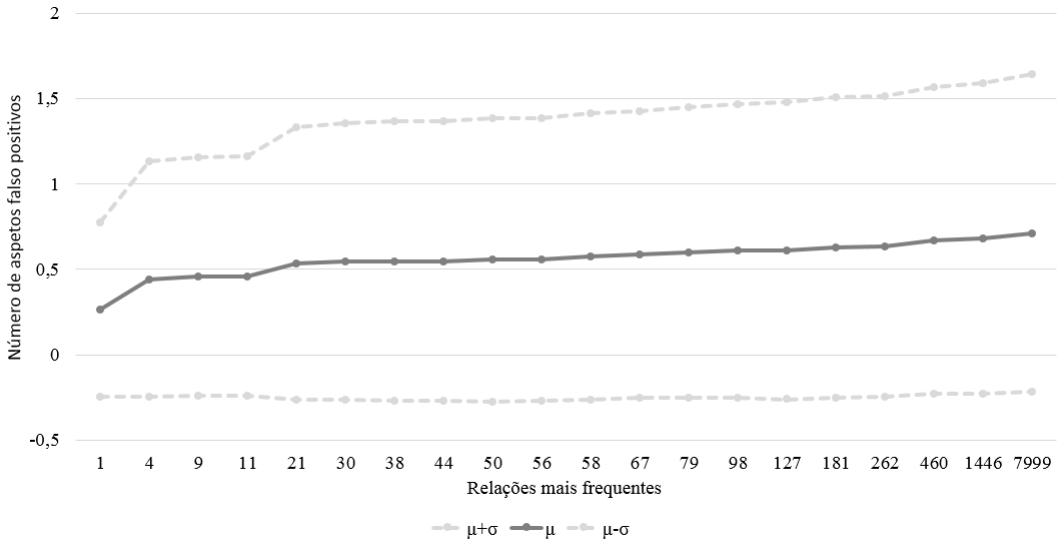

Figura 7.17: Média de número de aspetos falso positivos.

É possível verificar que se usarmos apenas a relação mais frequente obtemos 44.5% de Medida-F, no entanto o Erro de Não Reconhecimento é bastante alto (cerca de 67%). Há medida que se acrescenta mais relações a Medida-F vai aumentando ligeiramente, sendo que o erro não reconhecimento vai baixando. Já o erro de Demasiado Reconhecimento que tem um mínimo de 31% com apenas uma relação, à medida que novas relações são acrescentadas não sofre grandes alterações. Ou seja, à exceção do primeiro ponto do gráfico o erro de Demasiado Reconhecimento tem um acréscimo de apenas 5% desde que se usa quatro relações até usar as 7999.

É possível observar que desde as 30 relações até as 127 relações não há grandes alterações da Medida F (subida de 2%), no entanto o erro de não reconhecimento baixa quase 4%. Usando todas as relações extraídas o erro de não reconhecimento e o erro de demasiado reconhecimento acabam por tender para o mesmo resultado.

Dado os resultados, usar apenas a relação mais frequente não é uma solução uma vez que o erro de não reconhecimento é bastante alto, e a partir das quatro relações tem uma grande descida de quase 10%. Como tal, decidiu-se que o número de relações ideal seria 181 relações. Nesse ponto consegue uma descida do erro de não reconhecimento para 45.5% o que é muito perto do mínimo conseguido (47%). É também nesse ponto onde a Medida-F atinge o seu máximo de 57%.

No segundo teste realizado que tinha como objetivo perceber quais as restrições que se deviam colocar para a extração de aspetos, foram obtidos os resultados apresentados na tabela 7.23. Os resultados provêm do seguinte conjunto de testes:

- **Teste A** Este teste simula a opção de não existir qualquer tipo de restrições aos aspetos.
- **Teste B** Para esta teste, todos os aspetos candidatos que sejam palavras de opinião ou *stopwords* são excluídos.
- **Teste C** Para este teste, para além da restrição adicionada no Teste B, são usadas as restrições usando o *Theasaurus* e Corretor Ortográfico.
- **Teste D** Neste teste é adicionado as restrições que obrigam a que um aspeto não possa ter símbolos ou números.
- **Teste E** Para este teste, para além das restrições aplicadas nos testes anteriores, são usados os dicionários de referências temporais. Ou seja, um aspeto não pode ser uma palavra existente nesse dicionário.
- **Teste F** Neste teste, é adicionada a restrição de que o aspeto candidato não pode ser um substantivo, adjetivo ou verbo comum.
- **Teste G** Este teste, é adicionada a análise sobre verbos, em que dado um aspeto candidato è analisado se este contém sufixos verbais e possui um modo infinitivo. Embora todos os aspetos que sejam candidatos já tenham sido classificados como substantivos, esta restrição permite precaver de eventuais erros no identificador de classes gramaticais.
- **Teste H** Neste teste, é adicionada a restrição de que se aspeto candidato candidato quando processado individualmente não for considerado como um substantivo é excluído. Neste teste, todas as restrições e análises desenvolvidas estão incluídas.
- **Teste I** Neste teste, ao teste anterior, que inclui todas as análises e restrições foi removido a funcionalidade de correção de acentos.
- **Teste J** Por fim, este teste inclui todas as restrições desenvolvidas à exceção da restrição que engloba excluir aspetos candidatos que possuam um modo infinitivo.

Como se pode observar diferença entre não ter qualquer restrição (teste A) e ter as restrições mais básicas de *stopwords* e palavras de opinião (Teste B) é de cerca de 5%, o que mostra que a fase de análise e limpeza é bastante importante. Entre esses dois teste o erro de não reconhecimento aumenta 1% no entanto o erro de demasiado reconhecimento tem uma descida de 5%. Outra conclusão possível de fazer é que quando se acrescenta a

Métricas	Teste A	Teste B	Teste C	Teste D	Teste E	Teste F	Teste G	Teste H	Teste I	Teste J
Medida-F (%)	35.1	40.2	42.5	42.6	46.2	56.2	56.8	57.4	48.0	58.2
Erro - Não Reconhecimento (%)	23.7	24.8	27.2	27.2	27.3	39.6	45.5	45.5	45.7	39.8
Erro - Demasiado Reconhecimento (%)	77.1	72.5	69.8	69.8	66.1	46.8	40.6	39.3	56.9	43.5

Tabela 7.23: Testes ao tipo de limpeza a usar na extração de aspetos.

restrição sobre símbolos ou números (Teste D) a melhoria é insignificante, o que quer dizer que há partida já não existiam aspetos candidatos com essas condições.

A restrição que maior diferença fez foi quando se adicionou os dicionários de substantivos, adjetivos e verbos comuns, no Teste F. A diferença entre o Teste E e F é de 10%. O teste F apresentou uma descida de cerca de 20% no erro de demasiado reconhecimento, no entanto o erro de não reconhecimento aumentou cerca de 12%, o que pode querer dizer que os dicionários de palavras comuns contém palavras que em algumas situações podem ser aspetos.

Comparando o Teste H e o Teste I é possível observar que a correção ortográfica dos acentos nas palavras é muito importante, uma vez que na sua ausência a Medida-F desce quase 10%.

Por fim, comparando o Teste H e o Teste J, cuja diferença é apenas não usar a restrição do aspeto possuir um modo infinitivo, a melhoria é significante, conseguindo uma subida na Medida-F de quase 1%. No entanto, os valores mais relevantes estão no erro de não reconhecimento de que se mantém ao nível do Teste F (antes de acrescentar as restrições do sufixo e modo infinitivo). Já o erro de demasiado reconhecimento é um pouco acima do obtido no Teste H.

Dados os resultados há dois conjuntos de restrições que podem ser consideradas: o conjunto simulado no Teste H e o conjunto simulado no Teste J. A grande diferença entre os dois é que o Teste H possui um erro de não reconhecimento mais elevado (diferença de cerca de 6%), no entanto o erro de demasiado reconhecimento é mais baixo (diferença de cerca de 4%). Como tal, foi decidido que era mais importante um valor mais baixo de falsos positivos do que um valor mais baixo de falso negativos. Ou seja, é preferível ter menos erros nos aspetos que extraímos do que extrair todos os aspetos que devíamos. Como tal, optou-se pelas restrições simuladas na Teste H, cujos resultados estão apresentados em pormenor na tabela 7.24.

Total de Aspetos	1770
Precisão	60.6%
Abrangência	54.4%
Medida-F	57.4%
Erro - Não Reconhecimento	45.5%
Erro - Demasiado Reconhecimento	39.3%
Média de Aspetos Falso Positivos por frase	0.62 ($\sigma = 0.88$)

Tabela 7.24: Resultados para a ferramenta de extração de aspetos para a Língua Portuguesa.

Teste de Performance

O teste de performance realizado permite analisar o tempo necessário para extrair os aspetos dando uma determinada frase. Para este teste foi usado o mesmo corpus usado para os testes de qualidade já descrito em cima.

Resultados e Análise

Na Figura 7.18 é possível visualizar o tempo necessário para a extração de aspetos dependendo do tamanha de cada frase. É possível observar que o tamanho da frase influencia o tempo preciso para extrair os aspetos. Pode-se ver que numa frase muito curta a ferramenta precisa apenas de 6.7 milissegundos para extrair os aspetos, no entanto numa frase muito longa necessita de 99 milissegundos.

Comparando com a mesma ferramenta para a Língua Inglesa, percebe-se que apesar dos tempos serem aceitáveis estão significativamente superiores. Por isso, foi feita uma análise mais rigorosa e concluiu-se que 99% do tempo é gasto na construção dos *chunks* que no português são construídos usando uma abordagem diferente que obriga a construção da árvore de constituência.

É de notar, no entanto, que no corpus cerca de 58% das frases tem entre 40 a 160 palavras, o que pode representar que as frases mais frequentes são desse tamanho o que precisa de no máximo 27 milissegundos para a extração.

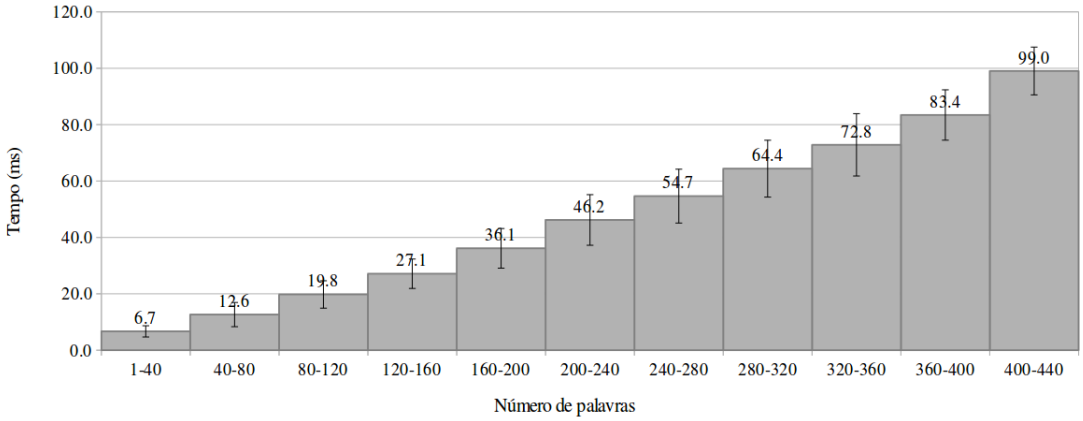

Figura 7.18: Tempo de extração de aspetos consoante o tamanho da frase.

7.3.3 Extração de Entidades

De forma a avaliar a ferramenta de extração de entidades foram efetuados alguns testes que avaliam duas componentes diferentes: a qualidade da ferramenta e a performance. Neste secção são apresentados e analisados os resultados obtidos.

Testes de Qualidade

O primeiro teste de qualidade realizado tem como objetivo perceber qual é o conjunto de relações de dependência que melhores resultados apresenta. Já o segundo teste permite perceber quais são as restrições, da fase de análise e limpeza das entidades candidatas, que produzem uma ferramenta com melhores resultados. Tal como em todas as ferramentas já descritas, todos os testes de qualidade foram repetidos 30 vezes de forma a obter resultados estatísticamente relevantes.

Os resultados de ambos os testes são medidos usando diferentes métricas como a Medida-F, o Erro de Não Reconhecimento e o Erro de Demasiado Reconhecimento que já foram descritas na secção 7.2.2.

Por fim, para a realização dos testes de qualidade foi desenvolvido um corpus com frases extraídas do Facebook da Vodafone Portugal[16] que é composto por 999 frases. Cada instância ou frase do corpus contém em média 1.02 entidades, sendo que no máximo temos 7 entidades e no mínimo temos 0 entidades por frase, como se pode ver na Figura 7.19.

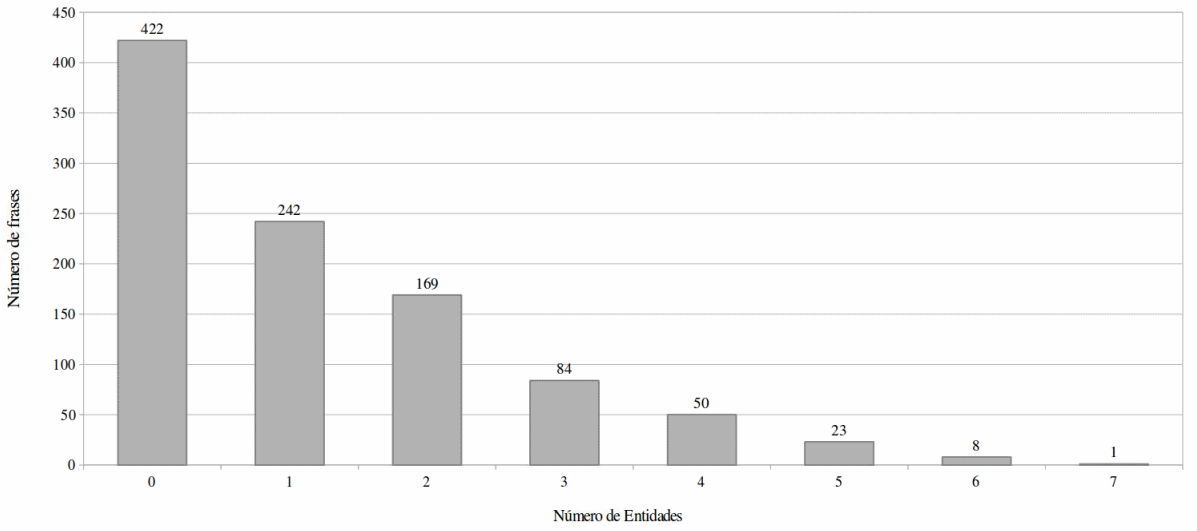

Figura 7.19: Distribuição do número de entidades no corpus de teste da extração de entidades para a Língua Portuguesa.

Resultados e Análise

Na Figura 7.20 está representada a evolução da ferramenta à medida que novas relações de dependência são adicionadas. Na Figura 7.21 é possível visualizar a influência que novas relações tem no número de entidades falso positivas. Tal como todos os testes anteriores, todas as repetições dos testes os resultados permaneciam iguais o que representa que todos os resultados tem um desvio padrão de 0.

Como se pode observar à medida que acrescentamos relações de dependência a Medida-F vai subindo significativamente, com uma diferença entre o mínimo e o máximo obtido de mais de 40%. Usando apenas a relação mais frequente obtém-se uma Medida-F muito baixa (cerca de 15%), sendo que se possui um erro de não reconhecimento muito alto de cerca de 90%. Seria de esperar que quanto mais relações se usam menor o erro de não reconhecimento é que é o que acontece. O erro de demasiado reconhecimento à exceção dos primeiros resultados mantém-se estável, mostrando-se menos dependente do número de relações usadas.

Dado os resultados, conclui-se que usar poucas relações de dependência não é solução uma vez que produz uma Medida-F mais baixa e um erro de não reconhecimento significativamente alto. Como tal, decidiu-se que o melhor conjunto de relações era se usarmos todas as relações conhecidas, ou seja as 666 relações de dependência. Usando todas as relações de dependência obtemos o melhor resultado obtido na Medida-F, com 56.6% e o erro de não reconhecimento no seu mínimo com 44.8%. Já o erro de demasiado reconhecimento, embora não seja o valor mínimo obtido, é muito próximo com uma diferença de apenas 0.5%. Assim ficamos com uma média de entidades falso positivas por frase de cerca de 0.48 que, embora seja o valor mais alto, ainda assim é um valor aceitável.

[16] Disponível em `https://www.facebook.com/vodafonePT`

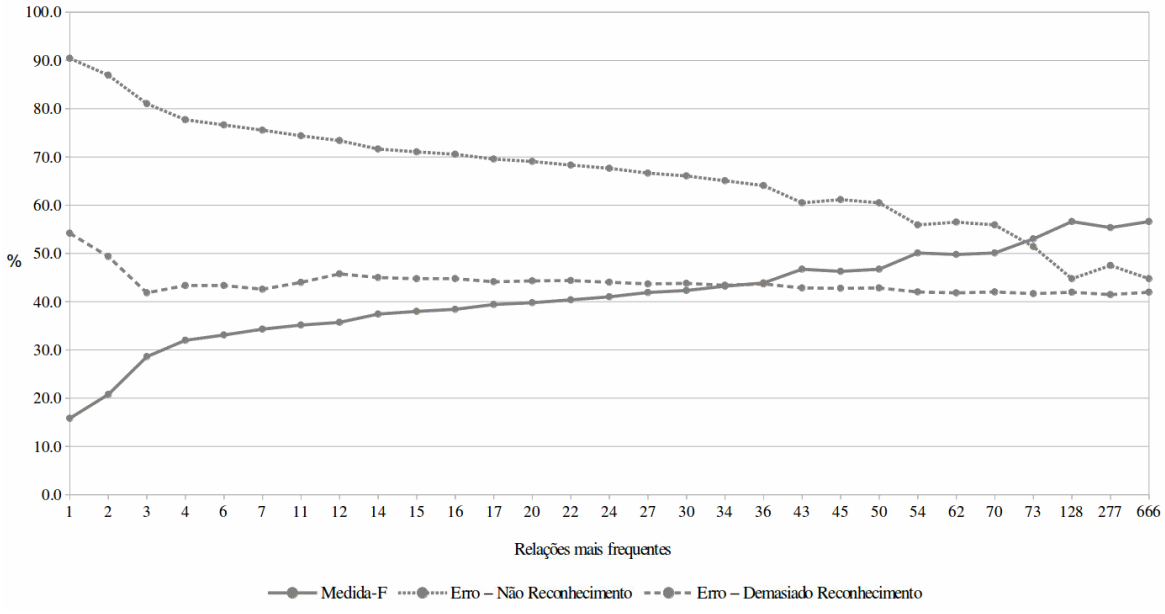

Figura 7.20: Resultados obtidos na extração de entidades para a Língua Portuguesa.

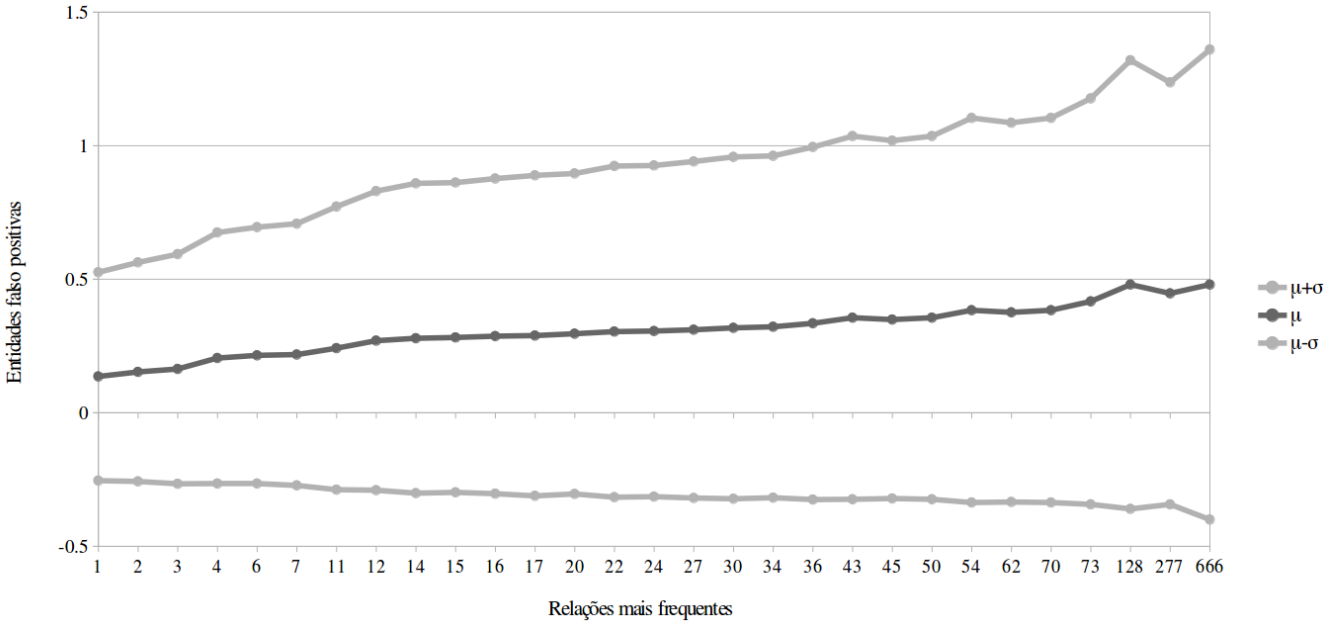

Figura 7.21: Média de número de entidades falso positivos.

Tal como explicado, o segundo teste tinha como objetivo perceber quais as restrições que se deviam colocar para a extração de entidades, e como tal foi realizado o seguinte conjunto de testes:

- **Teste A** Este teste simula a opção de não existir qualquer tipo de restrições às entidades.
- **Teste B** Para esta teste, todos as entidades candidatas que sejam palavras de opinião, *stopwords* ou palavras especiais como *urls*, menções ou *hashtags* são excluídos.
- **Teste C** Neste teste, os dicionários de referências temporais, nomes de pessoas, cidades e países foram adicionados. Todas as entidades candidatas que estejam

presentes nesses dicionários são excluídas.

- **Teste D** Para este teste, às restrições dos testes anteriores foram adicionados os dicionários de substantivos, adjetivos e verbos mais comuns e palavrões. Se a entidade candidata estiver presente num dos dicionários é excluída.
- **Teste E** Para este teste, foi adicionada uma nova restrição que permite perceber se a entidade candidata está escrita em inglês e se estiver se for um aspeto conhecido é excluída.
- **Teste F** Neste teste, para além de todas as restrições dos testes anteriores se a candidata for composta por símbolos, números ou for uma palavra de referência numérica esta é excluída.
- **Teste G** Neste teste, foi acrescentada a análise de verbos, que restringe que a entidade candidata não pode ter um sufixo verbal nem um modo infinitivo.
- **Teste H** Neste teste, se a entidade candidata quando processada isoladamente não for considerada um substantivo é excluída. Também foi adicionada a restrição de que se a palavra possuir o sufixo *-ção* que é utilizada nos substantivos derivados de verbos relacionados com ação a palavra é excluída da lista de entidades candidatas.
- **Teste I** Neste teste, todas as restrições desenvolvidas estão presentes. OU seja, ao teste anterior è adicionado a análise sobre o corretor ortográfico. Se o corretor ortográfico detectar que a candidata não é uma palavra conhecida e devolve um conjunto de palavras possíveis. A esse conjunto de palavras semelhantes se alguma delas for um aspeto conhecido a entidade é excluída.
- **Teste J** Por fim, neste teste todas as restrições desenvolvidas estão presentes expecto as restrições acrescentadas no Teste E que tenta perceber se a entidade candidata é um aspeto inglês conhecido.

Na tabela 7.25 é possível observar os resultados obtidos nos diferentes testes.

Métricas	**Teste A**	**Teste B**	**Teste C**	**Teste D**	**Teste E**	**Teste F**	**Teste G**	**Teste H**	**Teste I**	**Teste J**
Medida-F (%)	27.6	28.5	30.8	43.3	41.4	46.35	53.34	55.72	56.6	57.8
Erro - Não Reconhecimento (%)	29.0	30.9	31.2	35.1	39.2	43.09	44.3	44.75	44.75	42.6
Erro - Demasiado Reconhecimento (%)	82.8	81.9	80.1	67.4	68.4	60.89	48.77	43.77	41.95	41.6

Tabela 7.25: Testes ao tipo de restrições a aplicar na extração de entidades.

Como se pode observar, a diferença entre os três primeiros testes não é muita (apenas cerca de 3% melhor na Medida-F) . Ou seja, pode-se concluir que usando apenas as relações sem qualquer restrição já muitas *stopwords*, palavras de opinião, referências temporais e nomes de pessoas, cidades ou países são excluídas, por isso as restrições impostas no Teste B e C não apresentam diferenças muito significativas.

A maior diferença regista-se entre o Teste C e o Teste D, com uma melhoria de cerca de 13% na Medida-F. Ou seja, os dicionários de substantivos, adjetivo e verbos comuns é a restrição com uma maior valia, conseguindo descer assim o erro de demasiado reconhecimento mais de 13%. Já o erro de não reconhecimento sobre quase 4% no entanto o ganho é maior uma vez que reduz significativamente o número de falso positivos.

Outra restrição que aparenta ter uma grande importância é a análise dos verbos que é adicionada no Teste G. Nesse teste, a Medida-F sobe cerca de 7%, sendo que o erro de demasiado reconhecimento tem uma descida muito significativamente de cerca de 12%.

Entre o Teste D e E observa-se uma pequena descida na Medida-F (cerca de 2%), e um aumento de erros tanto para o erro de não reconhecimento como para o erro de demasiado reconhecimento. No Teste E é acrescentada a restrição que se a palavra for uma palavra inglesa e se for reconhecida como um aspeto inglês esta é removida. Dado os resultados, foi acrescentado o Teste J que simula todas as restrições desenvolvidas à exceção desta restrição. Como se pode observar, comparando o Teste I e o Teste J o sistema melhorou cerca de 1%, sendo que a maior melhoria registou-se a descida do erro de não reconhecimento em cerca de 2%.

Por fim, dado os resultados foi decidido que o conjunto de restrições que melhor sistema criava era o simulado no Teste J, cujos resultados em detalhe estão apresentados na tabela 7.26.

Total de Entidades	1202
Precisão	58.3%
Abrangência	57.3%
Medida-F	57.8%
Erro - Não Reconhecimento	42.6%
Erro - Demasiado Reconhecimento	41.6%
Média de Entidades Falso Positivas por frase	0.5 ($\sigma = 0.86$)

Tabela 7.26: Resultados para a ferramenta de extração de entidades para a Língua Portuguesa.

Um dos problemas detetados nesta ferramenta foram os erros derivados do uso de outras ferramentas, como o identificador de classes gramaticais. Por exemplo, na seguinte frase:

"*Já agora,o Vodafone smart 4 turbo ira ter actualizações futuras?*"

Os únicos substantivos detetados são "*Vodafone*" e "*actualizações*". Uma vez que "*smart*" é considerado um verbo pelo identificador de classes gramaticais a entidade complexa "*Vodafone smart 4 turbo*" não é detetada. Nesse exemplo apenas a entidade "*Vodafone*" é extraída. É de notar que este tipo de texto incluí muitas palavras que não são reconhecidas pelo dicionário da Língua Portuguesa o que pode dificultar estas análises base.

Um outro exemplo de erros no identificador de classes gramaticais é visivel no seguinte exemplo:

"*agradeço que me parem de ligar nao estou interessa na tv net e voz nem no red*"

Neste exemplo, a palavra "*agradeço*" é identificada como sendo um substantivo em vez de um verbo. Uma vez que é detetada como um substantivo, quando se construi a árvore de dependências a palavra "*agradeço*" está diretamente dependente do verbo "*estou*" por uma relação do tipo SUBJ que é a relação mais frequente para extrair entidades candidatas, por isso "*agradeço*" é considerado uma entidade. Como se pode perceber o erro cometido pelo identificador de classes gramaticais é propagado para a construção da árvore de dependências o que influencia a ferramenta a extrair erradamente uma entidade

que na realidade devia ter sido classificada como um verbo e por isso excluída logo como candidata.

Um outro problema detetado é que nos textos extraídos, muitas vezes as frases eram mal detetadas pelo identificador de frases, como por exemplo no texto:

" *eu que sempre aderi as promoçoes yorn e vodafone....ha muito que nao existe promoçoes de bonus em caregamento.....*"

Quando a ferramenta de separação de frases não consegue identificar as frases de forma correta o erro propaga-se para toda a ferramenta, por exemplo se a frase estiver mal identificada, o identificador de classes gramaticais tem mais dificuldades produzindo mais erros assim como a construção da árvore de dependências.

Por fim, é de notar que esta tarefa não é uma tarefa simples nem para nós os humanos. Existem entidades que são discutíveis, por exemplo na frase:

" *Aqui em casa, acabámos de enviar um email à Vodafone a manifestar a nossa vontade de cancelar a subscrição dos canais da Sport TV, visto que está a dar erro.* "

Neste exemplo as entidades extraídas são "*Vodafone*" e "*Sport TV*", no entanto é fácil de perceber que a publicação se concentra na entidade "*Sport TV*", sendo esta a mais relevante. Pode ser discutível se a "*Vodafone*" deve ser considerada também como uma entidade. No entanto, mesmo não sendo o tema principal do texto, existe um interação com a "*Vodafone*", que neste caso foi apenas enviar um email que pode ser insignificante para este texto especifico, no entanto pode ser na mesma uma entidade.

Teste de Performance

Tal como para a ferramenta anterior o teste de performance efetuado tem como objetivo analisar o tempo necessário para a extração de entidades a partir de uma frase. Para a realização deste teste foi usado o mesmo corpus usado para os testes de qualidade já descrito em cima.

Resultados e Análise

Na Figura 7.22 é possível observar o tempo necessário para a extração de entidades consoante os diferentes tamanhos das frases. Como era expectável quanto maior for a frase, mais tempo é necessário. Isso deve-se ao facto de que quanto mais palavras mais tempo ser necessário para construir as árvores de dependência e constituência.

Comparando com a mesma ferramenta para a Língua Inglesa, percebe-se que apesar dos tempos serem aceitáveis, estão significativamente superiores. Como tal foi feita uma análise mais rigorosa e conclui-se que mais de 50% do tempo é gasto na construção dos *chunks* e mais de 10% do tempo é gasto na construção da árvore de dependências. Todos o resto do tempo é gasto no pré-processamento da frase e na análise de da árvore de dependências.

7.3.4 Extração de Quíntuplos

Tal como para a mesma ferramenta em Inglês, foram realizados dois tipos de teste de forma a avaliar a ferramenta de extração de quíntuplos: testes de qualidade e testes de performance. Nesta secção cada um deles é descrito em detalhe e são apresentados e analisados os resultados de cada um.

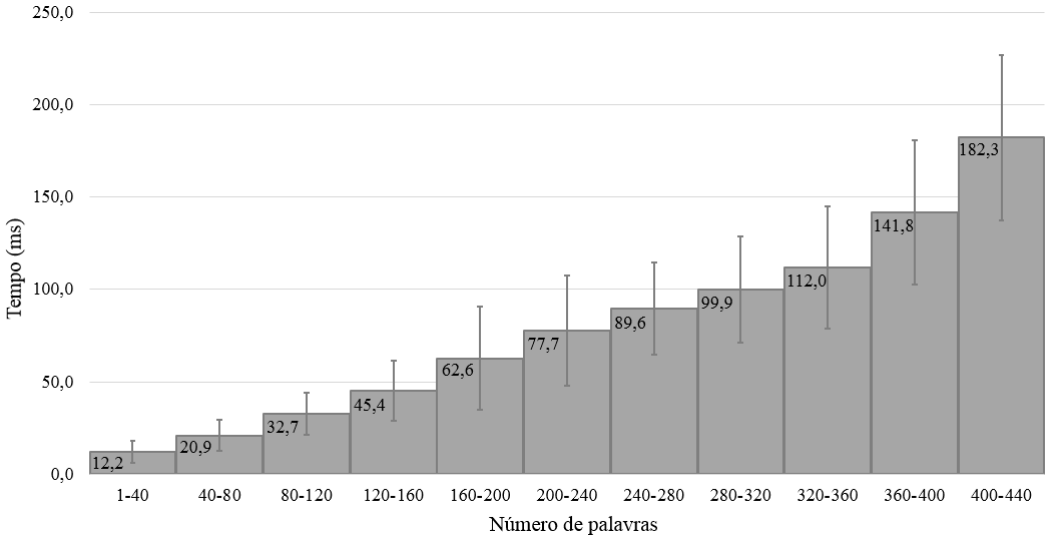

Figura 7.22: Tempo de extração de entidades consoante o tamanho da frase.

Testes de Qualidade

De forma a avaliar a ferramenta foi realizado um teste, cujo objetivo é perceber se através de um conjunto de entidades e aspetos dados a ferramenta é capaz de produzir quíntuplos de forma corretamente. Para tal, foi desenvolvido um corpus com cerca de 600 instâncias. Na Figura 7.23 é possível observar a distribuição presente no corpus tendo em conta o número de quíntuplos presentes em cada instância. É possível observar que 54% do corpus é composto por instâncias que apenas possuem um quíntuplo anotado, e que cerca de 80% possuem até dois quíntuplos. É de salientar que para este teste apenas foi testado a relação entre entidades e aspetos, ou seja, para cada uma das instâncias de teste foram fornecidas à ferramenta quais eram as entidades e os aspetos e a ferramenta construída os quíntuplos usando apenas essa informação.

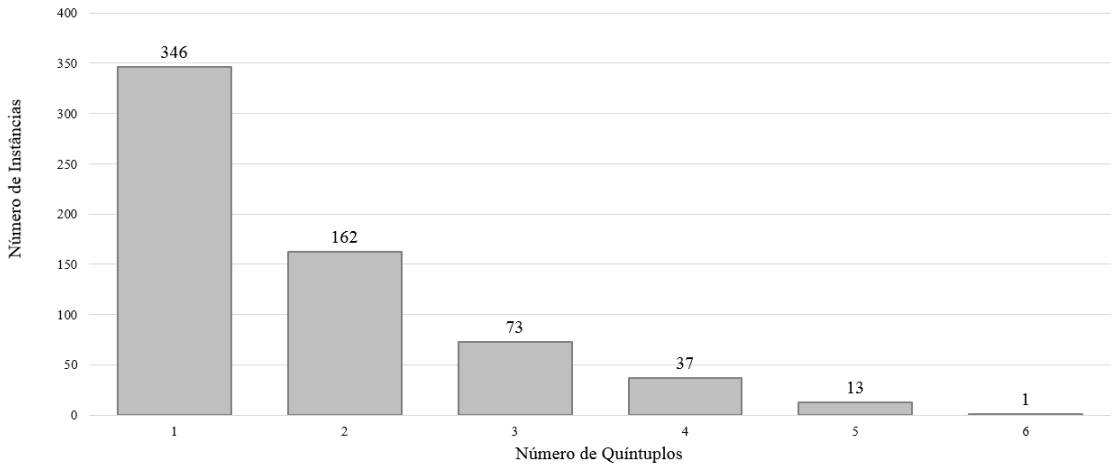

Figura 7.23: Distribuição dos quíntuplos no corpus de teste da extração de quíntuplos para a Língua Portuguesa.

A métrica usada para avaliar a ferramenta é a abrangência já mencionada anteriormente na equação 2.6.

Resultados e Análise

Na tabela 7.27 é possível observar o resultado obtido no teste efetuado. De todos os quíntuplos, a ferramenta é capaz de os identificar de forma correta cerca de 76% das vezes.

Total de Quíntuplos	1108
Abrangência	76.6%

Tabela 7.27: Resultados para a ferramenta de extração de quíntuplos para a Língua Portuguesa.

Tal como os resultados obtidos para a mesma ferramenta em Inglês, a ferramenta apresenta melhores resultados nas instâncias que têm apenas um quíntuplo (cerca de 91%). Quanto mais quíntuplos uma instância tem, mais erros a ferramenta apresenta. Por exemplo nas instâncias com três quíntuplos a ferramenta acerta 65%. Nas instâncias com cinco quíntuplos a ferramenta acerta apenas 38% desses quíntuplos. É de notar que no corpus usado apenas 2% das instâncias tem cinco quíntuplos, o que pode mostrar que essas situações não são muito frequentes.

De forma a testar a usabilidade da ferramenta, foram aglomeradas mais cerca de 10.000 frases extraídas do Facebook da Vodafone, e foram extraídos de forma automática os quíntuplos de cada frase. Na tabela 7.28 estão representadas as 3 entidades mais frequentes e os seus aspetos mais frequentes associados a uma polaridade média.

Analisando a primeira entidade, a "*Vodafone*", esta tem associada o aspeto "*Operador*" e "*Empresa*" como sendo aspetos diferentes, no entanto é discútivel se esses dois aspectos deveriam ser considerados o mesmo, obrigando a que a análise de polaridade tenham em conta os dois. Ou então, relacionado com abreviaturas, também o aspeto "*TV*" e "*Televisão*". O mesmo problema também acontece com entidades, em que para a mesma entidade existem diferentes formas em que esta é descrita. Por exemplo, a entidade "*Vodafone*" e a entidade "*Vodafone Portugal*".

Usando as informações da tabela podemos concluir que na generalidade as publicações têm, geralmente, uma conexão negativa. Sendo que as opiniões mais uníssonas são sobre a subscrição e a velocidade associada à entidade "*Box*", e a velocidade associada à entidade "*Internet*".

Teste de Performance

De forma a avaliar a ferramenta sobre a sua performance foi realizado um teste que calcula o tempo necessário para a extração de quíntuplos desde a extração de entidades e aspetos até ao cálculo da polaridade de cada quíntuplo.

Resultados e Análise

Como se pode observar na Figura 7.24 a fase que demora mais tempo é a fase de extração do texto relevante para cada quíntuplo. Todas as fases vão necessitando de mais tempo à medida que cresce o número de palavras na frase. Em média, numa frase mais pequena são necessários 0.15 segundos enquanto que numa frase com muito mais palavras são necessários cerca de 1.5 segundos.

Entidade Vodafone		Entidade Box		Entidade Internet	
Aspeto	**Polaridade**	**Aspeto**	**Polaridade**	**Aspeto**	**Polaridade**
General	-0.4	General	-0.3	General	-0.6
Clientes	-0.5	Subscrição	-1.0	Serviço	-0.4
Serviço	-0.5	Canais	-0.8	Cliente	-0.3
Operadora	-0.7	Clientes	0.8	Velocidade	-1.0
Loja	-0.5	Serviço	-1.0	Fibra	-0.7
Canais	-0.8	Campanha	-0.1	Localidade	-0.9
Telemóvel	-0.6	Preço	-0.7	Canais	-0.1
Instalação	-0.4	Velocidade	-1.0	Televisão	0.3
Empresa	0.2	Net	-0.7	Versões	-0.2
Tarifários	0.0	Ligação	-0.8	Cartões	0.1
Fidelização	-0.5	App	-0.1	Instalação	0.0
Adesão	-0.25	Pagamento	-0.8	Aplicação para Computador	-0.8
Rede	0.0	TV	0.3	Sistema	-0.3

Tabela 7.28: Resultados produzidos pela ferramenta de extração de quíntuplos da Língua Portuguesa.

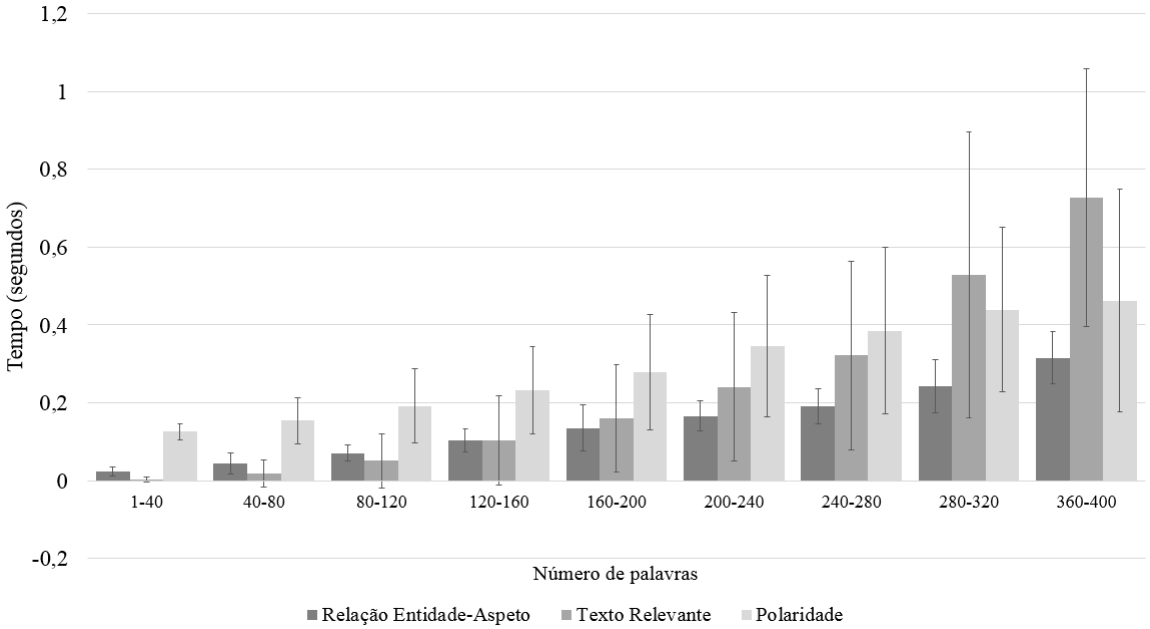

Figura 7.24: Tempo médio para cada fase por tamanho da frase.

Capítulo 8

Conclusão

A crescente utilização das redes sociais trouxe diferentes desafios ao estudo da linguagem natural. A falta de contexto e a presença de erros ortográficos são algumas das características que tornam difícil o processo de interpretação e extração de conhecimento de texto. Tanto a área de Processamento de Linguagem Natural como de *Text Mining* apresentam diversas ferramentas que pretendem resolver muitos desses problemas, sendo que este trabalho se concentra apenas na Extração de Opiniões.

O desenvolvimento de ferramentas de Extração de Opiniões apresenta diversos desafios, como por exemplo a fase de Extração de Polaridade, que é muitas vezes controversa uma vez que, em alguns casos, não existe um consenso de qual a correta polaridade a associar, tornando esta tarefa um pouco subjetiva. A construção destas ferramentas recorre, diversas vezes, a tarefas de Processamento de Linguagem Natural (PLN), como por exemplo a identificação de classes gramaticais. No entanto, a implementação dessas tarefas não foi o objetivo deste trabalho, uma vez que a *Wizdee* já possui uma biblioteca de tarefas base de PLN.

O trabalho apresentado neste documento teve como objetivo a criação de uma ferramenta que permitisse extrair informação de texto subjetivo. Mais concretamente, o objetivo foi criar uma ferramenta capaz de perceber o alvo de uma opinião, ou seja a sua entidade, perceber qual a característica especifica que a opinião retrata, ou seja o seu aspeto, e também a sua polaridade, ou seja se é uma opinião positiva, negativa ou neutra. No contexto da Wizdee, essas ferramentas têm de estar preparadas para texto de redes sociais, cuja opinião se refere a marcas ou produtos, e devem suportar a Língua Portuguesa e a Língua Inglesa.

Como tal, foi desenvolvido um conjunto de ferramentas como: ferramenta de extração de aspetos, ferramenta de extração de entidades e a ferramenta de extração de quíntuplos, que inclui as seguintes informações: Autor, Data, Entidade, Aspeto e Polaridade. Para todas essas ferramentas foi seguida uma abordagem linguística, onde são estudadas as relações de dependência entre as palavras numa frase. Essas relações permitem de diferentes formas extrair as informações necessárias para se perceber, por exemplo, a opinião que os clientes de determinada empresa têm acerca de um dos seus produtos, ou até das diferentes características do mesmo produto.

Um dos maiores desafios neste tipo de tarefas foi o tipo de texto que se estava a analisar. Muitas vezes, estava escrito de forma incorreta e com palavras dificilmente percetíveis. Todos esses aspetos provocam uma dificuldade no processamento quando são usadas ferramentas como o identificador de classes gramaticais ou o lematizador, que estão preparados para frases construídas corretamente. Naturalmente, todos os erros produzidos nessas ferramentas são propagados para outras análises, como por exemplo, quando se constrói a árvore de dependências, que se baseia em informações básicas como as classes gramaticais.

Este problema é reconhecido na área de PLN e como tal, começam a existir algumas ferramentas especializadas neste tipo de texto, mais concretamente para texto extraído do *Twitter*, como por exemplo identificador gramatical (Gimpel et al., 2011) ou *parser* de dependências (Kong et al., 2014). Foram realizados diversos testes que permitiram perceber qual o nível de qualidade da ferramenta. Durante o desenvolvimento, também se realizaram testes que ajudaram a tomar decisões de forma a construir uma ferramenta que produz melhores resultados. Os diferentes testes refletiram uma das maiores dificuldades, que foi reduzir o número de falso positivos na ferramenta de extração de aspetos e entidades, tanto para a Língua Inglesa como para a Portuguesa. No final, é feita uma análise de texto de redes sociais relacionado com duas empresas, a *Vodafone* e a *Samsung*, para o Português e Inglês, respetivamente, onde são apresentadas as informações extraídas. Desta informação é possível perceber quais são os produtos ou marcas que mais queixas apresentam, e quais são as características que mais descontentamento provocam. Por exemplo, no caso da *Samsung* consegue-se perceber que as atualizações de software aos telemóveis *Galaxy S4* e *Galaxy S5* provocam muitos comentários negativos e que a bateria também é muito criticada.

Quanto à ferramenta de extração de polaridade, foi escolhida uma abordagem de aprendizagem automática usando as Máquinas de Vetor de Suporte. Foram criados diversos recursos que foram usados para a fase de transformação dos textos em vetores de características. Para o Inglês, os resultados obtidos conseguem competir com os alguns dos melhores resultados obtidos noutros trabalhados. Apesar de em alguns corpus, a ferramenta, ter uma Medida-F um pouco mais abaixo do que a melhor equipa do *SemEval-2014*, noutros corpus a ferramenta desenvolvida obtém melhores resultados.

Naturalmente, uma das maiores dificuldades foi a falta de recursos que podem ser usados num contexto comercial, que se pronunciou mais na Língua Portuguesa e que se refletiu nos resultados alcançados pelas ferramentas. Uma outra dificuldade, é o nível de subjetividade que a classificação de polaridade apresenta, em que diferentes pessoas podem interpretar de forma diferente a mesma frase. Por fim, o uso da ironia, muitas vezes presente, tornou-se um grande desafio, no entanto, é de salientar que muitas vezes é, até para nós, humanos, difícil conseguir identificar esses casos.

De uma forma geral, pode-se concluir que os resultados obtidos nas diferentes ferramentas são bastante razoáveis tendo em conta as limitações existentes. Por exemplo, na ferramenta de extração de polaridade para o Inglês foi alcançada uma Medida-F de cerca de 63%, que compete com os resultados obtidos no *SemEval-2014* com uma diferença de apenas dois pontos percentuais. Já na mesma ferramenta mas para o Português alcançou-se uma Medida-F de 59%, um pouco mais baixa comparando com o Inglês no entanto, é de notar que o Português apresenta muito mais desafios. Nas ferramentas de extração de aspetos foi obtida uma Medida-F de 67% e 57%, para a Língua Inglesa e Portuguesa, respetivamente. Por outro lado, nas ferramentas de extração de entidades foi alcançada uma Medida-F, muito semelhante à ferramenta de extração de aspetos, de 65% e 57%, para a Língua Inglesa e Portuguesa, respetivamente. Por fim, na ferramenta de extração de quíntuplos foi obtida uma abrangência de 86% na Língua Inglesa e 76% na Língua Portuguesa.

Sendo assim, as contribuições conseguidas com este trabalho são as seguintes:

- Elaboração do Estado da Arte no domínio do Processamento de Linguagem Natural e *Text Mining*, com principal foco na análise de opiniões.
- Foi desenvolvida uma biblioteca que permite a extração da polaridade de uma opinião, com origem de uma rede social, que pode ser positiva, negativa ou neutra, tanto para Inglês como para Português.
- Foi desenvolvida uma biblioteca que permite a extração de informações relevantes

sobre uma opinião, como as entidades (que podem ser marcas, produtos ou serviços), e os aspetos (características das entidades), tanto para Inglês como para Português.

- Foi construído um conjunto de dicionários relevantes para cada ferramenta desenvolvida, para ambas as línguas suportadas.
- Foi desenvolvida uma ferramenta que permite construir uma árvore de dependências para a Língua Portuguesa.
- Experimentação realizada nas diferentes ferramentas, e o desenvolvimento de diferentes corpora anotados manualmente e desenvolvidos para cada ferramenta.
- Integração de todas as ferramentas com a plataforma Wizdee, pronto a ser utilizado em projetos comerciais. É de notar que à data deste documento está a ser inicializado um projeto com uma empresa que usa as ferramentas desenvolvidas.

Trabalho Futuro

As ferramentas desenvolvidas são já uma base de trabalho interessante, no entanto nesta área existem sempre novos rumos a explorar e novas abordagens. Por exemplo, na ferramenta de extração de aspetos, para além de extrair os aspetos explícitos, podem-se extrair os implícitos. Por exemplo, na frase "*O Samsung S5 é muito caro*", a palavra "caro" não é visto como uma aspeto, no entanto, no futuro podem-se desenvolver análises que permitam extrair que o aspeto é o preço. Ou seja, a partir de uma caracterização indireta perceber qual o aspeto associado. Um outro exemplo pode-se encontrar na frase "*O novo Iphone é muito fácil de usar.*", onde neste momento a ferramenta não extrai qualquer aspeto, no entanto no futuro pode-se extrair o termo "usabilidade" como aspeto. Também relacionado com o mesmo assunto, uma outra análise interessante seria, para além de extrair os aspetos, permitir perceber a razão para eles serem referidos. Por exemplo, na frase: "*A bateria no S5 é muito pesada*", é extraído o aspeto "bateria", no entanto não é analisado qual é o problema da bateria, que neste caso é o peso. Ou seja, para além de extrair a relação entre entidade e aspeto, é interessante extrair a relação entre aspeto-característica quando um aspeto está a ser caracterizado (no caso anterior é o peso).

Uma outra possibilidade de melhoramento das ferramentas desenvolvidas é permitir o suporte a um diferente tipo de texto. Mesmo que este trabalho se tenha focado apenas em textos provenientes de redes sociais, ou seja texto menos cuidado, esse não é o único texto relevante. Como tal, às ferramentas desenvolvidas é interessante adicionar o suporte aos dois tipo de texto, e para tal deve ser desenvolvido uma ferramenta capaz de analisar o texto e perceber se este é do tipo cuidado ou formal ou não e assim adaptar as análises posteriores consoante o tipo de texto. Por exemplo, a ferramenta de extração de polaridade foi treinada com textos extraídos da rede social *Twitter*, e por isso foi construído um conjunto de características muito direcionadas para a análise desse tipo de texto, o que pode refletir num pior resultado quando se aplicar o mesmo modelo para um tipo mais cuidado, em que as regras sintáticas são respeitadas e que não possuem *tokens* como as *hashtags*.

Por fim, uma outra tarefa que deve ser tida em conta é a melhoria de algumas ferramentas de PLN, nas quais foram detetadas pequenas falhas ao longo longo do desenvolvimento deste trabalho. Por exemplo, o identificador de frases tem mais dificuldades em repartir o texto se este possuir sinais de pontuação repetidos para terminar frases, como por exemplo no texto "*Eu adoro comer gelados!!!! Ainda por cima está calor....*". Uma outra falha detetada, e muito comum em texto das redes sociais, é a incapacidade de repartir o texto se não existir espaçamento entre as diferentes frases, como por exemplo "*Eu adoro comer gelados.Ainda por cima está calor.*". Uma outra ferramenta que apresenta falhas é o

lematizador, principalmente para a Língua Portuguesa. Este lematizador é baseado essencialmente em dicionários, e por isso se os dicionários apresentarem lacunas, o lematizador não consegue produzir o lema correto. Por exemplo, durante o desenvolvimento do projeto detetou-se que a ferramenta tinha muitas dificuldades em palavras que possuem a letra "*ç*" e em palavras que tenham sufixos verbais como "*veste-te*". Estes são só alguns exemplos de melhorias que deviam ser feitas.

Bibliografia

Afonso, S., Bick, E., Haber, R., and Santos, D. (2002). Floresta sintá(c)tica: um treebank para o português.

Aggarwal, C. C. (2011). An introduction to social network data analytics. In *Social Network Data Analytics*. Springer.

Aggarwal, C. C. and Zhai, C. (2012a). An introduction to text mining. In *Mining Text Data*. Springer US.

Aggarwal, C. C. and Zhai, C. (2012b). A survey of text classification algorithms. In *Mining Text Data*. Springer US.

Aggarwal, C. C. and Zhai, C. (2012c). A survey of text clustering algorithms. In *Mining Text Data*. Springer US.

Bell, J. (2014). *Machine Learning: Hands-On for Developers and Technical Professionals*. Wiley.

Bengio, Y. (2007). Learning deep architectures for ai. Technical report.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2013). A neural probabilistic language model. *Journal of Machine Learning Research*.

Bick, E. (2000). *The Parsing System Palavras, Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Aarhus: Aarhus University Press.

Branco, A., Silva, J., Costa, F., and Castro, S. (2011). Cintil depbank hand-book: Design options for the representation of grammatical dependencies.

Buchholz, S. and Marsi, E. (2006). Conll-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*.

Cardoso, N. (2008). Rembrandt-reconhecimento de entidades mencionadas baseado em relações e análise detalhada do texto. *PROPOR - Encontro do Segundo HAREM*.

Chinchor, N. (1997). Muc-7 named entity task definition.

Chintala, S. (2012). Sentiment analysis using neural architectures. *New York University, New York*.

Ciresan, D., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. *In Proceedings of Computer Vision and Pattern Recognition*.

ComScore and Group, T. K. (2007). Online consumer-generated reviews have significant impact on offline purchase behavior.

Crain, S. P., Zhou, K., Yang, S.-H., and Zha, H. (2012). Dimensionality reduction and topic modeling: From latent semantic indexing to latent dirichlet allocation and beyond. In *Mining Text Data*. Springer US.

Deng, L. and Yu, D. (2014). Deep learning: Methods and applications. Technical report.

Ding, X., Liu, B., and Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. *In Proceedings of the Conference on Web Search and Web Data Mining*.

dos Santos, C. N. and Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING), Dublin, Ireland*.

Ebert, S., Vu, N. T., and Schütze, H. (2015). Cis-positive: Combining convolutional neural networks and svms for sentiment analysis in twitter.

Feldman, R. (2006). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press.

Fillmore, C. J. (1982). *Frame Semantics. In Linguistics in the Morning Calm*. Hanshin Publishing Co.

Gamon, M., Aue, A., Corston-Oliver, S., and Ringger, E. (2005). Pulse: Mining customer opinions from free text. *In Proceedings of the 6th international conference on Advances in Intelligent Data Analysis*.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*.

Hoffman, T. (2008). Online reputation management is hot — but is it ethical? *Computerworld*.

Horrigan, J. A. (2008). Online shopping. *Pew Internet & American Life Project Report*.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. *In Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Hu, X. and Liu, H. (2012). Text analytics in social media. In *Mining Text Data*. Springer US.

Ivan Omar Cruz-Garcia, Alexander Gelbukh, G. S. (2014). Implicit aspect indicator extraction for aspect-based opinion mining. *submitted*.

Jackson, P. and Moulinier, I. (2007). *Natural Language Processing for Online Applications: Text retrieval, extraction and categorization*. John Benjamins Publishing, 2nd edition.

Jakob, N. and Gurevych, I. (2010). Extracting opinion targets in a single and cross-domain setting with conditional random fields. *In Proceedings of Conference on Empirical Methods in Natural Language Processing*.

Jiang, J. (2012). Information extraction from text. In *Mining Text Data*. Springer US.

Jolliffe, I. (2002). *Principal Component Analysis*. Springer.

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

Kim, E. (2013). Everything you wanted to know about the kernel trick. `http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html`. [Acedido em 6 de Julho de 2015].

Kobayashi, M. and Aono, M. (2007). Vector space models for search and cluster mining. In *Survey of Text Mining:Clustering, Classification, and Retrieval*. Springer.

Kong, L., Schneider, N., Swayamdipta, S., Bhatia, A., Dyer, C., and Smith, N. A. (2014). A dependency parser for tweets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, to appear*.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.

Li, X.-L., Zhang, L., Li, B., and Ng, S.-K. (2010). Distributional similarity vs. pu learning for entity set expansion. *In Proceedings of Annual Meeting of the Association for Computational Linguistics*.

Liu, B. (2010). Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing*. Chapman and Hall/CRC.

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

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space.

Mohammad, S. M., Kiritchenko, S., and Zhu, X. (2013). Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *In Proceedings of the Seventh International Workshop on Semantic Evaluation*.

Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., and Wilson, T. (2013). Semeval-2013 task 2: Sentiment analysis in twitter.

Nakov, P., Rosenthal, S., Stoyanov, V., and Ritter, A. (2014). Semeval-2014 task 9: Sentiment analysis in twitter.

Nilsson, J. (2014). User guide for malteval 1.0.

Pan, S. J., Ni, X., Sun, J.-T., Yang, Q., and Chen, Z. (2010). Cross-domain sentiment classication via spectral feature alignment. *In Proceedings of International Conference on World Wide Web*.

Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *In Proceedings of Meeting of the Association for Computational Linguistics*.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval.*

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: sentiment classication using machine learning techniques. *In Proceedings of Conference on Empirical Methods in Natural Language Processing.*

Popescu, A.-M. and Etzioni, O. (2005). Extracting product features and opinions from reviews. *In Proceedings of Conference on Empirical Methods in Natural Language Processing.*

Qiu, G., Liu, B., Bu, J., and Chen, C. (2009). Expanding domain sentiment lexicon through double propagation. *In Proceedings of International Joint Conference on Artificial Intelligence.*

Rocha, P. A. and Santos, D. (2000). Cetempúblico: Um corpus de grandes dimensões de linguagem jornalística portuguesa. *Actas do V Encontro para o processamento computacional da língua portuguesa escrita e falada, PROPOR.*

Sakaki, T., Okazaki, M., and Matsuo, Y. (2010). Earthquake shakes twitter users: real-time event detection by social sensors. *In Proceedings of the 19th international conference on World wide web.*

Schwaber, K. (2004). *Agile Project Management with Scrum.* Microsoft Press.

Severyn, A. and Moschitti, A. (2015). Unitn: Training deep convolutional neural network for twitter sentiment classification.

Silva, M. J., Carvalho, P., and Sarmentos, L. (2012). Building a sentiment lexicon for social judgement mining. *International Conference on Computational Processing of Portuguese (PROPOR).*

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

Su, Q., Xu, X., Guo, H., Guo, Z., Wu, X., Zhang, X., Swen, B., and Su, Z. (2008). Hidden sentiment association in chinese web opinion mining. *In Proceedings of International Conference on World Wide Web.*

Sutton, C. and McCallum, A. (2010). An introduction to conditional random fields.

Tan, A.-H. (1999). Text mining: The state of the art and the challenges. *Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases.*

Tang, D., Wei, F., Qin, B., Liu, T., and Zhou, M. (2014). Coooolll: A deep learning system for twitter sentiment classification. *In Proceedings of the 8th International Workshop on Semantic Evaluation.*

V., R. Y. and A., K. S. (2015). Aspect extraction using conditional random fields.

Weiss, S. M., Indurkhya, N., Zhang, T., and Damerau, F. (2005). *Text Mining: Predictive Methods for Analyzing Unstructured Information.* Springer.

Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining - Practical Machine Learning Tools and Techniques.* Morgan Kaufmann.

Yarowsky, D. (2010). Word sense disambiguation. In *Handbook of Natural Language Processing*. CRC Press.

Zhang, Y., Lei, T., Barzilay, R., Jaakkola, T., and Globerson, A. (2014). Steps to excellence: simple inference with refined scoring of dependency trees. In *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics*.