Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática

Reconhecimento Automático de Humor Verbal

André David Fernandes Clemêncio

Dissertação no contexto do Mestrado em Engenharia Informática, especialização em Sistemas Inteligentes orientada pelo Professor Hugo Oliveira e pela Professora Ana Alves apresentada à Faculdade de Ciências e Tecnologia, Departamento de Engenharia Informática

Setembro 2019

UNIVERSIDADE D
COIMBRA

Agradecimentos

Em primeiro lugar queria agradecer aos meus pais pelo apoio incondicional, pela compreensão e por me terem dado todos os meios para que tenha aqui chegado. O percurso académico que agora termina representa uma pequena retribuição por tudo o que fizeram e continuam a fazer por mim. Espero que vos tenha deixado orgulhosos.

Queria deixar um agradecimento especial aos meus orientadores, professor Hugo Oliveira e professora Ana Alves, pela paciência, constante disponibilidade e pelo contributo que deram a este trabalho. Muito obrigado!

Por último queria agradecer também aos meus amigos e familiares, pelo apoio e presença constantes. Sem vocês não seria a mesma coisa.

Este trabalho foi parcialmente financiado pela iniciativa INCoDe 2030 da Fundação Portuguesa para a Ciência e a Tecnologia (FCT), no âmbito do projeto de demonstração AIA, "Apoio Inteligente a empreendedores (chatbots)"; e pelo projeto SOCIALITE (PTDC/EEISCR/2072/2014), co-financiado pelo COMPETE 2020, Portugal2020 - Programa Operacional de Competitividade e Internacionalização (POCI), FEDER (Fundo Europeu de Desenvolvimento Regional) da União Europeia e FCT.

Resumo

Existem cada vez mais sistemas inteligentes desenvolvidos com o objetivo de executar várias tarefas, que são normalmente associadas ao ser humano. O processamento de linguagem humana e o uso da mesma para comunicar é uma dessas tarefas. Aqui pode incluir-se o reconhecimento de humor que será mais um passo para que exista cada vez uma melhor comunicação entre máquinas e humanos.

Este trabalho propõe implementar um modelo computacional que, através de técnicas de Processamento de Linguagem Natural e de Classificação Automática de Texto, consiga reconhecer humor na língua portuguesa, um tema para o qual não conhecemos.

Para isso, foram estudadas as abordagens seguidas por diferentes autores no reconhecimento de humor em inglês. Posteriormente foram organizados alguns conjuntos de dados com textos curtos humorísticos e não-humorísticos, que nos permitiram começar a testar e a analisar os primeiros resultados para o reconhecimento de humor em português. O objetivo será ainda identificar e extrair características da linguagem que sejam relevantes no nosso problema.

Optou-se por uma abordagem de aprendizagem computacional supervisionada, onde os modelos criados tiraram partido, não só de características lexicais, mas também de características específicas do humor. Obtiveram-se resultados que chegaram a 88% de *accuracy* e de F1 para alguns conjuntos de dados o que nos leva a querer que conseguimos chegar a um modelo devidamente diferenciador e que consegue identificar instâncias humorísticas de tipos específicos.

Palavras-Chave

Processamento de Linguagem Natural, Classificação Automática de Texto, Reconhecimento de Humor, Inteligência Artificial

Abstract

There are more and more intelligent systems developed with the purpose of performing various tasks which are usually associated with the human being. Human language processing and its usage for communication purposes is one of those tasks. Here we can include humor recognition, that would be one more step to ensure better communication between machines and humans.

The purpose of this work is to develop a computational model that, through Natural Language Processing and Text Classification techniques, can recognize humor written in the Portuguese language, for which we do not know any other specific work on this topic.

Towards this goal, approaches proposed by different authors for the task of humor recognition, in English, were first studied. Later, some datasets were built, which allowed us to start testing a baseline approach and analyzing the first results for humor recognition in Portuguese. Besides (binary) humor recognition we will also tackle the goal of identifying and extracting language features that are relevant to our main problem.

We opted for a supervised computational learning approach where the created models took advantage, not only of lexical characteristics but also of specific characteristics of humor. We obtained results that reached 88% for both accuracy and F1 for some datasets which means that we managed to build a properly differentiator model that can identify humorous instances of specific types.

Keywords

Natural Language Processing, Text Classification, Humor Recognition, Artificial Intelligence

Conteúdo

Acrónimos

IA Inteligência Artificial. 1, 5, 6, 14, 15

NLTK *Natural Language Toolkit*. 47

PLN Processamento de Linguagem Natural. 1, 3, 5–7

SVM *Support Vector Machine*. xi, 11, 20, 24, 33–37, 39, 41, 49–52, 57

TF-IDF *Term Frequency Inverse Document Frequency*. xiii, 9, 32–37, 39, 41, 51, 57

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Assistimos a uma constante evolução tecnológica onde a área da Inteligência Artificial (IA) tem assumido um papel preponderante. Existem cada vez mais sistemas inteligentes desenvolvidos para executar as mais variadas tarefas, que implicam normalmente a demonstração de capacidades típicas dos seres humanos. Uma dessas será a de processar a linguagem humana, neste contexto referida como linguagem natural, e usá-la para comunicar.

Mas o Processamento de Linguagem Natural (PLN) por parte dos agentes inteligentes é dificultado pela existência de vários fenómenos, tais como a variabilidade linguística, a ambiguidade, ou utilização de figuras de estilo como a ironia ou o sarcasmo. Por exemplo, a mesma palavra ou expressão pode ter significados diferentes em diferentes contextos, mas também há formas diferentes de transmitir a mesma intenção, usando, por exemplo, diferentes palavras. Para além de, em determinados contextos, ser possível exprimir determinada intenção dizendo precisamente o seu contrário. Ou seja, o PLN está longe de ser uma tarefa trivial.

O reconhecimento automático de humor pode ser visto como uma sub-tarefa do PLN e será mais um passo para que a comunicação entre máquinas e humanos esteja cada vez mais ao nível da comunicação entre humanos. Mais propriamente, a capacidade de reconhecer humor numa determinada linguagem é considerada um sinal de fluência nessa língua. Por exemplo, existem cada vez mais agentes conversacionais ou *chatbots* que têm como objetivo interagir com humanos através da linguagem natural. Se esses agentes tiverem a capacidade de reconhecer interações humorísticas, poderão alterar as suas ações, que podem ser desvalorizadas, se for caso disso, ou até levar a uma resposta ao mesmo nível, recorrendo a mecanismos de geração automática de humor (Gonçalo Oliveira e Rodrigues, 2018).

Entre a investigação feita no domínio do PLN, há alguns trabalhos focados no reconhecimento de humor. Contudo, a maior parte deles centra-se no inglês. Por outro lado, este trabalho foca-se na língua portuguesa, onde não foi possível identificar nenhum trabalho com o mesmo objetivo.

1.1 Objetivos

Este trabalho tem como principal objetivo desenvolver modelos para o reconhecimento automático de humor expresso em texto. O desenvolvimento dos modelos envolverá a

experimentação de diferentes algoritmos de classificação automática de texto, onde um vasto leque de caraterísticas será explorado.

Dentro das características a serem exploradas estão as características lexicais, que não são mais que as palavras, ou sequências de palavras, que constituem um texto. Além das características lexicais, serão também exploradas outras características que possam ser relevantes para o reconhecimento de humor. Com base em trabalho relacionado, serão identificadas e extraídas características específicas do humor. Entre outras, destaca-se a presença de palavras com diferentes sentidos ou a utilização de antónimos, que podem indicar a presença de humor. Para dar início a esta fase será necessário identificar ferramentas e recursos externos que possam ser usados na extração das caraterísticas definidas a partir de texto em português.

Será também avaliada a relevância que as características têm no contexto do problema com o objetivo de identificar características que estejam fortemente ligadas ao humor, sejam elas lexicais ou mais específicas.

Os modelos desenvolvidos serão aplicados depois a diferentes textos com diferentes estilos onde se espera que consigam obter uma boa performance.

1.2 Abordagem

Pretende-se então tratar o problema de reconhecimento automático de humor como um problema de classificação de texto tradicional, onde será aplicado um método de aprendizagem computacional supervisionada. Por ser um trabalho inicial, o modelo irá basear-se apenas nas características lexicais, que não são mais que as palavras que constituem o texto. Conjuntos de dados que contenham exemplos humorísticos e não humorísticos serão fornecidos de modo a treiná-lo para posteriormente se realizar uma classificação binária entre textos humorísticos e não-humorísticos. Para avaliar o desempenho do modelo, este deverá ser treinado e validado numa parte dos dados, sendo que a restante parte será usada exclusivamente para o teste.

Para a criação de coleções para treino e teste do sistema, será necessário recolher tanto textos humorísticos (exemplos positivos) como não humorísticos (exemplos negativos). Numa fase inicial pretendemos focar-nos nos textos curtos. A recolha passará então por encontrar piadas curtas em português. Já para os textos não humorísticos poderemos utilizar outros textos curtos, tais como perguntas de resposta curta, títulos de notícias ou provérbios.

Numa fase mais adiantada o objetivo será extrair dos textos, de forma automática, características específicas do humor que tenham sido usadas por outros sistemas. A comparação deste modelo com o anterior e a utilização de diferentes conjuntos de características permitirão tirar conclusões acerca das características mais importantes no reconhecimento de humor em português.

Ao longo da realização do trabalho podemos deparar-nos com alguns problemas. O primeiro está logo na recolha e organização dos conjuntos de dados a usar para o treino e para o teste do modelo. Teremos de ter especial atenção nesta fase, visto que é importante que os dados não sejam demasiado diferentes e que a sua principal diferença seja apenas a presença ou não de humor. Caso contrário, corremos o risco de estar a aprender outras distinções que não especificamente o humor.

1.3 Principais Contribuições

No final deste trabalho chegou-se a um conjunto de modelos para o reconhecimento de humor em português baseados em diferentes características. Os modelos apresentam bons resultados quando aplicados a diferentes textos com estilos diferentes. Poderão ser aplicados em diferentes contextos, tais como a seleção de publicações numa rede social ou a integração num agente conversacional.

Foi também identificado um conjunto de características relevantes associadas à presença de humor que podem ser usadas, por exemplo, noutros estudos ou no desenvolvimento de sistemas mais simples para o efeito. Além da identificação destas características foram apresentados todos os procedimentos para a sua extração.

Outra das contribuições deste trabalho foi a recolha e criação de conjuntos de dados com textos curtos humorísticos e não humorísticos, usados no desenvolvimento do sistema. Todos os conjuntos de dados serão disponibilizados para trabalhos futuros que desejem abordar o mesmo problema, podendo ser usados como referência.

Por último, foi publicado um artigo científico onde estão descritas algumas experiências feitas durante a execução do trabalho (Clemêncio, Alves e Gonçalo Oliveira, 2019).

1.4 Estrutura do Documento

Este documento encontra-se estruturado da seguinta maneira:

- No segundo capítulo, é feita uma contextualização dos principais temas que iremos abordar, nomeadamente PLN, Classificação de Texto e ainda alguns pontos em relação ao estudo do humor.
- No terceiro capítulo é analisado o estado da arte. São investigados os métodos utilizados nos trabalhos relacionados com reconhecimento automático de humor em texto, assim como as abordagens que os diferentes autores adotaram. São ainda revistos trabalhos em temas relacionados para a língua portuguesa, nomedamente na análise de sentimento ou reconhecimento de ironia.
- No quarto capítulo são descritos todos os conjuntos de dados usados e feitas as primeiras experiências, recorrendo apenas a características lexicais.
- No quinto capítulo são descritas todas as características específicas do humor que foram extraídas. São ainda feitas experiências ao modelo recorrendo somente a estas características, assim como ao conjunto de todas (específicas do humor e lexicais).
- O documento termina com uma conclusão.

Capítulo 2

Conceitos Fundamentais

Podemos definir "linguagem natural" como a linguagem que os humanos usam para comunicar. Línguas como o português ou o inglês são exemplos de linguagens naturais. Chamam-se linguagens naturais por oposição às linguagens formais, nomeadamente as linguagens de programação.

O PLN é um ramo da IA com inúmeras aplicações no âmbito das interações entre humanos e computadores onde o objetivo é facilitar a comunicação entre os mesmos, através da linguagem natural.

Se conjugarmos o PLN com a classificação de texto podemos obter várias aplicações onde o reconhecimento automático de humor se inclui. O facto de um computador conseguir entender o que um humano está a comunicar num tom humorístico pode ter vantagens pela forma como a conversa pode ser abordada (pela máquina) a partir desse momento. Ao identificar uma interação humorística, o computador pode alterar as ações a tomar, podendo desvalorizar a interação, ou até responder também num nível humorístico.

Neste capítulo vão ser abordados os conteúdos mais importantes no contexto da realização deste trabalho. Na secção 2.1 é introduzido o conceito de PLN. Na secção 2.2 falamos sobre Classificação Automática de Texto e é ainda descrito o seu processo. São também descritos alguns algoritmos de aprendizagem habitualmente aplicados a classificação de texto assim como descritas algumas métricas normalmente usadas de modo a avaliar um classificador. Na secção 2.3 é abordado o tema do Humor. São mencionadas algumas teorias do Humor assim como discutido o tópico de Humor Computacional.

2.1 Processamento de Linguagem Natural

O campo da IA tem vindo a acompanhar a constante evolução tecnológica a que vamos assistindo ao longo dos anos. Tem como objetivo desenvolver agentes inteligentes que consigam simular as capacidades humanas de modo a resolver os mais variados problemas.

A habilidade dos computadores conseguirem, eficazmente, processar e entender a linguagem humana pode ser o ponto que define a chegada de máquinas verdadeiramente inteligentes, (Jurafsky e Martin, 2009). Podemos assumir esta afirmação na medida em que o uso eficiente da uma língua para comunicar (no caso, a linguagem humana) está diretamente ligado às capacidades cognitivas dos seres humanos.

Turing (1950) introduziu aquele que ficou conhecido como o teste de Turing. Neste

teste o PLN seria essencial, na medida em que o objetivo era que um agente conseguisse interpretar perguntas feitas em linguagem natural para posteriormente conseguir dar uma resposta adequada. Para Turing, o uso correto da linguagem humana por parte de um agente computacional constituía uma capacidade essencial para determinar a sua inteligência.

A área de PLN está então ligada à IA e tem como principal objetivo desenvolver modelos computacionais para processar a linguagem humana e utilizá-la para comunicar. Tendo esta capacidade, as máquinas conseguem executar uma quantidade variada de tarefas. Incluem-se aqui aplicações de nível mais baixo, como a identificação da função gramatical das palavras, a identificação das palavras mais relevantes para a compreender um texto e a sua associação a uma entrada num dicionário ou ontologia.

De modo a que uma aplicação consiga executar essas mesmas tarefas é necessário que exista um conhecimento da linguagem, aliás, é isso que distingue as aplicações de PLN de outras aplicações. Ainda que nem todas as aplicações necessitem de aceder a todos, podemos representar conhecimento sobre a língua em vários níveis, nomeadamente:

- **Fonética e fonologia**: relacionado com os sons produzidos pelas palavras de uma determinada linguagem.
- **Morfologia**: conhecimento acerca estrutura, da formação e da classificação de cada palavra.
- **Sintaxe**: conhecimento acerca da estrutura das frases.
- **Semântica**: conhecimento do significado das palavras
- **Pragmática**: relação entre o significado das palavras e o real objetivo das interações.
- **Discurso**: análise para lá da frase, que implica ter acesso a um contexto maior.

Muitas das tarefas em processamento de linguagem natural podem ser vistas como resolver ambiguidades num dos seus níveis (Jurafsky e Martin, 2009). Identificar o significado com que uma palavra está a ser usada pode ser bastante complicado. Consideremos os seguintes exemplos:

1. *O que é mais difícil do que colocar um elefante no banco de trás de um carro? Colocar dois elefantes no banco de trás de um carro!*
2. *Quem é Nodar Dschawachischwili? Chefe do Banco Central da Geórgia.*

Nas frases em questão a palavra "banco" assume dois significados diferentes. Se na primeira se refere ao assento, já na segunda frase refere-se à instituição bancária. Neste caso seria necessário proceder a uma desambiguação do sentido da palavra de modo a perceber o significado com que a palavra estava a ser usada em cada uma das frases. O objetivo dos modelos e algoritmos desenvolvidos para PLN é muitas vezes resolver este tipo de ambiguidades.

Dentro dos modelos mais utilizados em PLN estão os modelos probabilísticos, que são uma forma de lidar com a ambiguidade. Mais propriamente, quando ela existe, escolhe-se a opção mais provável.

A existência de ambiguidade pode ser considerada a principal diferença entre linguagem natural e as linguagens formais. Fica então claramente mais difícil processar linguagem natural do que outro tipo de linguagens, como são exemplo as linguagens de programação.

As linguagens de programação são constituídas por um conjunto de regras sintáticas e semânticas usadas para definir um determinado programa de computador. Esta definição contrasta fortemente com a definição de linguagem natural. A forma como nós comunicamos uns com os outros, seja através da escrita ou do diálogo é o que definimos por linguagem natural. Existem então várias componentes que fazem com que a linguagem natural seja bastante diferente de linguagens formais.

Podemos começar por mencionar o facto de existir um número infinito de palavras e referir também que a linguagem natural está em constante evolução e que é constituída por imensos fenómenos que a tornam ainda mais complexa. Para além da ambiguidade, outros fenómenos que tornam o PLN desafiante incluem a vagueza, a variabilidade linguística (o mesmo conceito ou intenção pode ser expresso de formas consideravelmente diferentes), a anáfora, ou a elipse, para além de figuras de estilo como a ironia, a metáfora, ou a antítese. Na Tabela 2.1 estão descritos e exemplificados estes mesmos fenómenos.

	Descrição	**Exemplo**
Anáfora	Expressão que se refere a uma outra expressão presente na mesma frase ou texto.	Como é que a **Amelia** conseguiu escapar? **Ela** fugiu para o deserto.
Elipse	Supressão de parte de uma frase facilmente subentendida pelo contexto da frase.	Para onde vais amanhã? *[Vou]* A Coimbra.
Ironia	Uso de palavra (expressão) no sentido oposto ao que se deveria usar.	Ele corre tão **rápido** como uma tartaruga!
Metáfora	Transporta uma palavra (ou expressão) do seu sentido literal para um sentido figurado.	Este problema é só **a ponta do iceberg**.
Antítese	Utilização de duas palavras com significados opostos na mesma frase.	Faça **sol** ou faça **chuva**, estarei lá amanhã.

Tabela 2.1: Descrição e exemplos de alguns fenómenos linguísticos e figuras de estilo

Tudo isto difere muito comparando com linguagens de programação onde existem regras definidas previamente e que não mudam. Por tudo isto, o desafio de processar linguagem natural acaba por ser bastante complexo comparando com outro tipo de linguagens.

Existem inúmeras aplicações de alto nível quando se fala em PLN, como a interpretação de textos ou diálogos, tradução desses textos para outra língua, ou resposta automática a perguntas. Aplicações que têm tido grande atenção nos últimos anos são os agentes conversacionais (*chatbots*). O objetivo destes agentes será processar e compreender a linguagem natural, fazendo com que sejam capazes de dialogar com os seus utilizadores.

O PLN representa então uma capacidade bastante importante quando falamos em agentes inteligentes. O facto da linguagem e do pensamento estarem fortemente ligados faz com que a tecnologia que envolva PLN seja extremamente importante no desenvolvimento e na evolução de futuras tecnologias.

2.2 Classificação Automática de Texto

Uma tarefa relacionada com o PLN é a classificação automática de texto. O seu principal objetivo é a atribuição de uma categoria, dentro de um conjunto de categorias pré-definidas, a textos, curtos ou longos, escritos em linguagem natural.

Na maioria dos casos em que falamos de classicação de texto, a abordagem mais usada

será a aprendizagem computacional supervisionada (Jurafsky e Martin, 2009). Nesse tipo de abordagem o classificador é treinado tendo por base o mapeamento de um conjunto inicial de documentos às suas categorias, de acordo com o desejado (conjunto de treino). Posteriormente, o objetivo será classificar automaticamente os novos documentos de acordo com as categorias conhecidas, recorrendo para isso ao conhecimento adquirido previamente.

O facto de, normalmente, não ser simples definir o que um texto tem de conter para ser associado a uma determinada categoria faz com que esta abordagem seja a mais comum. Tudo isto difere de uma abordagem não supervisionada em que não existe nenhuma informação externa fornecida ao classificador. No chamado *clustering*, documentos são agrupados entre si, de acordo com a sua semelhança, mas não há uma noção de categoria bem definida.

Na figura 2.1 está representado o processo de classificação de texto de acordo com Song et al. (2014).

2.2.1 Dados de Entrada

Na fase inicial do processo de classificação de texto é fornecido um conjunto de dados que irão servir para treinar o classificador. Os dados, neste caso, serão documentos textuais. Todos os dados fornecidos são antes categorizados manualmente, o que pode ser feito manualmente, por alguém que ofereça garantias de qualidade neste processo, por exemplo, por ser especialista no domínio ou estar suficientemente dentro dos objetivos. É comum não usar todos os dados categorizados na fase de treino e reservar um sub-conjunto para aferir a qualidade dos resultados obtidos com o classificador treinado (dados de teste).

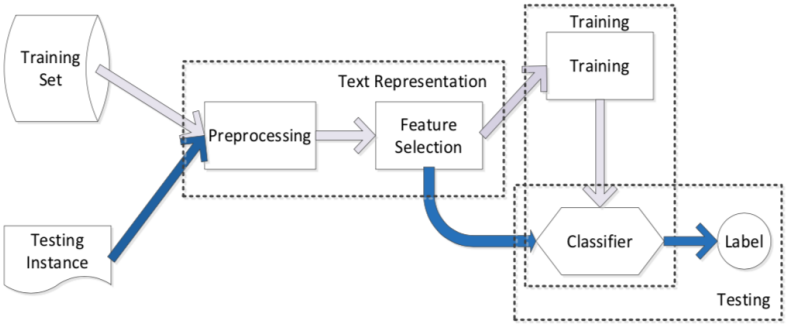

Figura 2.1: Processo de classificação de texto (Song et al., 2014)

2.2.2 Representação dos Dados

De forma a facilitar a manipulação dos documentos, é necessário representá-los de uma forma mais amigável, mas que reflita o seu conteúdo através de um conjunto de características (em inglês, *features*), a explorar no processo de classificação. A extração dessas caraterísticas é normalmente antecedida por uma fase de pré-processamento. Dentro do pré-processamento podem ocorrer várias etapas, tais como:

- **Tokenização**: divisão do texto em unidades fundamentais / átomos (em inglês, *tokens*) que, neste caso, serão as palavras ou, possivelmente, expressões multi-palavra (e.g. "Nova Iorque", "tese de mestrado").

- **PoS Tagging**: processo de atribuir a cada palavra a respectiva função gramatical em contexto (em inglês, *part-of-speech*, *PoS*). Por exemplo, é aqui que se identificam os substantivos, verbos, adjetivos ou advérbios, determinantes, preposições, entre outros, que podem ter diferente utilidade no processo de classificação.

- **Remoção de *"Stopwords"***: remoção de palavras que, por serem demasiado frequentes, podem ser consideradas irrelevantes para o problema de classificação. Apesar do conjunto de stopwords poder variar com o objetivo ou domínio do problema, é comum incluir preposições e determinantes, classes fechadas de palavras, menos importantes para interpretar o texto do que, por exemplo, substantivos, verbos, adjetivos e advérbios.

- **Normalização**: processo de reduzir palavras a uma forma mais simples, que permita associar palavras com a mesma raíz ao mesmo conceito e assim reduzir o número de características. Isto pode ser feito através do processo de *stemming* (remoção das terminações) ou lematização (desflexionização). O primeiro requer pouco ou nenhum conhecimento linguístico e o resultado podem não ser exatamente palavras (o radical de "gostava" será "gost"). Já o segundo depende de conhecimento linguístico, tal como a função gramatical das palavras e mesmo o acesso a um léxico morfológico, mas deve resultar em palavras que existem na língua e que aparecem num dicionário (o lema de "gostava" será "gostar").

Dependendo do objetivo e dos recursos disponíveis para a língua em questão, o pré-processamento pode ser mais complexo, e extrair outras características. Pode interessar ir para além da palavra e considerar sequências de palavras, como os n-gramas (Houvardas e Stamatatos, 2006), em análise de sentimentos é normal contar o número de palavras com polaridade tipicamente positiva e negativa (Agarwal et al., 2011) e em reconhecimento de humor são também extraídas características como a aliteração ou a prensença de palavras antónimas nas frases (Mihalcea e Strapparava, 2005). No entanto, em muitos casos o pré-processamento é ainda mais simples e pode basear-se apenas na tokenização e, eventualmente, aplicar remoção de stopwords ou o *stemming*. Isso será suficiente para representar documentos através do conjunto das suas tokens/stems, independentemente da sua ordem, normalmente chamado de modelo *bag-of-words*. Quando outras características são consideradas, é comum concatenar os seus valores num vetor que terá sempre a mesma estrutura para cada documento.

Seguidamente é feita a extração das características que, neste caso, são as palavras que constituem o texto. Para o peso de caraterísticas é comum usar-se um algoritmo como o *TF-IDF*, que basicamente, o determina a frequência relativa das palavras num documento específico, comparando depois com a proporção inversa dessas mesmas palavras mas em relação ao conjunto de todos os documentos (Salton e Buckley, 1988). Este cálculo permite-nos perceber o quão relevante é uma determinada palavra num documento. Assumindo que $tf_{i,j}$ será o número de ocorrências de um termo i num documento j, N o número total de documentos e df_i o número de documentos que contêm o termo i, o cálculo para o valor de TF-IDF ($W_{i,j}$) é feito da seguinte forma:

$$W_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \tag{2.1}$$

Para além do *TF-IDF* pode também ser usada uma matriz que associa termos e documentos (*term-document matrix*) onde cada linha representa uma palavra do vocabulário e cada coluna representa um determinado documento. Numa matriz deste tipo, cada

posição corresponde ao número de vezes que uma palavra (definida pela linha) aparece num documento (definido pela coluna) (Jurafsky e Martin, 2009). Transpondo isto para um exemplo real, iríamos ter uma matriz constituída por N vetores, onde N é o número total de documentos considerado, em que cada vetor assumia uma dimensão V, onde V corresponde ao tamanho do vocabulário.

Dentro das características usadas para a representação de documentos podem estar ainda algumas relações entre as palavras consituintes de um documento, por exemplo o valor de similaridade que um par de palavras tem. De modo a captar essas relações é necessário que cada palavra tenha uma representação vetorial. Por exemplo, o cálculo da similaridade pode ser feito através do cálculo da distância entre dois vetores, correspondentes a duas palavras. Essas representações vetoriais têm então o nome de *word embeddings*. O método usado para construir *word embeddings* será o *Word2Vec*. O *Word2Vec* recebe como *input* um conjunto de dados e produz um conjunto de vetores, que podem ter várias dimensões, onde cada palavra do conjunto de dados corresponde a um vetor desse conjunto.

2.2.3 Seleção de Características

Muitas vezes, em problemas de classificação de texto, existe uma grande quantidade de características representantes dos dados. No entanto, muitas destas características acabam por ser irrelevantes para o objetivo final. Além disso, o uso de uma grande quantidade de características pode levar a que haja um sobre-ajuste (em inglês, *overfitting*) do modelo, fazendo com que este, apesar de apresentar bons resultados em dados de treino, não consiga generalizar e tenha maus resultados na classificação de dados que nunca tenha visto.

Será necessário então recorrer a métodos que selecionem as características mais relevantes para o problema em questão. Esta seleção de características fará com que se reduza o tempo de execução de um algoritmo de aprendizagem e ainda que se consiga chegar a um conjunto de características mais geral de modo a combater o *overfitting* (Dash e Liu, 1997) .

Um bom método de seleção de características terá como principal objetivo escolher as características que ofereçam mais informação acerca das classes do problema e eliminar as que sejam mais redundantes. Dentro dos métodos mais efetivos para o efeito estão o *Information Gain* e ainda o teste do qui-quadrado (χ^2) (Houvardas e Stamatatos, 2006).

- ***Information Gain***: calcula o valor de informação obtido para a previsão de uma classe, pela presença ou ausência de um termo num documento (Yang e Pedersen, 1997). Dado um conjunto de dados, é calculado o valor de *Information Gain* para cada característica. Depois disso são removidas todas as características em que o valor calculado está abaixo de um limite previamente definido.

- $\boldsymbol{\chi^2}$: o teste estatístico do χ^2 calcula a dependência que um termo t tem em relação a uma classe c. Assumindo que A representa o número de vezes que t e c ocorrem ao mesmo tempo, B representa o número de vezes que t ocorre sem c, C representa o número de vezes que c ocorre sem t, D representa o número de vezes que nem t nem c ocorrem e que N é o número total de documentos. O valor do teste pode ser calculado da seguinte forma:

$$\chi^2(t, c) = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)} \tag{2.2}$$

O valor do teste será 0 quando t for completamente independente de c. Quanto maior for o valor obtido no teste, mais dependência existirá entre o termo e a classe.

2.2.4 Treino

De modo a treinar o sistema será necessário definir que instâncias serão usadas tanto para treinar, como para testar o sistema. À falta de dados suficientes ou de um bom critério para fazer a divisão anterior, pode optar-se pela técnica de *k-fold cross-validation*. Aí o conjunto dos dados é aleatoriomente dividido em k subconjuntos iguais. Valores comuns de k são 5 ou 10. Desses k subconjuntos, um deles funcionará como o conjunto de teste. Este processo será repetido k vezes, com cada subconjunto a funcionar uma vez como conjunto para o teste.

Por fim é preciso escolher o algoritmo com que o classificador será treinado. Entre os algoritmos mais utilizados em tarefas de classificação (Aggarwal e Zhai, 2012) de texto estão:

- **Naive Bayes**: algoritmo que faz uso do Teorema de Bayes, que assume que todas as características são indepedentes. Apesar de ser uma suposição que pode não fazer muito sentido em problemas reais, a verdade é que este algoritmo costuma ter boas performances em problemas de classificação (McCallum e Nigam, 1998). Pelo facto de assumir essa independência, o algoritmo consegue aprender os pârametros de cada característica separadamente o que vai simplificar bastante o processo de aprendizagem, especialmente quando existe um grande número de características (o que normalmente acontece em problemas de classificação de texto).

  Para um documento d, de todas as classes $c \epsilon C$, retorna a classe mais provável a que o documento pertence. Usando o Teorema de Bayes, a probabilidade pode ser formulada como:

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} \tag{2.3}$$

- ***SVM***: baseado no Pincípio da Minimização do Risco Estrutural em que a ideia será encontrar uma hipótese h para a qual consigamos garantir o menor erro (Joachims, 1998). Basicamente, recebendo dados já categorizados (aprendizagem computacional supervisionada), o algoritmo retorna o hiperplano ótimo, de modo a separar os exemplos positivos dos exemplos negatovos, com a maior margem possível. As margens não são mais do que as distâncias entre a linha do hiperplano e o ponto mais próximo dessa linha, de cada classe. Na figura 2.2 podemos visualizar um exemplo de como funciona o algoritmo:

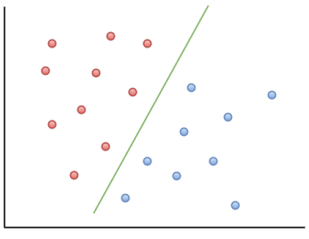

Figura 2.2: Visualização do melhor hiperplano retornado pelo *SVM*

- ***Decision Trees***: tem como objetivo prever o valor de uma variável (classe a que um documento pertence) através da aprendizagem de regras de decisão simples que são deduzidas através das características dos dados. As características podem ser representadas como os nós internos da árvore sendo que cada ramo representa o resultado do teste feito a essa característica. Por fim, os nós terminais representam a classe.

  A figura 2.3 representa uma pequena *Decision Tree* onde o objetivo é saber se uma pessoa é do sexo masculino ou feminino baseada apenas em duas características, a altura e o peso da pessoa.

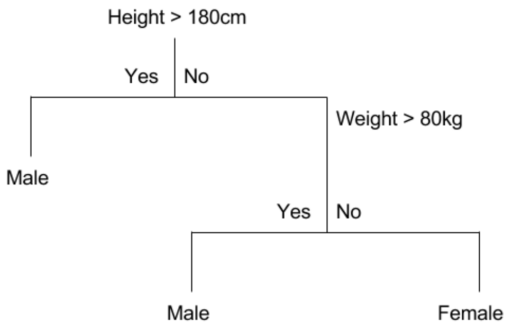

Figura 2.3: Exemplo de *Decision Tree*

- ***Random Forest***: este algoritmo não é mais que um grande número de *Decision Trees* que operam como um conjunto (*ensemble*). Individualmente cada *Decision Tree* prevê uma determinada classe, sendo no fim escolhida a classe com mais votos. Basicamente está a ser aplicado o conceito da sabedoria de multidões. Interessa também que as *Decision Trees* usadas não estejam correlacionadas, com o objtetivo de não evoluírem todas para a mesma direção (prever a mesma classe), que pode estar errada.

2.2.5 Teste

Na fase de testes o objetivo será aferir o desempenho do classificador. Para o efeito podemos recorrer ao cálculo de algumas métricas que nos podem indicar se obtivemos, ou não, uma boa classificação do texto. De modo a perceber como é feito o cálculo dessas métricas assumimos que:

True Positive: número de casos classificados corretamente como positivos

False Positive: número de casos classificados incorretamente como positivos

True Negative: número de casos classificados corretamente como negativos

False Negative: número de casos classificados incorretamente como negativos

Dentro das métricas mais relevantes podemos mencionar:

- ***Accuracy***: percentagem de instâncias de teste que foram corretamente classificadas

no total

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \tag{2.4}$$

- ***Precision***: percentagem de instâncias de teste corretamente classificadas dentro das que foram classificadas como pertencendo a uma determinada classe

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{2.5}$$

- ***Recall***: percentagem de instâncias de teste corretamente classificadas dentro de todas as que deveriam ter sido classificadas como pertencendo a uma determinada classe

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{2.6}$$

- ***F1 score***: considera os valores de *Precision* e de *Recall* e calcula a sua média harmónica. O seu valor máximo será 1 caso os valores de *Precision* e de *Recall* tenham ambos este valor, sendo 0 o pior valor.

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.7}$$

2.2.6 Aplicações

Existem inúmeras aplicações onde se pode dar uso à classificação de texto. Podemos começar por mencionar a análise de sentimento em textos (Pang e Lee, 2004). Definir se o autor expressa um sentimento positivo ou negativo por determinado objeto pode ser considerada uma tarefa de classificação de texto. A sua forma mais simples será uma classificação binária, onde para cada palavra se atribui um sentimento positivo ou negativo.

A deteção de *spam* (Jindal e Liu, 2007) ou mesmo o reconhecimento de humor são outras das aplicações importantes. Acabam por ser também classificações binárias onde o que se pretende é definir se um e-mail, por exemplo, contém ou não *spam* ou se uma frase é ou não humorística. Dentro das tarefas relacionadas podemos ainda referir a determinação de um autor de um texto e ainda as suas características como a idade ou o género.

2.2.7 Classificação de Textos Curtos

Títulos de notícias, mensagens de texto, comentários na web, piadas curtas ou mesmo posts em redes sociais fazem parte de um tipo de texto que tem como característica o facto de não serem compostos por mais que uma ou duas frases.

Se considerarmos a classificação destes textos vamos chegar à conclusão que os métodos utilizados para classificar documentos longos podem não resultar perfeitamente. Nos textos curtos nem sempre existe informação em quantidade suficiente para que os métodos de aprendizagem tradicional possam ser aplicados diretamente (por exemplo, número de ocorrências de uma palavra). Esses textos tendem ainda a ser extremamente ambíguos e sem informação contextual importante.

As características muito próprias dos textos curtos que são caracterizados por serem constituídos por poucas palavras e pela muita dispersão de termos faz com que, classificá-los, seja uma tarefa muito complicada. A classificação de textos curtos pode ser baseada numa análise semântica do texto (Song et al., 2014) de onde se podem retirar características semânticas que ajudam no processo de classificação.

2.3 Estudo do Humor

A principal reação que as pessoas têm ao humor é o riso que, muitas vezes, é até uma resposta impulsiva do nosso corpo. Partindo desse princípio conseguimos facilmente afirmar que todos os textos humorísticos têm como principal objetivo, fazer rir. Existem diversas teorias que propõem explicar o humor, tentando perceber o que é que, exatamente, nos provoca uma reação de riso. O Humor Computacional é então um ramo da IA que utiliza os computadores no estudo do humor em que o objetivo poderá ser gerar ou reconhecer humor.

2.3.1 Teorias do Humor

De modo a definir o que consideramos, ou não, humorístico, muitas vezes mencionamos o facto de, numa frase, existirem ideias completamente opostas. A incongruência é então uma das teorias que tenta explicar o fenómeno do humor colocando a apresentação de ideias distintas numa frase como um dos principais fatores para determinar se essa frase é humorística.

Tagnin (2005) baseia-se na teoria da incongruência para explicar o humor e a autora afirma que: "Se entendermos aquilo que é esperado como o convencional na linguagem, ou seja, aquilo que foi consagrado pelo uso, podemos afirmar que o humor pode ser obtido através da quebra de convencionalidade". Considerando que para dominar a convencionalidade de uma certa língua é necessário que sejamos fluentes nessa mesma língua, uma forma de criar e entender humor, será quebrar essa mesma convencionalidade.

Para que o humor seja compreendido é então necessário que o leitor conheça as expressões convencionais de uma linguagem natural, de modo a que repare na manipulação de alguns fenómenos linguísticos que estão descritos e exemplificados na tabela 2.2.

	Descrição	**Exemplo**
Homonímia	Duas palavras diferentes com a mesma grafia e com o mesmo som.	"verão" (substantivo) e "verão" (verbo)
Homofonia	Palavras com o mesmo som mas grafias diferentes.	"cela" (substantivo) e "sela" (verbo)
Polissemia	Uma mesma palavra com diferentes sentidos.	A palavra "letra" pode significar o elemento básico do alfabeto, texto de uma canção ou a caligrafia de uma pessoa.
Paronímia	Palavras com som e grafia semelhantes mas com significados diferentes.	"cumprimento" e "comprimento"

Tabela 2.2: Descrição e exemplos de algumas relações entre palavras

Outra das teorias que propõem explicar o humor baseia-se na hostilidade. O sentimento de superioridade sobre alguém ou mesmo a agressividade tendo algo como alvo é tido muitas vezes como causa para uma situação humorística (Raskin, 2008). Uma situação de infortúnio relacionada com outras pessoas pode ser motivo de riso, visto que faz com que nos sintamos superiores em relação a elas.

O humor pode ainda funcionar como um mecanismo de libertação de alguma tensão psicológica ou mesmo física. Por vezes uma forma de conseguirmos ultrapassar algumas inibições sociais será com a utilização de humor nessas situações.

2.3.2 Humor Computacional

O objetivo da IA é desenvolver máquinas inteligentes que tenham capacidades cognitivas semelhantes ao humanos. Aqui inclui-se a capacidade de comunicar usando a linguagem humana. O humor é muitas vezes associado à quebra da convencionalidade de uma linguagem (Tagnin, 2005). Tanto a capacidade de reconhecer como de gerar humor é sinal de fluência numa linguagem, seja num humano ou numa máquina. Daqui podemos concluir que essas mesmas capacidades estão totalmente associdas à inteligência.

O humor computacional (Binsted et al., 2006) é um campo da IA que usa os computadores tanto para gerar, como para reconhecer humor. O humor é visto como uma característica que pode, no fundo, fazer um agente computacional parecer mais humano (Raskin, 2008). O principal objetivo no campo do humor computacional será adicionar as capacidades humorísticas (que são inerentes aos humanos) aos computadores, tornando-os mais inteligentes.

O humor que normalmente encontramos expresso em texto pode ter diferentes estruturas sintáticas. Dentro das mais comuns, e que são normalmente utilizadas em estudos relativos ao humor computacional, estão as *one-liners* ou ainda pequenos diálogos. *One-Liners* são piadas curtas normalmente constituídas por uma frase (que pode ser pergunta/resposta):

> *O que é um byfe? São oito bifes.*
>
> *Porque é que os fotões não fazem pizza? Porque não têm massa.*

Já os diálagos são constituídos por mais que uma frase e podem, ou não, conter uma introdução:

> *-Sabes onde é que um elefante se esconde bem? Atrás de um morango.*
> *-Já viste algum elefante atrás de um morango?*
> *-Não! Estás a ver como ele se esconde bem...*

Conseguimos concluir que a utilização de humor facilita a interação social e permite melhorar a comunicação entre humanos. Tendo como grande objetivo fazer com que os computadores comuniquem de uma forma natural com os humanos, então o uso do humor, tanto na geração como no reconhecimento, será uma parte crucial para isso acontecer. Além disso, fazer com que os computadores consigam gerar e reconhecer humor pode dar-nos informações de como o cérebro humano trata, não só o humor, mas também a linguagem e o conhecimento em geral (Binsted et al., 2006).

Capítulo 3

Trabalho Relacionado

Existem alguns trabalhos relacionados com o reconhecimento de humor em texto, na sua maioria em inglês. Já para a língua portuguesa não é conhecido nenhum trabalho nesta área. Neste capítulo iremos analisar os trabalhos desenvolvidos no âmbito do reconhecimento de humor.

Vamos ainda estudar alguns trabalhos relacionados com o tema do reconhecimento de humor, como a deteção de ironia ou a análise de sentimento, sendo que aqui já existe algum trabalho desenvolvido para o português.

3.1 Reconhecimento Automático de Humor

No que diz respeito a trabalhos relacionados com o reconhecimento de humor existem alguns que vão para além da classificação binária. Em 2017 a tarefa número 6 do SemEval[1] propunha classificar "*tweets*" humorísticos e dividia-se em 2 sub-tarefas. A primeira seria classificar qual, de dois "*tweets*" dados, seria o mais engraçado. Na segunda tarefa o objetivo era atribuir uma classificação a um conjunto de "*tweets*", do mais engraçado para o menos engraçado.

No entanto, a maior parte do estudo feito neste campo aborda o problema como um problema de classificação binária. Nesta secção vamos analisar as características, abordagens e *datasets* que foram utilizados pelos vários autores nos trabalhos relacionados com o reconhecimento de humor.

3.1.1 Características

Mihalcea e Strapparava (2005, 2006) definiram três características estilísticas que seriam relevantes para detetar humor.

- **Aliteração**: a aliteração é uma figura de estilo que está relacionada com os aspetos fonéticos das palavras. Consiste em repetir sons semelhantes em palavras de uma frase. Estudos no campo do humor (Ruch, 2002) indicam que as propriedades fonéticas das piadas são tão importantes como o seu conteúdo. Muitas piadas curtas baseiam-se neste aspeto com o objetivo de captar a atenção do leitor, como podemos ver nos seguintes exemplos:

[1] `http://alt.qcri.org/semeval2017/task6/` último acesso em Agosto de 2019

Veni, Vidi, Visa: I came, I saw, I did a little shopping.

Infants dont't enjoy infancy like adults do adultery.

De modo a extrair esta característica os autores identificaram o número de cadeias de aliteração existentes em cada exemplo do conjunto de dados que utilizaram. As cadeias foram extraídas com o auxílio do *CMU Pronunciation Dictionary*[2].

- **Antonímia**: a antonímia representa a relação entre palavras com um significado oposto. Visto que o humor é muitas vezes baseado em algum tipo de incongruência (ideias contraditórias), é natural que muitas *one-liners* contenham palavras com sentidos opostos. Os seguintes exemplos mostram piadas com esta característica:

  A clean desk is a sign of a cluttered desk drawer.

  Always try to be modest and be proud of it!

  A extração desta característica foi feita recorrendo à base de conhecimento lexical *WordNet* (Miller, 1995), onde se iria buscar a relação de antonímia entre nomes, verbos, adjetivos e advérbios.

- **Vocabulário Calão**: o vocabulário calão é constituído por palavras com um significado grosseiro ou rude. O humor baseado em vocabulário calão é muito popular, logo seria uma *feature* interessante a explorar. Mihalcea e Strapparava (2005) definiram que frases com alguma orientação sexual poderiam ser indicadoras da presença de humor, apresentando depois dois exemplos:

  The sex was so good that even the neighbors had a cigarette.

  Artificial Insemination: procreation without recreation.

  A identificação de palavras com esta conotação foi feita recorrendo a uma extensão da *WordNet*, *WordNet Domains*[3] extraindo todas as palavras categorizadas com o domínio "*Sexuality*".

Para além das características já mencionadas Sjöbergh e Araki (2007) consideraram ainda a existência de ambiguidade nas frases, o uso de expressões idiomáticas ou mesmo a presença de outros indicadores a que chamaram *Joke Words*.

- **Ambiguidade**: a ambiguidade é um fenómeno indissociável do humor. De modo a medir a ambiguidade de uma determinada palavra numa frase foi contado o número de sentidos que lhe poderiam ser atribuídos, com o auxílio de um dicionário online[4].

- **Expressões Idiomáticas**: as expressões idiomáticas são conjuntos de duas ou mais palavras que se caracterizam por não se conseguir definir o seu significado através do seu sentido literal. As *one-liners* são muitas vezes reformulações de alguns provérbios populares ou expressões idiomáticas. Para extrair esta característica foram retirados da web 3000 provérbios da língua inglesa. Posteriormente fez-se a comparação de cada um desses provérbios com cada frase no conjunto de dados, verificando se existia uma grande quantidade de sobreposição de termos.

[2] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict` último acesso em Agosto de 2019

[3] `http://wndomains.fbk.eu/` último acesso em Agosto de 2019

[4] `https://www.dictionary.com/` último acesso em Agosto de 2019

- ***Joke Words***: palavras que, quando usadas, são fortes indicadores de que a frase pode ser considerada humorística. Por exemplo, animais são muitas vezes mencionados em piadas (foi dado o exemplo da palavra "*ducks*"). De modo a identificar estas palavras foi construída uma lista, recorrendo ao conjunto de dados de treino, que continha palavras que ocorriam pelo menos 5 vezes em piadas e que fossem 5 vezes mais comuns em piadas do que em frases sem piada. Podem incluir-se ainda passagens curtas que são comuns em muitas piadas (*"change a light bulb"*). Para capturar estes conjuntos foram construídas listas semelhantes às anteriores mas com bi-gramas ou tri-gramas.

Mihalcea e Pulman (2007) identificaram a orientação negativa das frases e o uso de vocabulário centrado na pessoa como sendo as características que melhor caracterizavam os exemplos humorísticos do seu conjunto de dados. Muitas das piadas que foram analisadas continham palavras com uma conotação negativa ("*bad*", "*illegal*", "*wrong*", "*error*", "*failure*", "*mistake*"), como se pode observar nos seguintes exemplos dados pelos autores:

> *"When everything comes your way, you are in the* ***wrong*** *lane"*
>
> *"User* ***error****: replace user and press any key to continue"*

Recorrer a palavras ligadas à pessoa, por exemplo, "*I*", "*me*" ou "*you*" também foi identificado por Mihalcea e Pulman (2007) como sendo um forte indicador da presença de humor, assim como o facto de serem mencionados grupos sociais ("*lawyer*" ou "*programmer*") ou relações pessoais ("*wife*", "*husband*" ou "*son*"), como nos seguintes exemplos:

> *"Of all the things* ***I*** *lost,* ***I*** *miss my mind the most"*
>
> *"****You*** *can always find what* ***you*** *are not looking for"*
>
> *"It was so cold last winter that I saw a* ***lawyer*** *with his hands in his own pockets."*

Reyes, Rosso e Buscaldi (2009) basearam-se nas características utilizadas por Mihalcea e Pulman (2007) e ainda tiveram especial atenção a outros aspectos, tais como as "*Wh-Phrases*", que são frases que contêm pronomes interrogativos ("*What*", "*Who*", etc):

> *"****What*** *are the 3 words you never want to hear while making love? Honey, I'm home!"*

Yang et al. (2015) utilizaram características já mencionadas como a deteção de incongruência nas frases, a identificação de ambiguidades, ocorrência de palavras com polaridade negativa ou ainda o estilo fonético (uso de rima ou aliteração).

Barbieri e Saggion (2014) tinham como objetivo detetar não só humor mas também ironia no Twitter. Para isso construíram alguns grupos de características. Exploraram a frequência das palavras, comparando palavras usadas habitualmente na língua inglesa com palavras usadas raramente, com o objetivo de perceber se o uso de ambas numa frase criava algum tipo de incongruência (que seria sinal de humor). Estudaram também a estrutura dos "*tweets*" recolhidos analisando o número de palvras ou sinais de pontuação. Analisaram por fim a intensidade dos adjetivos e advérbios presentes nas frases, a ambiguidade existente ou a polaridade (positiva ou negativa) dos "*tweets*".

Na tabela 3.1 podemos ver as principais características usadas nos trabalhos que foram analisados:

	Aliteração	Rima	Calão	Ambiguidade	Polaridade	Antonímia
Mihalcea e Strapparava (2005)	X	X	X			X
Sjöbergh e Araki (2007)	X		X	X		X
Mihalcea e Pulman (2007)	X		X		X	X
Reyes, Rosso e Buscaldi (2009)			X	X	X	
Yang et al. (2015)	X	X		X	X	
Reyes, Rosso e Buscaldi (2012)				X	X	

Tabela 3.1: Características usadas pelos diferentes autores

3.1.2 Abordagens

Um dos pontos em comum em quase todos os estudos relacionados com o tema é que tratam o problema como um problema de classificação binária onde o modelo computacional tenta reconhecer se existe, ou não, humor tendo um conta certas características da linguagem.

Mihalcea e Strapparava (2005) realizam três experiências diferentes de modo a estudar o reconhecimento de humor:

- Na primeira experiência o classificador tem por base as características que foram descritas anteriormente (aliteração, antonímia e calão) que são características numéricas e funcionam como heurísticas. Aqui definiu-se um valor mínimo admitido para que uma frase fosse considerada humorística. Estes valores foram aprendidos automaticamente usando uma árvore de decisão (em inglês, *Decision Tree*) aplicada num pequeno conjunto de exemplos humorísticos e não humorísticos.
- Trataram ainda o reconhecimento de humor como sendo uma tarefa de classificação de texto tradicional. Neste caso o classificador é treinado apenas com o conteúdo textual que lhe é fornecido, sem características adicionais. Aqui compararam-se resultados quando utilizados diferentes algoritmos usados para classificação de texto (Naive Bayes e *SVM*).
- Por último, foi feita uma experiência onde se explorava a combinação de ambas as características para o reconhecimento de humor, que seriam tanto as características baseadas apenas no texto da frase como as características específicas.

Mihalcea e Strapparava (2005) obtiveram os melhores resultados quando realizaram a terceira experiência onde juntaram as características estilísticas com as características apenas baseadas no conteúdo do texto. Os resultados obtidos obtiveram sempre uma *accuracy* acima dos 80%.

Sjöbergh e Araki (2007) optaram por testar o classificador usando o apenas o conjunto de características estilísticas que definiram anteriormente. Aqui destacou-se a presença de *Joke Words* como sendo a característica mais relevante na tarefa do reconhecimento de humor.

Já Mihalcea e Pulman (2007) estudaram duas questões. Inicialmente foi testada a hipótese de que textos humorísticos e textos não-humorísticos conseguiam ser distinguidos através de uma tarefa de classificação de texto que se baseava apenas nas características da linguagem. Auxiliaram-se também dos algoritmos Naive Bayes e *SVM* que foram selecionados pela performance obtida em outros trabalhos. A hipótese foi testada tendo em conta dois conjuntos de dados diferentes, um constituído por "*one-liners*" e outro constituído por piadas mais longas. A hipótese acabou por ser confirmada e, posteriormente, foi estudado que conjunto de características melhor caracterizavam os exemplos humorísticos.

Aqui chegaram a uma conclusão interessante, que foi o facto de a orientação negativa das frases ser considerada uma das características mais importantes para a deteção de humor. Normalmente associamos o humor a efeitos positivos o que nos poderia a levar a pensar que frases com conotação positiva seriam mais interessantes para reconhecer humor do que frases com conotação negativa.

Yang et al. (2015) aplicaram o algoritmo *Random Forest* (conjunto de *Decision Trees*) de modo a realizar o método de *10-fold cross validation* em dois conjuntos de dados. A incongruência destacou-se como sendo das características que mais contribui para o reconhecimento de humor. Para além da tentativa do reconhecimento de humor através das características específicas que foram usadas, foi também desenvolvido um método onde se tentava identificar a palavra, ou palavras, que desencadeavam o humor (*humor anchor*). Eram identificadas algumas palavras candidatas e, seguidamente, era aplicado um algoritmo. Cada frase tinha inicialmente uma pontuação humorística (*score*) que era calculada com o auxílio de um classificador treinado anteriormente. O algoritmo escolhia a palavra (do conjunto de palavras candidatas) que, quando retirada da frase, gerava um maior decremento na pontuação humorística atribuída. Na figura 3.1 está representado este processo para a frase "*I am glad that I know sign language. It is pretty handy.*".

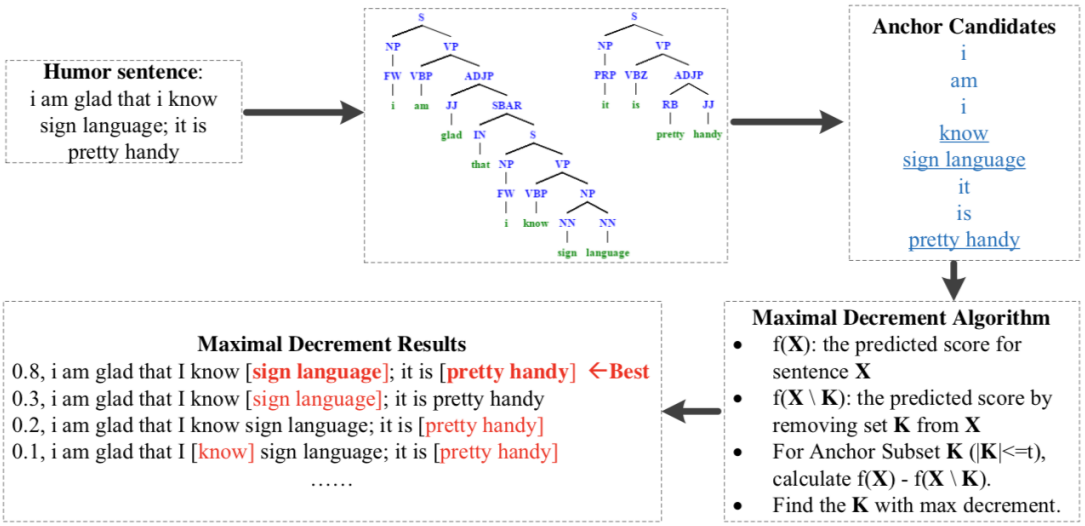

Figura 3.1: Processo de extração de palavras que desencadeiam humor (*Humor Anchor*) (Yang et al., 2015)

Barbieri e Saggion (2014) abordaram também o problema como uma classificação binária onde iriam apenas identificar se um "*tweet*" seria irónico/humorístico ou não. Escolheram para isso dois algoritmos baseadas em árvores de decisão, que foram o *Random Forest* e *Decision Tree*.

3.1.3 Datasets

Uma das componentes essencias para a tarefa de reconhecimento de humor é a construção de conjuntos de dados (*datasets*) com instâncias positivas e negativas que servirão de base para o classificador ser treinado e testado. As instâncias humorísticas utilizadas são, na grande parte dos casos, piadas curtas (*one-liners*) que são encontradas em inúmeras páginas e blogs na web. Já com os exemplos negativos é necessária especial atenção.

Escolher conjuntos de dados não humorísticos que, estruturalmente, sejam muito diferentes dos exemplos humorísticos pode facilitar demasiado a tarefa e levar a desempenhos elevados do classificador o que, contudo, pode ser enganador. Não queremos que o classificador aprenda a distinguir humor baseando-se apenas em diferenças em termos de extensão do texto ou diferenças de vocabulário.

Mihalcea e Strapparava (2005) recolheram dados humorísticos da web através de um algoritmo que permitia retirar um conjunto grande de piadas começando apenas com uma pequena lista (constituída por *one-liners* identificadas manualmente). O conjunto de exemplos positivos era constituído por um total de 16000 *one-liners*. Em relação aos dados não humorísticos tentaram selecionar frases que fossem semelhantes em termos de estrutura aos exemplos positivos. Foram retirados exemplos de três fontes distintas, sendo elas:

- **Títulos de notícias** da agência *Reuters* que consistiam em frases curtas e com uma sintaxe bastante simples (Lewis et al., 2004).
- **Provérbios** de uma coleção encontrada na web. Têm uma estrutura muito semelhante às piadas curtas, não tendo, no entanto, o efeito cómico.
- **Frases do *British National Corpus***: um *corpus* que contém frases de vários tipos, domínios e géneros. Foram selecionadas aquelas que tivessem o conteúdo mais parecido com as *one-liners* (Burnard, 1995).

Os conjuntos de dados utilizados em Mihalcea e Strapparava (2006) foram exatamente os mesmos, adicionando apenas mais uma fonte para os dados não humorísticos. Aqui foram também retiradas frases não humorísticas do *Open Mind Common Sense* que é uma coleção de 800000 de frases curtas em inglês (Singh (2002)) de onde foram utilizadas as primeiras 16000. Na figura 3.2 podemos ver exemplos de frases incluídas neste *dataset*.

One-liners
Take my advice; I don't use it anyway.
I get enough exercise just pushing my luck.
Beauty is in the eye of the beer holder.

Reuters titles
Trocadero expects tripling of revenues.
Silver fixes at two-month high, but gold lags.
Oil prices slip as refiners shop for bargains.

BNC sentences
They were like spirits, and I loved them.
I wonder if there is some contradiction here.
The train arrives three minutes early.

Proverbs
Creativity is more important than knowledge.
Beauty is in the eye of the beholder.
I believe no tales from an enemy's tongue.

OMCS sentences
Humans generally want to eat at least once a day.
A file is used for keeping documents.
A present is a gift, something you give to someone.

Figura 3.2: Exemplos de frases incluídas nos *datasets* utilizados por Mihalcea e Strapparava (2006)

Em Sjöbergh e Araki (2007) o conjunto de exemplos positivos é constituído por 6100 piadas retiradas da web. Depois de retiradas, todas as piadas foram manualmente revistas

de modo a perceber se realmente tinham uma conotação humorística. No que diz respeito ao conjunto de exemplos não humorísticos, foram também usadas frases do *British National Corpus* por ter sido o conjunto de dados mais difícil de distinguir das *one-liners* noutros estudos (Mihalcea e Strapparava (2005)). Dentro do *dataset* de exemplos negativos, todas as frases mais pequenas que a piada mais pequena e maiores que a piada maior, foram cortadas.

Mihalcea e Pulman (2007) usaram não só *one-liners* (10 a 15 palavras), mas também piadas mais longas (1000 a 10000 caracteres). O processo de recolha, tanto das *one-liners*, como das frases não humorísticas foi igual ao feito em Mihalcea e Strapparava (2006), tendo sido recolhidos exemplos das mesmas fontes. Já para as piadas mais longas, foram recolhidos 1125 artigos humorísticos publicados no jornal "*The Onion*". O correspondente conjunto de dados negativos foi criado tendo por base três diferentes fontes: artigos escritos no *Los Angeles Times*, textos extraídos do *British National Corpus* e notícias retiradas do *Foreign Broadcast Information Service*.

Tanto em Reyes, Rosso e Buscaldi (2009) como em Yang et al. (2015) foi utilizado o *dataset* criado em Mihalcea e Strapparava (2005), para os exemplos humorísticos. Adicionalmente, em Yang et al. (2015), foi ainda utilizado mais um conjunto de dados humorísticos que continha piadas retiradas de um website (*Pun of the Day*[5]). Já os dados não humorísticos usados em Yang et al. (2015) foram obtidos através de diferentes fontes onde se incluem notícias escritas no *New York Times* ou textos retirados do *Yahoo! Answers*.

Por último Barbieri e Saggion (2014) utilizaram um conjunto de dados constituído por 40000 "*tweets*" divididos igualmente em 4 diferentes tópicos: Ironia, Educação, Humor e Política. Estes "*tweets*" foram selecionados pela procura das respetivas *hashtags*, neste caso foram: *#irony*, *#education*, *#humour* e *#politics*.

3.2 Reconhecimento de Fenómenos Relacionados com o Humor

Para além do humor, existem muitas outras formas de comunicação que utilizam figuras da linguagem para demonstrar um significado que pode não ser literal. Dentro deste campo podemos incluir a ironia ou o sarcasmo. Existem vários trabalhos no domínio da deteção de ironia em texto e poderá ser interessante fazer a análise de alguns, visto o tema ter bastantes pontos em comum com o reconhecimento de humor.

Reyes, Rosso e Buscaldi (2012) propõe algumas características para o estudo do deteção de ironia. Entre elas está a polaridade, que acaba por ser uma propriedade comum tanto no uso da ironia, como no uso de humor. Tenta-se perceber se o uso de palavras com uma certa polaridade (positiva ou negativa) pode ser um indicador da presença de ironia. É ainda mencionado que a ironia pode também estar relacionada com a incongruência (situações não esperadas) ou com a ambiguidade, características que também são muito associadas ao humor.

O conjunto de dados que foi utilizado foi um conjunto de 50000 "*tweets*" dividido em 5 sub-conjuntos. Dos 5 sub-conjuntos (10000 "*tweets*" cada), 4 deles foram recolhidos tendo por base a pesquisa por *hashtag* específica, que no caso foram *#irony*, *#politics*, *#humor* e *#technology*. O último dos sub-conjuntos não tinha nenhuma restrição específica. A

[5] `https://www.punoftheday.com/` último acesso em Agosto de 2019

experiência dividiu-se em 2 fases. A primeira fase foi focada em representar cada frases do conjunto de dados tendo por base as características definidas. Na segunda fase foi feita a classificação do dados, onde se usou uma *Decision Tree*.

O objetivo do trabalho de Buschmeier, Cimiano e Klinger (2014) era detetar ironia em revisões feitas por utilizadores em alguns websites. Além da polaridade, que já foi mencionada anteriormente, foram propostas outras características de modo a auxiliar a deteção de ironia. Entre elas estão:

- **Hipérbole**: relacionada com a polaridade das palavras. A hipérbole era detetada quando existia uma sequência de 3 palavras com a mesma polaridade (positiva ou negativa).
- **Sinais de Pontuação**: existência de múltiplos sinais de exclamação numa frase.
- **Interjeições**: ocorrência de termos como "haha", "wow" ou "lol".

Para realizar a tarefa de classificação baseada nas características mencionadas foram usados 4 classificadores (*SVM*, Naive Bayes, *Random Forest* e *Decision Trees*). O conjunto de dados consistia em 1254 revisões de utilizadores no site da Amazon das quais 437 seriam irónicas.

Ao contrário do reconhecimento de humor, na deteção de ironia e de sarcasmo existe algum trabalho no que diz respeito à língua portuguesa.

No trabalho de Freitas et al. (2014) o objetivo passava por classificar "*tweets*" escritos em português como irónicos/sarcásticos. Para isso foram definidas algumas características que iriam auxiliar a avaliação dos dados, entre as quais:

- **Expressões**: além das expressões de riso ("hahaha") que podem indicar ironia, foram ainda tidas em conta expressões como "na boa", "só que não", "sim,"(a palavra "sim" seguida de uma vírgula pode representar ênfase no que vai ser dito a seguir) que quando usadas podem denotar ironia presente na frase.
- **Hashtags**: "*tweets*" que continham hashtags como *#ironia*, *#sarcasmo*, *#joking* e *#kidding* são fortes candidatos a conter ironia ou sarcasmo no seu conteúdo.
- **Lista de *Emoticons***: foi também considerada uma lista de *emoticons* que expressam um sentido humorístico.
- **Sinais de Pontuação**: utilização excessiva de sinais de interrogação ("????"), de exclamação ("!!!!") ou mesmo de ambos ("!!!???") nos "*tweets*" pode indicar um conteúdo irónico ou humorístico.

A construção do conjunto de dados que iriam ser analisados por Freitas et al. (2014) passou por recolher "*tweets*" que continham a expressão "Fim do mundo", isto por ser um dos temas mais falados no momento da recolha (rumores do apocalipse Maia em 2012) e em que muitos tinham probabilidade de serem irónicos ou sarcásticos. Todos os "*tweets*" da base de dados foram primeiro anotados manualmente. Depois deste processo foi feita a classificação automática dos dados.

Carvalho et al. (2009) exploraram algumas características da linguagem que estão associadas ao uso de ironia em Português. Dentro dessas características podemos mencionar alguns exemplos interessantes:

- **Formas diminutivas**: os diminutivos das palavras são muitas vezes usados em Português para demonstrar um sentimento positivo ou de afeto. Ainda assim podem também ser usados num contexto sarcástico ou irónico de modo a expressar depreciação em relação a algo.
- **Interjeições**: podem conter informação importante acerca do sentimento ou emoções de quem as escreve. Os autores analisaram um pequeno conjunto de interjeições ("bravo", "força", "obrigadinho", ...) de modo a aferir se seriam relevantes na deteção de ironia.
- **Expressões de Riso**: visto que os dados analisados neste estudo seriam dados gerados por utilizadores na web, o uso de expressões como "LOL" ou de onomatopeias como "AH" ou "EH" foi visto também como uma pista para o reconhecimento de ironia.

O conjunto de dados utilizado por Carvalho et al. (2009) foi recolhido de sites de jornais portugueses de onde se retiraram cerca de 250000 *posts* escritos pelos utilizadores, o que perfazia um total de mais de 1 milhão de frases. Posteriormente a classificação poderia ser feita definindo se o texto era irónico, não irónico, indefinido ou mesmo ambíguo.

Capítulo 4

Reconhecimento Baseado em Características Lexicais

O objetivo principal deste trabalho é o desenvolvimento de um modelo computacional que tenha capacidade de reconhecer humor verbal na língua portuguesa. Numa primeira fase o problema do reconhecimento de humor foi abordado como um problema de classificação de texto tradicional, baseado apenas em características do texto, onde será aplicado um método de aprendizagem computacional supervisionada. Para isso foram construídos conjuntos de dados que nos permitissem treinar e testar o modelo criado. Todos os conjuntos de dados são constituídos por textos curtos e obtidos a partir de diversas fontes na Web.

Neste capítulo serão descritos todos os conjuntos de dados utilizados e as representações usadas para as características lexicais. Por último são relatadas as primeiras experiências feitas com o modelo criado e analisados os resultados obtidos.

4.1 Conjuntos de Dados

De modo a desenvolver um classificador com base em aprendizagem computacional supervisionada para o reconhecimento de humor, é necessário que existam dados onde se possam treinar e testar o modelo. Para esse efeito foram construídos três conjuntos de dados, cada um com exemplos de instâncias positivas (humorísticos) e exemplos de instâncias negativas (não humorísticos).

Tendo em conta que se iria realizar uma classifação com base em textos curtos, todos os exemplos recolhidos teriam de seguir essa estrutura. Devido ao facto de existirem poucas fontes de onde se possam obter frases com conotação humorística escritas em português, estes foram recolhidos em primeiro lugar. Posterioremente foram selecionados os exemplos negativos. Aqui foi necessário ter o cuidado de recolher exemplos negativos que fossem o mais parecido possível com os exemplos positivos, e onde a principal diferença fosse a presença ou não do registo humorístico. O objetivo será que o modelo criado consiga reconhecer humor e não apenas diferenciar os exemplos através de características que não são relevantes no caso do nosso problema.

De seguida são descritos os três conjuntos de dados que foram utilizados durante este trabalho. Para cada um deles são mencionados o número de instâncias positivas e negativas existentes, a sua estrutura e ainda as fontes de onde foram retiradas. Existiu também a preocupação de balancear todos os conjuntos de dados criados, uma vez que ao exis-

tirem mais exemplos de uma determinada classe, o modelo, por vezes, pode favorecer a classe maioritária levando a que depois tenhamos valores de exatidão (em inglês, *accuracy*) enganadores, daí a tentativa de balanceamento de todos os conjuntos de dados.

4.1.1 Primeiro Conjunto de Dados (D1)

O conjunto de dados D1 é constituído por 1400 instâncias, mais propriamente 700 exemplos humorísticos e 700 exemplos não humorísticos. Os exemplos humorísticos são compostos por piadas curtas (em inglês, *one-liners*), todas com uma estrutura de pergunta/resposta, retiradas da Web. As *One-Liners* foram obtidas, na sua maioria, de uma coleção de anedotas escritas em Português com o nome de "Anedotário Português"[1]. Além da fonte já mencionada, foram também recolhidas algumas piadas de uma página de Facebook denominada "O Sagrado Caderno das Piadas Secas"[2].

De modo a garantirmos resultados que nos permitissem aferir realmente se o nosso modelo conseguiria reconhecer humor, os exemplos negativos deveriam ser também do tipo pergunta/resposta. Os primeiros exemplos negativos foram retirados de um *corpus*, Multieight-04 (Magnini et al., 2004), que contém 700 perguntas e respostas de cultura geral escritas em várias línguas, de onde se selecionaram apenas as escritas em Português. Um dos problemas desde logo detetado foi o facto de a palavra "porque" aparecer bastantes vezes nos exemplos positivos ao contrário do que acontecia nos exemplos negativos, onde quase não aparecia. Isto faria com que, baseado apenas na presença da palavra "porque" numa frase, o classificador conseguisse facilmente distinguir as instâncias positivas das instâncias negativas. De modo a combater este problema, foram recolhidos mais perguntas/respostas sem conotação humorística desta vez oriundas de um website[3] onde a maioria dos exemplos continham a palavra "porque".

Houve uma maior facildade na recolha de exemplos negativos do que na recolha de exemplos positivos, como era inicialmente previsto. Isto fez com que o número de exemplos negativos recolhidos fosse significativamente maior que o número de exemplos positivos. Visto que tínhamos 700 exemplos positivos o objetivo seria ter o mesmo número de exemplos negativos, logo teria de haver uma limitação dos exemplos negativos. Uma das técnicas utilizadas normalmente para combater problemas de balanceamento em conjuntos de dados é a seleção de uma subamostra da classe com maior número de exemplos (em inglês, *undersampling*). A intenção seria então selecionar uma subamostra de todos os exemplos negativos que fosse composta por 700 instâncias. Para isso foram escolhidos aleatoriamente 350 exemplos negativos de cada fonte utilizada (Multieight-04 e "*osporques.com*"). O conjunto de dados D1 ficou então constituído por:

- 700 exemplos positivos (*One-Liners*)
- 700 exemplos negativos:
  - 350 do *corpus* Multieight-04
  - 350 do website "*osporques.com*"

A tabela 4.1 mostra alguns exemplos de instâncias contidas no conjunto de dados D1. A tabela inclui o texto e ainda a classe a que o mesma pertence: 'H' para frases humorísticas e 'N' para frases não humorísticas.

[1] `https://ltpf.files.wordpress.com/2011/01/omaiscompleto-anedotc3a3c2a1ri.pdf` último acesso em Agosto de 2019

[2] `https://www.facebook.com/CadernoDasPiadas/` último acesso em Agosto de 2019

[3] `http://osporques.com/` último acesso em Maio de 2019

Exemplos	**Classe**
Qual é a língua menos falada no mundo? Língua Gestual.	H
Porque é que os polícias não gostam de sabão? Porque preferem deter gente.	H
O que é um byfe? São oito bifes.	H
Como se chama o filho do escritor e Prémio Nobel Thomas Mann? Golo Mann.	N
O que é a UNICEF? Fundo das Nações Unidas para a Infância.	N
Porque o vento nos refresca ? O vento refresca-nos por uma razão muito simples, é pelo facto de levantar a camada fina de pelo que está espalhada pelo nosso corpo todo.	N

Tabela 4.1: Exemplos de instâncias contidas em D1

4.1.2 Segundo Conjunto de Dados (D2)

O conjunto de dados D2 surgiu como uma tentativa de incluir outro estilo de humor neste trabalho. Em D1 todos os exemplos seguem uma estrutura de pergunta/resposta, sendo que em D2 a abordagem acabou por ser diferente. O objetivo passou por treinar e testar o modelo com exemplos humorísticos, também em piadas curtas, mas que não fossem do tipo pergunta/resposta. Não existem muitas fontes online em que se tenha acesso a *One-Liners*, sendo que as que foram encontradas eram maioritariamente do tipo pergunta/resposta. A solução encontrada foi a utilização de títulos de notícia humorísticos com recurso à rede social Twitter, especificamente à página do "Inimigo Público"[4]. Esta página é, basicamente, um suplemento do jornal "Público" mas que as notícias têm sempre uma conotação humorística.

O conjunto de dados D2 contém um total de 4000 frases, onde houve também um balanceamento das instâncias positivas e negativas, ou seja, em D2 temos 2000 exemplos positivos assim como 2000 exemplos negativos. Foram retirados 2000 títulos de notícia deste tipo correspondentes aos meses de dezembro de 2018 até fevereiro de 2019. Todos os *tweets* recolhidos foram previamente "limpos", descartando os referentes a *retweets* e retirando partes que não pertenciam ao corpo da notícia ("Hoje na edição impressa:", por exemplo).

Sendo o objetivo ter exemplos negativos semelhantes aos positivos, nada melhor que usar também títulos de notícia mas desta vez sem nenhuma conotação humorística. A primeira fonte de instâncias negativas de D2 foi então a página do jornal "Público"[5] no Twitter. De modo a que os temas referentes às notícias retiradas para os exemplos positivos e negativos não fossem muito diferentes, os *tweets* recolhidos do jornal "Público" pertenciam ao mesmo espaço temporal que os recolhidos do "Inimigo Público". Juntaram-se um total de 1000 *tweets* retirados do jornal que passaram igualmente pelo processo de "limpeza" já referido para os exemplos positivos. As restantes 1000 instâncias negativas correspondem a provérbios da língua portuguesa obtidos através do Dicionário aberto de Calão e Expressões Idiomáticas[6] disponibilizados online, na página do projeto Natura, da Universidade do Minho.

No final, o D2 ficou então composto por:

- 2000 exemplos positivos (Inimigo Público)
- 2000 exemplos negativos:
  - 1000 notícias do jornal "Público"

[4] `https://twitter.com/inimigo` último acesso em Agosto de 2019
[5] `https://twitter.com/Publico` último acesso em Agosto de 2019
[6] `https://natura.di.uminho.pt/~jj/pln/proverbio.dic` último acesso em Agosto de 2019

- 1000 provérbios portugueses

Na tabela 4.2 estão alguns exemplos de instâncias humorísticas e não humorísticas contidas no conjunto de dados D2:

Exemplos	**Classe**
PS quer refazer acordo à esquerda ou ameaça com aliança à direita com a equipa de futebol do Canelas 2010.	H
Espectáculo no MEO Sudoeste cancelado porque DJ perdeu a pen.	H
Marcelo vai visitar a nado todas as ilhas da Grécia.	H
Ministério das Finanças ainda não recebeu pedidos de pré-reforma no Estado.	N
Derrocada sem causar danos em estrada em São Miguel.	N
Mais vale ter um pássaro na mão do que dois a voar.	N

Tabela 4.2: Exemplos de instâncias contidas em D2

4.1.3 Terceiro Conjunto de Dados (D3)

Um terceiro conjunto de dados surgiu da união de instâncias em D1 e D2 e tinha como objetivo chegar a um modelo que conseguisse reconhecer humor de uma forma mais generalizada.

Começando pelos exemplos positivos, em D1 existem 700 *One-Liners* (perguntas e respostas) e em D2 existem 2000 títulos de notícia humorísticos (Inimigo Público). De modo a haver um balanceamento entre exemplos vindos das duas fontes, o conjunto de dados D3 acabou por contar com 1400 instâncias positivas. Foram usadas todas as 700 piadas de D1 e foram ainda selecionados aleatoriamente 700 títulos do "Inimigo Público" contidos em D2.

Em relação a exemplos negativos, contamos com quatro diferentes fontes. Temos 350 frases do *corpus* Multieight-04 e 350 frases do website "*osporques.com*" em D1 e ainda 1000 títulos do jornal "Público" e 1000 provérbios em D2. Já que os exemplos positivos perfaziam um total de 1400 frases, foram escolhidas também 1400 frases para exemplos negativos de modo a garantir mais uma vez o balanceamento do conjunto de dados. Do mesmo modo que foi feito para os exemplos positivos, foram retirados exemplos em igual quantidade das quatro diferentes fontes de exemplos negativos.

Finalmente o conjunto de dados D3 ficou composto por:

- 1400 exemplos positivos:
  - 700 *One-Liners*
  - 700 títulos de notícia humorísticos (Inimigo Público)
- 1400 exemplos negativos:
  - 350 notícias do jornal "Público"
  - 350 provérbios portugueses
  - 350 frases do *corpus* Multieight-04
  - 350 frases do website "*osporques.com*"

4.2 Características Lexicais

A abordagem inicial ao problema foi a de tentar reconhecer humor apenas com base em características lexicais presentes nos textos, que seriam fornecidos ao modelo. Estas características lexicais não são mais que as palavras que constituem esses mesmos textos. Além das palavras na sua forma normal, para cada frase (ou documento) é feita a lematização das suas palavras, ou seja, a variante da palavra que aparece no dicionário. Portanto, além de se considerar as palavras de um documento como características, considera-se ainda a forma lematizada dessas palavras.

De modo a obter a forma lematizada de cada palavra seria necessário realizar um pré-processamento ao texto que foi feito recorrendo ao *toolkit* NLPyPort (Ferreira, Gonçalo Oliveira e Rodrigues, 2019) otimizado para português. Dentro do pré-processamente existiram as etapas de tokenização (divisão do texto em *tokens*), *PoS Tagging* (identificação da função gramatical de cada palavra) e só depois de lematização. Consideremos então o seguinte documento assim como o resultado da sua lematização:

Marcelo vai visitar a nado todas as ilhas da Grécia.

Marcelo ir visitar o nado todo o ilha de o Grécia.

Neste caso específico, e descartando as palavras com apenas uma letra, as caraterísticas lexicais a considerar são:

[“marcelo”, “vai”, “visitar”, “nado”, “todas”, “as”, “ilhas”, “da”, “grécia”, “ir”, “todo”, “ilha”, “de”]

Este processo é repetido para todos os documentos presentes num qualquer conjunto de dados onde, no final, temos todas as palavras (e os seus lemas) que são consideradas como características.

Para além de considerarmos apenas uma palavra (unigrama) como uma característica lexical podemos também considerar diferentes números de sequências (n-gramas) de palavras, procurando assim capturar informação linguística mais rica. Para o mesmo exemplo considerado em cima, caso se utilizem bigramas e trigramas de palavras (nestes casos consideramos apenas a forma normal da palavra, descartando os lemas), a lista de características finais seria:

[“marcelo”, “vai”, “visitar”, “nado”, “todas”, “as”, “ilhas”, “da”, “Grécia”, “ir”, “todo”, “ilha”, “de”, “marcelo vai”, “vai visitar”, “visitar nado”, “nado todas”, “todas as”, “as ilhas”, “ilhas da”, “da grécia”, “marcelo vai visitar”, “vai visitar nado”, “visitar nado todas”, “nado todas as”, “todas as ilhas”, “as ilhas da”, “ilhas da grécia”]

Dado que os modelos de aprendizagem computacional estão preparados para lidar com números, e não diretamente com texto, o próximo passo será converter as palavras para números. Existem diferentes formas de representar as características como números. A forma mais simples será a representação de cada palavra pela contagem do número de vezes que a mesma ocorre num determinado documento. Considerando um pequeno conjunto de documentos:

Isto é um documento de exemplo!

Este é o segundo exemplo.

Terceiro documento.

As características consideradas neste caso são:

["de", "documento", "este", "exemplo", "isto", "segundo", "terceiro", "um", "ser"]

Pegando nesta lista de características, cada documento do conjunto de dados é representado por um vetor que, em cada posição, tem o número de vezes que cada palavra considerada, correspondente à mesma posição na lista, ocorre. O resultado final é uma matriz n x m onde n é o número de documentos do conjunto de dados e m é o número de características consideradas. Neste caso a matriz final, X, seria:

$$X = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Pode também ser usado o algoritmo TF-IDF, onde, em vez da contagem do número de vezes que uma palavra ocorre, é considerada a frequência relativa de uma palavra num documento comparando depois com a proporção inversa dessa mesma palavra em relação ao conjunto de todos os documentos. A matriz, X, neste caso ficaria:

$$X = \begin{bmatrix} 0.49 & 0.37 & 0 & 0.37 & 0.49 & 0 & 0 & 0.49 & 0.35 \\ 0 & 0 & 0.62 & 0.47 & 0 & 0.62 & 0 & 0 & 0.43 \\ 0 & 0.61 & 0 & 0 & 0 & 0 & 0.80 & 0 & 0 \end{bmatrix}$$

4.3 Experiências

Nesta fase queremos explorar as caraterísticas lexicais para treinar classificadores que consigam descriminar entre textos humorísticos e não-humorísticos avaliando posteriormente o desempenho dos classificadores aprendidos em dados de teste. Para perceber o desempenho que obteve o classificador serão calculadas algumas métricas: *accuracy*, *precision*, *recall* e F1.

O objetivo será comparar o desempenho dos vários algoritmos utilizados com os diferentes parâmetros aplicados. Dentro destes pârametros estão incluídos os algoritmos de aprendizagem, as diferentes representações das características lexicais, a quantidade de características usadas (incluir bigramas ou trigramas de palavras). É feita ainda uma seleção das características mais relevantes de modo a perceber o número ideal de características que o modelo deve considerar.

Em todas as experiências realizadas existiu sempre uma divisão dentro dos conjuntos de dados em que 80% dos exemplos serviam para treino e validação do modelo, através de *10-fold cross validation*, e os restantes 20% serviam como exemplos para o teste. Dentro desta divisão entre exemplos para treino e teste garantiu-se sempre o balanceamento dos dados. Devido a este balanceamento as métricas usadas como referência serão os valores de *accuracy* e de F1.

De seguida são descritas todas as experiências realizadas e apresentados os respetivos resultados. É também feita uma análise aos resultados obtidos de modo a retirarmos algumas conclusões.

4.3.1 Algoritmo de Aprendizagem e Representação de Características

A primeira experiência seria então a de perceber com que algoritmo de aprendizagem seria possível construir o modelo com melhores resultados para o nosso problema, assim como escolher a melhor representação para as características lexicais. O facto de optarmos por uma combinação de algoritmo e representação das características será benéfico na medida em que se podem fazer todas as experiências seguintes com base nesta escolha, em vez de se testar para todas as combinações possíveis. A escolha dos algoritmos de aprendizagem que iriam ser utilizados recaiu sobre aqueles que foram os mais utilizados nos trabalhos relacionados. Foram usadas as implementações do *scikit-learn* (biblioteca de aprendizagem computacional para Python) para os algoritmos escolhidos, sendo eles:

- **SVM**: foi usada a classe *SVC()* com um *kernel* linear, tendo sido usados todos os outros parâmetros com os valores padrão
- ***Naive Bayes* (NB)**: usada a classe *MultinomialNB()* com todos os parâmetros a assumirem os valores padrão
- ***Decision Tree* (DT)**: utilizada a classe *DecisionTreeClassifier()* com todos os parâmetros a assumirem os valores padrão
- ***Random Forest* (RF)**: utilizada a classe *RandomForestClassifier()* com todos os parâmetros a assumirem os valores padrão

Cada um dos algoritmos mencionados irá ser testado recorrendo a duas representações diferentes das características lexicais:

- **Contagem (*Count*)**: aplicada a função *CountVectorizer()* do *scikit-learn*, que recebia os valores de frequência mínima (parâmetro *min_df*) e de frequência máxima (parâmetro *max_df*). Todos os restantes parâmetros da função assumiam os valores padrão.
- **TF-IDF**: aplicada a função *TfidfVectorizer()* do *scikit-learn*. Recebia também os valores de frequência mínima (*min_df*) e de frequência máxima (*max_df*). Todos os restantes parâmetros da função assumiam os valores padrão.

Dentro das características lexicais houve ainda uma pequena limitação, de modo a que não consideremos muitas características que não têm preponderância no contexto do nosso problema. Definiu-se então que as características consideradas teriam de existir pelo menos em dois documentos dentro do conjunto de dados (frequência mínima, correspondente ao parâmetro *min_df*), e ainda que não podiam existir em mais de 75% dos documentos presentes no conjunto de dados (frequência máxima, correspondente ao parâmetro *max_df*). Com valor de frequência mínima garantimos que palavras que aparecem apenas uma vez em um documento, por isso demasiado específicas para descriminar, são descartadas. Já com o valor de frequência máxima estamos a descartar as palavras que aparecem em muitos documentos e que, por isso, não serão suficientemente relevantes para descriminar entre argumentos. As palavras mais frequentes são normalmente palavras funcionais (e.g.,

determinantes, preposições), que contribuem pouco para o significado dos documentos e são, normalmente, consideradas *stopwords* e ignoradas em problemas de classificação.

As tabelas 4.3, 4.4, 4.5, 4.6, 4.7 e 4.8 apresentam os resultados do treino/validação (obtidos com recurso a um *10-fold cross validation*) e teste para os diferentes conjuntos de dados, com os diferentes algoritmos de aprendizagem e as diferentes representações das características lexicais. Em cada tabela estão apresentados os valores de *accuracy*, *precision*, *recall* e de F1.

	Treino				**Teste**			
	SVM	**NB**	**DT**	**RF**	**SVM**	**NB**	**DT**	**RF**
Accuracy	0.89 ± 0.03	$\mathbf{0.92 \pm 0.03}$	0.85 ± 0.03	0.87 ± 0.02	0.83	**0.84**	0.80	0.82
Precision	0.89 ± 0.04	$\mathbf{0.91 \pm 0.04}$	0.87 ± 0.04	0.90 ± 0.04	0.84	0.83	0.86	**0.88**
Recall	0.89 ± 0.03	$\mathbf{0.93 \pm 0.03}$	0.83 ± 0.04	0.84 ± 0.06	0.80	**0.85**	0.76	0.74
F1	0.89 ± 0.03	$\mathbf{0.92 \pm 0.03}$	0.85 ± 0.03	0.87 ± 0.03	0.82	**0.84**	0.81	0.80

Tabela 4.3: Resultados de treino e teste para D1 usando *Count* para a representação das características lexicais

	Treino				**Teste**			
	SVM	**NB**	**DT**	**RF**	**SVM**	**NB**	**DT**	**RF**
Accuracy	$\mathbf{0.93 \pm 0.02}$	$\mathbf{0.93 \pm 0.02}$	0.81 ± 0.02	0.88 ± 0.02	**0.85**	**0.85**	0.81	0.81
Precision	$\mathbf{0.94 \pm 0.02}$	0.92 ± 0.03	0.83 ± 0.03	0.92 ± 0.04	0.84	0.83	0.78	**0.88**
Recall	0.92 ± 0.04	$\mathbf{0.93 \pm 0.02}$	0.79 ± 0.04	0.83 ± 0.04	0.85	**0.87**	0.86	0.71
F1	$\mathbf{0.93 \pm 0.02}$	$\mathbf{0.93 \pm 0.01}$	0.81 ± 0.02	0.87 ± 0.03	**0.85**	**0.85**	0.82	0.79

Tabela 4.4: Resultados de treino e teste para D1 usando TF-IDF para a representação das características lexicais

Começamos por analisar os resultados obtidos em D1. Olhando para a tabela 4.3, onde o modelo é testado usando a contagem como representação das características lexicais, os melhores valores de *accuracy* e de F1 obtiveram-se quando se usou o algoritmo Naive Bayes com um resultado de 84% para ambos. Muito perto destes valores ficou o algoritmo SVM, com resultados de 83% e 82% para a *accuracy* e F1, respetivamente. Ainda que não muito distantes dos valores já referidos, com o uso dos algoritmos DT e RF obtiveram-se os piores resultados. Aquando do uso do algoritmo TF-IDF para a representação das características foram obtidos os resultados mais altos, como mostra a tabela 4.4. Atingiu-se o máximo de 85% para os valores de *accuracy* e F1, usando tanto SVM como NB. Os algoritmos DT e RF acabaram por ser novamente os que conseguiram piores resultados nestes pârametros.

	Treino				**Teste**			
	SVM	**NB**	**DT**	**RF**	**SVM**	**NB**	**DT**	**RF**
Accuracy	$\mathbf{0.86 \pm 0.03}$	0.84 ± 0.02	0.73 ± 0.03	0.78 ± 0.02	**0.85**	0.81	0.74	0.76
Precision	$\mathbf{0.89 \pm 0.02}$	0.81 ± 0.02	0.74 ± 0.03	0.88 ± 0.03	**0.87**	0.75	0.76	0.85
Recall	0.83 ± 0.05	$\mathbf{0.90 \pm 0.03}$	0.72 ± 0.04	0.64 ± 0.04	0.82	**0.93**	0.72	0.62
F1	$\mathbf{0.86 \pm 0.03}$	0.85 ± 0.02	0.73 ± 0.03	0.74 ± 0.03	**0.85**	0.83	0.74	0.72

Tabela 4.5: Resultados de treino e teste para D2 usando *Count* para a representação das características lexicais

	Treino				**Teste**			
	SVM	**NB**	**DT**	**RF**	**SVM**	**NB**	**DT**	**RF**
Accuracy	$\mathbf{0.86 \pm 0.01}$	0.84 ± 0.01	0.76 ± 0.02	0.81 ± 0.02	**0.83**	0.80	0.68	0.71
Precision	0.86 ± 0.01	0.80 ± 0.01	0.78 ± 0.03	$\mathbf{0.87 \pm 0.01}$	0.78	0.74	0.75	**0.87**
Recall	0.87 ± 0.03	$\mathbf{0.91 \pm 0.02}$	0.72 ± 0.03	0.74 ± 0.05	0.90	**0.93**	0.54	0.49
F1	$\mathbf{0.86 \pm 0.02}$	0.85 ± 0.01	0.74 ± 0.02	0.80 ± 0.03	**0.84**	0.83	0.63	0.63

Tabela 4.6: Resultados de treino e teste para D2 usando TF-IDF para a representação das características lexicais

Em relação a D2 foram obtidos resultados muito semelhantes aos conseguidos em D1. Com a análise da tabela 4.5 conseguimos perceber que o algoritmo SVM acabou por se destacar dos restantes obtendo valores de 85% para *accuracy* e F1 seguindo-se o NB (81% *accuracy*, 83% F1). Neste caso os algoritmos DT e RF apresentaram resultados claramente inferiores que os dois primeiros algoritmos. Para o mesmo conjunto de dados, mas utilizando agora o algorimto TF-IDF, os resultados foram praticamente os mesmos, como atesta a tabela 4.6. O algoritmo SVM voltou a apresentar os melhores resultados, aparecendo depois o NB.

	Treino				**Teste**			
	SVM	**NB**	**DT**	**RF**	**SVM**	**NB**	**DT**	**RF**
Accuracy	$\mathbf{0.84 \pm 0.02}$	0.83 ± 0.02	0.76 ± 0.02	0.81 ± 0.01	**0.77**	**0.77**	0.72	0.74
Precision	0.87 ± 0.04	0.80 ± 0.03	0.76 ± 0.02	$\mathbf{0.88 \pm 0.04}$	0.83	0.75	0.75	**0.87**
Recall	0.81 ± 0.03	$\mathbf{0.90 \pm 0.03}$	0.75 ± 0.05	0.72 ± 0.03	0.68	**0.81**	0.68	0.56
F1	$\mathbf{0.84 \pm 0.02}$	$\mathbf{0.84 \pm 0.02}$	0.76 ± 0.02	0.79 ± 0.01	0.75	**0.78**	0.71	0.68

Tabela 4.7: Resultados de treino e teste para D3 usando *Count* para a representação das características lexicais

	Treino				**Teste**			
	SVM	**NB**	**DT**	**RF**	**SVM**	**NB**	**DT**	**RF**
Accuracy	$\mathbf{0.85 \pm 0.02}$	0.83 ± 0.03	0.73 ± 0.03	0.78 ± 0.03	**0.78**	**0.78**	0.68	0.71
Precision	$\mathbf{0.86 \pm 0.02}$	0.81 ± 0.03	0.74 ± 0.03	$\mathbf{0.86 \pm 0.05}$	0.78	0.76	0.71	**0.87**
Recall	0.84 ± 0.04	$\mathbf{0.88 \pm 0.04}$	0.72 ± 0.06	0.68 ± 0.04	0.79	**0.82**	0.61	0.51
F1	$\mathbf{0.85 \pm 0.02}$	0.84 ± 0.03	0.73 ± 0.03	0.76 ± 0.04	0.78	**0.79**	0.66	0.64

Tabela 4.8: Resultados de treino e teste para D3 usando TF-IDF para a representação das características lexicais

Por último, para o conjunto de dados D3 os resultados foram um pouco piores. O maior valor de *accuracy* conseguido foi de 78% (SVM e NB com TF-IDF) e o de F1 foi de 79% (NB com TF-IDF, SVM conseguiu 78%). Tanto em D1 como em D2 existe apenas uma fonte de exemplos positivos, que seguem todos a mesma estrutura, e duas fontes de exemplos negativos. No entanto, em D3, isso não acontece. O facto de D3 ser constituído por instâncias retiradas de um maior número de fontes que os outros conjuntos de dados, faz com que exista uma maior diversidade de exemplos neste conjunto de dados. Logo, essa diversidade, faz com que se torne mais complicado para o modelo conseguir melhores resultados, acabando por se perceber a performance aqui obtida.

Analisando os resultados de uma forma geral, fica claro que os algoritmos de aprendizagem SVM e NB são os que apresentam melhores resultados para todos os conjuntos de dados e representação de caraterísticas. Em relação à representação das características lexicais, quando é usado o TF-IDF os resultados são ligeiramente melhores em D1 e D3, e

ligeiramente piores em D2. Em todas as experiências seguintes será usada a combinação de SVM e TF-IDF por ter sido a mais constante e com melhores resultados no geral.

4.3.2 Intervalo de N-Gramas

Nas primeiras experiências, as características lexicais usadas eram compostas apenas por uma palavra (unigramas). No entanto, a combinação de algumas palavras pode dar origem a características importantes de modo a que consigamos reconhecer humor. Além de usar apenas unigramas, a intenção passava também por perceber que número de palavras (N) combinadas resultaria em características lexicais que dariam melhores resultados para o nosso problema.

Foram então definidas cinco combinações diferentes:

- **Apenas Unigramas (n=1)**: usada nas experiências iniciais. Características são compostas pelas palavras e respetivos lemas.
- **Unigramas e Bigramas (n=1 ou n=2)**: aos unigramas juntam-se bigramas de palavras.
- **Unigramas, Bigramas e Trigramas (n=1 ou n=2 ou n=3)**: à combinação anterior juntam-se trigramas de palavras.
- **Apenas Bigramas (n=2)**: características lexicais são apenas bigramas de palavras.
- **Bigramas e Trigramas (n=2 ou n=3)**: adição dos trigramas à combinação anterior.

Os valores de frequência mínima e de frequência máxima usados são os mesmos, independentemente de se usar unigramas, bigramas ou trigramas. Como na experiência inicial, os valores de frequência mínima e de frequência máxima passados à função *TfidfVectorizer* são 2 e 0.75, respetivamente. Para a obtenção dos bigramas e trigramas foi necessário passar mais um parâmetro à função, o *ngram_range*. Este parâmetro é composto por dois valores, que são o limite inferior e o limite superior para os valores de ngramas a serem extraídos, sendo o valor padrão *(1,1* onde só se extraem unigramas. Logo, para se extraírem unigramas e bigramas o valor do pârametro teria de ser *(1,2)*, adicionando também trigramas ficaria *(1,3)*. Para se extrairem apenas bigramas o valor seria de *(2,2)* e por último, bigramas e trigramas teria de ser *(2,3)*. As tabelas 4.9, 4.10, 4.11, 4.12, 4.13 e 4.14 apresentam os resultados obtidos nas experiências:

	1	**1 ou 2**	**1 ou 2 ou 3**	**2**	**2 ou 3**
Accuracy	0.92 ± 0.03	$\mathbf{0.94 \pm 0.02}$	$\mathbf{0.94 \pm 0.03}$	0.91 ± 0.04	0.91 ± 0.02
Precision	0.93 ± 0.02	$\mathbf{0.96 \pm 0.03}$	0.95 ± 0.03	0.94 ± 0.04	0.95 ± 0.03
Recall	0.91 ± 0.06	$\mathbf{0.93 \pm 0.03}$	$\mathbf{0.93 \pm 0.04}$	0.87 ± 0.06	0.88 ± 0.03
F1	0.92 ± 0.03	$\mathbf{0.94 \pm 0.02}$	$\mathbf{0.94 \pm 0.03}$	0.91 ± 0.05	0.91 ± 0.03

Tabela 4.9: Resultados de treino para D1 usando diferentes gramas de palavras como características lexicais

	1	**1 ou 2**	**1 ou 2 ou 3**	**2**	**2 ou 3**
Accuracy	0.85	**0.86**	**0.86**	0.79	0.78
Precision	0.84	0.89	0.88	**0.94**	0.93
Recall	**0.85**	0.83	0.83	0.63	0.60
F1	0.85	**0.86**	0.85	0.75	0.73

Tabela 4.10: Resultados de teste para D1 usando diferentes gramas de palavras como características lexicais

	1	**1 ou 2**	**1 ou 2 ou 3**	**2**	**2 ou 3**
Accuracy	$\mathbf{0.86 \pm 0.02}$	$\mathbf{0.86 \pm 0.01}$	$\mathbf{0.86 \pm 0.02}$	0.80 ± 0.02	0.80 ± 0.01
Precision	0.85 ± 0.03	$\mathbf{0.86 \pm 0.01}$	$\mathbf{0.86 \pm 0.03}$	0.85 ± 0.03	0.84 ± 0.01
Recall	$\mathbf{0.87 \pm 0.03}$	0.86 ± 0.02	0.85 ± 0.04	0.74 ± 0.04	0.74 ± 0.02
F1	$\mathbf{0.86 \pm 0.02}$	$\mathbf{0.86 \pm 0.01}$	$\mathbf{0.86 \pm 0.02}$	0.79 ± 0.03	0.79 ± 0.01

Tabela 4.11: Resultados de treino para D2 usando diferentes gramas de palavras como características lexicais

	1	**1 ou 2**	**1 ou 2 ou 3**	**2**	**2 ou 3**
Accuracy	0.83	**0.84**	0.83	0.77	0.77
Precision	0.78	0.81	0.81	**0.85**	**0.85**
Recall	**0.90**	0.87	0.87	0.66	0.65
F1	**0.84**	**0.84**	**0.84**	0.74	0.74

Tabela 4.12: Resultados de teste para D2 usando diferentes gramas de palavras como características lexicais

	1	**1 ou 2**	**1 ou 2 ou 3**	**2**	**2 ou 3**
Accuracy	0.85 ± 0.03	$\mathbf{0.86 \pm 0.02}$	$\mathbf{0.86 \pm 0.02}$	0.82 ± 0.03	0.83 ± 0.01
Precision	0.86 ± 0.02	0.87 ± 0.02	$\mathbf{0.89 \pm 0.02}$	0.86 ± 0.04	0.88 ± 0.02
Recall	$\mathbf{0.84 \pm 0.04}$	0.83 ± 0.04	$\mathbf{0.84 \pm 0.02}$	0.76 ± 0.03	0.76 ± 0.02
F1	0.85 ± 0.03	0.85 ± 0.03	$\mathbf{0.86 \pm 0.02}$	0.80 ± 0.03	0.81 ± 0.02

Tabela 4.13: Resultados de treino para D3 usando diferentes gramas de palavras como características lexicais

	1	**1 ou 2**	**1 ou 2 ou 3**	**2**	**2 ou 3**
Accuracy	0.78	**0.79**	**0.79**	0.73	0.72
Precision	0.78	0.81	0.81	**0.91**	0.90
Recall	**0.79**	0.76	0.76	0.50	0.49
F1	0.78	0.78	**0.79**	0.65	0.63

Tabela 4.14: Resultados de teste para D3 usando diferentes gramas de palavras como características lexicais

Todas as experiências foram realizadas recorrendo ao SVM como algoritmo de aprendizagem e usando o algoritmo TF-IDF para a representação das características. O objetivo aqui será comparar os resultados da experiência onde se usam apenas unigramas (palavras únicas e seus lemas) com os resultados obtidos nas restantes quatro combinações, de modo a perceber se existe alguma melhoria.

Com a análise dos resultados podemos concluir que a inclusão dos unigramas é fundamental sendo que não existe uma grande variância quando, a estes, juntamos bigramas e até trigramas. Para além de existirem menos bigramas e trigramas de palavras do que unigramas, o facto de considerarmos um valor de frequência mínimo de 2 faz com que, no fim, tenhamos ainda menos exemplos de bigramas e trigramas considerados. Alguns deles serão, claramente, úteis para o modelo, tanto que os resultados nunca pioram com a sua inclusão, ou ficam iguais ou melhoram. No entanto, não têm uma grande relevância para os conjuntos de dados que utilizámos. Provavelmente, em conjuntos de dados maiores, a adição de bigramas e trigramas de palavras possa ter uma maior importância.

Para o conjunto de dados D1, os valores de *accuracy* e de F1 aumentam 1 ponto quando se adicionam os bigramas à combinação inicial de unigramas. O mesmo acontece para D3 onde existe também um aumento de 1 ponto nos mesmos valores, mas neste caso acontece aquando do uso da combinação entre unigramas, bigramas e trigramas.

Devido aos resultados obtidos e, de modo a termos uma maior diversidade de características, a combinação entre unigramas, bigramas e trigramas será a escolhida para os testes ao modelo nas experiências seguintes.

4.3.3 Análise de Relevância das Caraterísticas Lexicais

Na construção do nosso modelo uma das principais preocupações seria diminuir ao máximo o chamado *overfitting*. Este fenómeno ocorre quando um modelo não consegue generalizar a partir dos dados de treino para dados que nunca viu. Muitas vezes conseguem-se até resultados muito bons para os dados de treino, no entanto quando se testa em outros dados os resultados baixam consideravelmente. Para ajudar a combater o *overfitting* e, ao mesmo tempo, diminuir a complexidade do modelo, pode ser feita uma seleção das características mais relevantes. O objetivo seria que o modelo não considerasse características que, apesar de aparecerem algumas vezes nos dados de treino, no cômputo geral do problema não seriam importantes. Ao mesmo tempo, este teste permitirá ajudar a compreender o conteúdo dos conjuntos de dados e em que caraterísticas os classificadores terão mais tendência em se apoiar.

Para se perceber que características deveriam ser usadas foi feito um teste do χ^2. Quanto maior for o valor obtido para cada característica, maior será a relevância que ela terá para a classificação final. A tabela 4.15 mostra as 10 características que obtiveram os maiores valores no teste do χ^2, para os três conjuntos de dados usados:

D1		**D2**		**D3**	
um	16.80	quem	37.21	quem	25.62
porque que	15.82	não	21.74	porque	23.45
porque	15.78	porque	20.52	que que	14.72
que que	14.86	marcelo	18.84	porque que	14.51
porque que os	14.19	sócrates	17.91	sabem	13.54
sabem	14.07	bruno	17.89	porque que os	13.48
que um	13.03	de carvalho	16.08	que um	12.81
alentejanos	12.77	bruno de	15.30	os alentejanos	12.32
os alentejanos	12.61	bruno de carvalho	15.30	alentejanos	11.68
quem	12.58	costa	15.23	um	11.65

Tabela 4.15: Características com maior valor no teste do χ^2 para os três conjuntos de dados

Os exemplos humorísticos presentes no conjunto de dados D1 são piadas curtas em que muitas delas contêm pronomes ou advérbios interrogativos. Devido a esse facto, acaba por

não surpreender a presença dos unigramas "sabem", "porque" ou "quem". A presença da palavra "um" em primeiro lugar acaba por ser uma surpresa já que é considerada uma *stopword* da língua portuguesa no entanto, em D1, aparece em menos de 75% dos documentos, valor este que foi definido como sendo a frequência máxima possível.

Uma estrutura muito comum também dentro das piadas curtas é começar a frase por "Porque é que..." o que faz com que o bigrama "porque que" e o trigrama "porque que os" obtenham também um elevado valor no teste do qui-quadrado. Devem mencionar-se ainda as características "alentejanos" e "os alentejanos" devido ao elevado número de piadas sobre alentejanos, o que faz com que a presença desta palavras esteja associada ao humor. A título de curiosidade, outras palavras muito associadas ao humor mas que não estão nas com maior χ^2 em D1 são, "loira" (10.82), "elefante"(9.10), "lâmpada" (9.04) ou "cúmulo"(8.27).

Em relação a D2, voltam a aparecer as características "quem" e "porque" com elevados valores. Visto que este conjunto de dados é constituído por notícias, tanto humorísticas como não humorísticas, além de alguns provérbios, é natural que muitas das características relevantes estejam associadas a entidades, neste caso nomes de pessoas. Daí a presença das características "marcelo", "sócrates", "costa" e "bruno de carvalho", nomes que foram bastante mencionados nos títulos de notícia à data da recolha.

Por último, D3 é composto por exemplos de todas as fontes, sendo que as características mais relevantes foram muito parecidas às de D1.

4.3.4 Seleção de Características

A seleção de características foi feita, como já referido, com recurso ao teste de χ^2. Para se fazer o teste do χ^2 foi usada a função *SelectKBest()* do *scikit-learn*. Esta função recebia dois parâmetros, a função a partir da qual se calculavam as melhores características, que no nosso caso foi a função do χ^2, e ainda o número de características a selecionar. As características mais relevantes seriam as que obtivessem um valor mais alto no teste de χ^2, sendo descartadas aquelas que tivessem valores mais baixos. Quanto maior fosse o valor obtido no teste, maior seria a dependência entre a característica e a classe — no nosso caso, humor ou não humor — logo, teria uma maior relevância para o nosso problema.

O próximo passo seria definir um número máximo de características a considerar. Olhando para os conjuntos de dados usados, verifica-se que o menor de todos é D1 que contém 1400 instâncias no total. Foi então definido que o número máximo de características a considerar seriam 1400. Para se chegar ao valor a usar recorreu-se ao método *Grid Search*, que serve para encontrar o pârametro ótimo para o modelo (neste caso, o número de características) que resulte em melhores resultados. Definiu-se então um conjunto de valores que iam desde apenas 200 características a 1400, em saltos de 200. O melhor valor, otimizado para garantir a melhor F1, foi calculado com base nas instâncias de treino do conjunto de dados em questão e posteriormente testado nos dados para teste. Este processo repetia-se três vezes para cada conjunto de dados de modo a que se gerassem três valores candidatos, visto que o melhor número de características para os dados de treino podia não ser o melhor para os dados de teste.

Como foi anteriormente referido, está a usar-se o SVM como algoritmo de aprendizagem e o TF-IDF para a representação das características. As características são constituídas por unigramas (e seus lemas), bigramas e trigramas de palavras. As tabelas 4.16, 4.17 e 4.18 apresentam os resultados das experiências para os três números de características que obtiveram os melhores resultados no conjunto de treino (por ordem), para os diferentes

conjuntos de dados.

	Treino			Teste		
	1200	**1400**	**1000**	**1200**	**1400**	**1000**
Accuracy	0.96 ± 0.02	0.96 ± 0.02	0.96 ± 0.02	0.86	0.86	**0.87**
Precision	0.97 ± 0.02	0.97 ± 0.03	0.97 ± 0.03	0.88	0.88	**0.89**
Recall	$\mathbf{0.96 \pm 0.02}$	$\mathbf{0.96 \pm 0.03}$	0.94 ± 0.03	**0.84**	0.82	**0.84**
F1	0.96 ± 0.02	0.96 ± 0.02	0.96 ± 0.02	0.86	0.85	**0.87**

Tabela 4.16: Resultados de treino e teste para D1 para diferentes números de características consideradas

	Treino			Teste		
	1400	**1200**	**1000**	**1400**	**1200**	**1000**
Accuracy	0.88 ± 0.02	0.88 ± 0.02	0.88 ± 0.02	0.82	0.82	0.82
Precision	0.91 ± 0.02	0.91 ± 0.03	0.91 ± 0.02	0.80	**0.81**	**0.81**
Recall	$\mathbf{0.85 \pm 0.02}$	$\mathbf{0.85 \pm 0.02}$	0.83 ± 0.02	**0.85**	0.84	0.83
F1	$\mathbf{0.88 \pm 0.02}$	$\mathbf{0.88 \pm 0.02}$	0.87 ± 0.02	0.82	0.82	0.82

Tabela 4.17: Resultados de treino e teste para D2 para diferentes números de características consideradas

	Treino			Teste		
	1400	**1000**	**1200**	**1400**	**1000**	**1200**
Accuracy	$\mathbf{0.90 \pm 0.02}$	0.89 ± 0.03	0.89 ± 0.02	0.77	0.77	0.77
Precision	$\mathbf{0.96 \pm 0.02}$	$\mathbf{0.96 \pm 0.03}$	0.95 ± 0.02	0.82	**0.83**	0.82
Recall	$\mathbf{0.84 \pm 0.03}$	0.83 ± 0.04	0.83 ± 0.04	**0.70**	0.68	0.68
F1	$\mathbf{0.89 \pm 0.02}$	$\mathbf{0.89 \pm 0.03}$	0.87 ± 0.02	0.75	0.75	0.75

Tabela 4.18: Resultados de treino e teste para D3 para diferentes números de características consideradas

A tabela 4.16 mostra os resultados obtidos para o conjunto de dados D1. Os valores do número de características obtidos pelo método *Grid Search* foram 1200, 1400 e 1000, por esta ordem. Apesar de não existir uma diferença muito significativa, quando o modelo considera as 1000 características mais relevantes os valores de *accuracy* e de F1 atingem 87% sendo que foram os mais altos entre todas as experiências. Já para os conjuntos de dados D2 e D3 respetivamente nas tabelas 4.17 e 4.18, não existe diferença nos valores de *accuracy* e de F1. Quando comparados os resultados obtidos nesta experiência, para D2 e D3, com os obtidos nas outras experiências reparamos que baixaram um pouco, ao contrário do que acontece em D1. Apesar disso, o facto de selecionarmos apenas as caracterrísticas mais importantes, irá fazer com que o modelo seja mais generalizado e não considere palavras ou bigramas/trigramas de palavras muito específicos que estejam presentes apenas nos conjuntos de dados analisados e que não tenham nenhuma relevância para o problema.

4.4 Discussão

Neste capítulo abordámos o problema do reconhecimento de humor com base apenas em caraceterísticas lexicais. Depois de criados os conjuntos de dados necessários foram feitas diferentes experiências para avaliar o modelo produzido.

Na primeira experiência foram testadas diferentes formas de representação das características lexicais, por contagem ou recorrendo ao algoritmo TF-IDF, e diferentes algoritmos de aprendizagem. Começaram por se obter resultados bastantes positivos, com valores de *accuracy* de 85% para D1 e de 84% para D2, o que vem na linha do que foi conseguido em trabalhos relacionados para a língua inglesa. Em D3 os resultados não chegaram a valores tão altos, ainda assim obteve-se um máximo de 78% de *accuracy* e 79% de F1. Como já foi referido, estes resultados obtidos em D3 são facilmente justicáveis tendo em conta a diversidade de exemplos existente neste conjunto de dados.

A segunda experiência passou por avaliar se a inclusão de bigramas e/ou trigramas de palavras ajudava no nosso problema. Testaram-se diferentes combinações de n-gramas de palavras e, no fim, conclui-se que não existia uma grande diferença entre três das combinações testadas, sendo que as combinações que não incluíam unigramas tiveram muito piores resultados. Usar apenas unigramas, ou juntar bigramas ou ainda juntar bigramas e trigramas não influenciava a performance do modelo. Ainda assim, e como forma de aumentar a diversidade de características e também de generalizar o modelo criado, decidiu-se usar a combinação que incluía unigramas, bigramas e trigramas de palavras.

A última experiência tinha como objetivo selecionar um número de características relevantes para o problema de modo a combater o *overfitting*. Foi definido um limite máximo e um limite mínimo de características a considerar, 1400 e 200 respetivamente. Dentro destes valores foram testados números de características em saltos de 200, começando em 200 e terminando em 1400. Para cada conjunto de dados foram selecionados os três melhores valores com base nos dados de treino e, posteriormente, foram feitos testes nos dados de teste. Apesar de em D2 e D3 os resultados baixarem um pouco em relação às experiências anteriores, em D1 consegui-se atingir-se um máximo de 87% para *accuracy* e F1, usando as 1000 melhores características.

Posto isto, e depois de todas as experiências realizadas, conseguimos chegar a valores de pârametros que tornam o modelo o mais generalizado possível e que garantem bons resultados:

- **Algoritmo de Aprendizagem**: SVM
- **Representação de Características**: TF-IDF
- **Características usadas**: Unigramas, Bigramas e Trigramas
- **Número de Características a considerar**: 1000

Capítulo 5

Exploração de Características Específicas do Humor

No capítulo anterior o problema do reconhecimento de humor foi tratado como sendo um problema tradicional de classificação de texto, onde as características consideradas não são mais do que as palavras que constituem os diversos documentos. No entanto, o humor pode estar expresso numa frase através de características que não só as palavras a um nível superficial. Com base no trabalho relacionado acerca do reconhecimento de humor para outras línguas, foram identificadas várias características que podiam ser úteis para o nosso problema.

Neste capítulo serão enumeradas todas as características específicas do humor que foram extraídas assim como os recursos linguísticos a que recorremos para essa extração. Serão descritas inicialmente as experiências feitas com recurso às características específicas do humor. De seguida, descrevem-se experiências onde se procurou tirar partido de caraterísticas dos dois tipos: lexicais e específicas de humor. Finalmente, são ainda feitas pequenas experiências na classificação de adivinhas potencialmente humorísticas, que foram geradas automaticamente.

5.1 Características

A presença de determinadas características linguísticas numa frase pode estar fortemente associada à presença, ou não, de humor. Dentro do trabalho relacionado com o reconhecimento de humor são mencionados inúmeros exemplos de características que podem ser úteis para a resolução do problema. Visto que nenhum do trabalho relacionado estava direcionado para a língua portuguesa o objetivo seria identificar as características que fossem mais significantes e perceber que recursos linguísticos existentes poderiam ser usados de modo a auxiliar a extração. De seguida são enumeradas as características consideradas para o trabalho. Para cada característica é explicado também o seu processo de extração.

- **Palavras fora do Vocabulário**: identificação de palavras que não fazem parte do vocabulário da língua portuguesa. O facto de, por vezes, existirem palavras que são "inventadas" de modo a que uma frase tenha uma conotação humorística foi a razão para a extração desta característica

  Para a extração recorreu-se a um repositório que contém vetores de palavras, ou *word embeddings* em inglês, pré-treinados numa grande coleção de textos em português

e com diferentes algoritmos (Hartmann et al., 2017)[1]. O modelo utilizado foi um *Word2Vec* CBOW (*Continuous Bag of Words*) com vetores com dimensão de 300. A característica será representada pelo número de palavras de uma frase, que não estejam representadas no modelo. Consideremos a seguinte frase:

O que é um byfe? São oito bifes.

Neste caso a característica assumia um valor de 1, visto apenas existir uma palavra fora do vocabulário ("byfe"). A presença desta palavra pode remeter para um significado humorístico o que, neste caso, acaba por se confirmar.

- **Incongruência** (2 características): a existência de incongruência numa frase é muitas vezes associada à presença de humor. A presença de ideias opostas numa frase é considerado um dos principais fatores para definir se uma frase é humorística. Fazia então todo o sentido tentar captar esta característica.

  Com base no modelo já referido baseado em vetores de palavras foi calculada a similaridade entre palavras com o objetivo de capturar um valor associado à incongruência. No *word2vec*, cada palavra é representada por um vetor denso de números, calculado com base numa grande coleção de textos. Assim, o valor de similaridade de um par de palavras corresponde ao cosseno dos vetores dessas palavras no espaço vetorial do *word2vec*, ou seja, pode variar entre -1 (menos similar) até 1 (mais similar). Visto que o valor da característica não pode ser negativo para que se consiga calcular o valor do teste de χ^2, foi sempre somada uma unidade ao valor de similaridade retornado pelo modelo, variando agora entre 0 e 2. Foram então calculadas duas caraterísticas:

  #1 : A primeira característica é o valor médio de similaridade entre todos os pares de palavras no texto.

  #2 : A segunda característica é o valor de similaridade mais baixo para um par de palavras no texto.

  Quanto mais baixo for o valor obtido para ambas as características maior o indicador de incongruência, por se estarem a usar palavras que, normalmente, não são usadas em conjunto. Consideremos os seguintes títulos de notícia retirados do conjunto de dados D2 em que o primeiro é humorístico e o segundo é não humorístico:

  1. *Cavaco queixa-se: imposto sobre o açúcar é plano maquiavélico para impedi-lo de comer bolo-rei.*
  2. *Escola de 9,5 milhões em Guimarães deve abrir em Setembro.*

  Recorrendo ao modelo mencionado, para a primeira característica, obtiveram valores de 0.98 para o primeiro exemplo e 1.09 para o segundo exemplo reforçando a ideia de que os textos humorísticos terão valor médios de incongruência mais baixos. Já para a segunda característica e relativamente ao primeiro exemplo o valor de similaridade mais baixo foi de 0.60 para o par de palavras "impedi-lo" e "de". Para o segundo exemplo foi obtido 0.82 para o par "milhões" e "de". Ainda que os pares de palavras obtidos não tenham muito significado no contexto do nosso problema, voltamos a ter um valor mais baixo no exemplo humorístico do que no exemplo não humorístico.

[1] `http://nilc.icmc.usp.br/embeddings` último acesso em Agosto de 2019

- **Polaridade** (3 características): de forma a considerar o sentimento no reconhecimento de humor, calculou-se a polaridade tipicamente transmitida pelas palavras usadas, que pode ser positiva ou negativa. A presença de palavras (que podem ser adjetivos, nomes ou mesmo verbos) com conotação negativa em frases está muitas vezes associada à presença de humor (Mihalcea e Pulman, 2007) pelo que poderá ser interessante explorar também esta característica.

  Para a extração desta característica recorreu-se ao SentiLex-PT(Carvalho e Silva, 2015), um léxico de polaridades para português. Foram calculadas três características com base na polaridade das palavras:

  #1 : número de palavras com polaridade positiva.

  #2 : número de palavras com polaridade negativa.

  #3 : diferença entre as duas características anteriores. Caso a diferença seja negativa (mais palavras com polaridade negativa do que com polaridade positiva) a característica toma o valor de 0. Caso a diferença seja nula, a característica fica com o valor de 1. Finalmente, se a diferença for positiva, a característica terá o valor de 2.

  A título de exemplo vamos considerar a seguinte frase, presente num dos conjuntos de dados:

  > *Sabes qual é a diferença entre um acidente e um desastre? A minha sogra cai a um poço. É um acidente. Foi lá um bombeiro e salvou-a. É um desastre.*

  Analisando o exemplo em cima, o valor da característica #1 seria 1 visto que apenas a palavra "salvou-a" tem polaridade positiva de acordo com o léxico utilizado. Já em relação à característica #2 são identificadas três palavras com polaridade negativa, "desastre" (por duas vezes) e "cai", tomando um valor de 3. Uma vez que excistem três palavras com polaridade negativa e uma palavra com polaridade positiva a diferença é positiva, logo o valor da característica #3 será de 0.

- **Vocabulário Calão**: visto que o humor baseado em vocabulário calão é bastante popular, seria interessante considerar esta característica. O objetivo seria identificar o número de palavras que poderiam ser considerados como vocabulário calão, numa frase.

  Para a extração desta característica recorreu-se ao Dicionário Aberto de Calão e Expressões Idiomáticas[2]. Veja-se o seguinte exemplo, retirado de D1:

  > *Sabem o que o autoclismo diz à água? Vai à merda!*

  Se considerarmos a frase em cima, esta característica tomaria o valor de 1 já que a palavra "merda" pertence ao Dicionário de Calão.

- **Antonímia**: o humor está muitas vezes associado à presença de palavras com ideias contrárias numa frase, pelo que esta característica poderá ser relevante no contexto do nosso problema. A representação desta característica irá passar pela identificação do número de pares de palavras (na sua forma lematizada) que têm uma relação de antonímia.

  Para a extração desta característica recorreu-se a bases de conhecimento lexical para português (Gonçalo Oliveira, 2018) que incluam a relação de antonímia. Um exemplo disso mesmo está representado na seguinte frase:

[2] `https://natura.di.uminho.pt/~jj/pln/calao/calao.dic.txt` último acesso em Agosto de 2019

Qual é a melhor Universidade do mundo? A de Évora porque entram alentejanos e saem engenheiros.

As palavras "entrar" e "sair" (lemas das palavras "entram" e "saem"), que estão incluídas na frase, são um exemplo de um par de palavras que têm uma relação de antonímia. Neste caso, a característica teria o valor de 1.

- **Ambiguidade** (2 características): é muito frequente o humor tirar partido dos diferentes sentidos da mesma palavra, logo seria importante considerar a ambiguidade dos textos.

  De modo a calcular a ambiguidade de cada palavra, recorreu-se à OpenWordNet-PT (Paiva, Rademaker e Melo, 2012), onde lemas de palavras portuguesas estão associados aos seus diferentes sentidos. Dentro da ambiguidade foram definidas duas características a extrair:

  #1 : Média do número de sentidos de cada palavra usada, na forma lematizada.

  #2 : Número de sentidos da palavra mais ambígua de uma frase. A palavra mais ambígua será, logicamente, a que tiver um maior número de sentidos associados.

Como exemplo para demonstrar o cálculo das duas características relacionadas com a ambiguidade vamos considerar o seguinte título de notícia:

Caça à multa vai bater ainda novos recordes quando as autocaravanas também passarem a pagar IMI.

O primeiro passo feito é a lematização da frase, visto que o recurso utilizado utiliza os lemas das palavras para identificar o número de sentidos. Recorrendo ao *toolkit* NLPyPort o resultado da lematização foi:

Caça a o multa ir bater ainda novo recorde quando o autocaravana também passar a pagar IMI.

A primeira característica assumiu um valor de 7. Este valor corresponde ao número médio de sentidos de cada palavra que constitui a frase. Na tabela 5.1 são mostrados os números de sentidos atribuídos às palavras da frase, sendo que as palavras que não aparecem na tabela têm todas apenas 1 sentido.

Palavra	**Nº de Sentidos**
bater	40
ir	23
passar	19
pagar	10
novo	7
ainda	5
caça	4
multa	2

Tabela 5.1: Número de sentidos atribuídos a cada palavra

A partir da tabela conseguimos também perceber que o valor da segunda característica será 40 uma vez que é o número de sentidos da palavra mais ambígua, que neste caso é "bater".

- **Aliteração** (4 características): a constante repetição dos mesmos sons em palavras da mesma frase está muitas vezes associada à presença de humor. Esta figura de estilo tem o nome de aliteração.

  Uma simplificação para captar a aliteração num texto passa por identificar a repetição de determinados caracteres ou palavras. Decidiu-se então calcular o maior número de ocorrências de diferentes n-gramas de caracteres recorrendo à função *ngrams()* do *Natural Language Toolkit* (NLTK). N-gramas de caracteres não são mais do que a sequência de n caracteres que aparecem numa frase sendo que nunca passam o limite das palavras. A característica é representada pelo número de ocorrências do unigrama, bigrama, trigrama e tetragrama numa frase daí existirem quatro características diferentes para representar a aliteração. Veja-se o seguinte exemplo:

  > *Quantos técnicos de som são precisos para mudar uma lâmpada? Um, dois. Um, dois.*

  Na tabela 5.2 estão representados os valores das quatro características correspondentes à aliteração, tendo em conta o exemplo em cima.

	Caracteres	**Nº Ocorrências**
Unigrama	"s"	8
Bigrama	"os" e "is"	3
Trigrama	"doi" e "ois"	2
Tetragrama	"dois"	2

Tabela 5.2: Número de ocorrências de n-gramas de caracteres

- **Entidades Mencionadas** (11 características): principalmente em perguntas e respostas de cultura geral, ou em títulos de notícia, são muitas vezes mencionadas entidades de vários tipos. Visto que os conjuntos de dados que foram construídos contêm tantos textos desse tipo, esta característica poderia ser-nos ser útil. As características extraídas foram:

  - As primeiras 10 características não são mais que o número de entidades mencionadas na frase, agrupadas pelas 10 diferentes categorias incluídas na coleção HAREM(Freitas et al., 2010): Abstração, Acontecimento, Coisa, Local, Obra, Organização, Pessoa, Tempo, Valor, Outro.
  - A última característica é o número total de entidades mencionadas identificadas.

  Considerando o seguinte título de uma notícia:

  > *Costa quer Portugal com "nova visão" a disputar fundos comunitários.*

  Se olharmos para o exemplo são encontradas duas entidades, uma do tipo "Pessoa" ("Costa") e a outra do tipo "Local" (a palavra "Portugal"). A última característica teria o valor 2 visto que o total de entidades encontradas foram apenas duas.

- **Imaginabilidade e Concretude**: a imaginabilidade corresponde à facilidade e rapidez com que conseguimos, a partir de uma determinada palavra, evocar uma imagem mental correspondente a essa palavra. Já a concretude representa o grau com que as palavras estão associadas a objetos, coisas ou lugares que podem ser experienciados pelos sentidos.

  A extração destas característica consistiu na média do valor de imaginabilidade e concretude das palavras de uma frase, recorrendo a uma base de dados(Soares et

al., 2017) que contém valores de imaginabilidade e concretude para 3800 palavras escritas em português. Consideremos o seguinte exemplo:

> *Rui Rio quer árbitro português de ténis que enfrentou Serena Williams como Procurador-geral da República.*

Apenas 3 palavras do exemplo foram encontradas na base de dados que foi utilizada sendo elas "árbitro" (6.05 como valor de imaginabilidade e 6.03 como valor de concretude), "português" (4.51 e 4.58) e "ténis" (6.02 e 6.05). A todas as palavras não encontradas na base de dados é atribuído um valor de 0 para ambas as características. Os valores finais das características foram então de 1.04 para a Imaginabilidade e 1.05 para a Concretude.

5.2 Experiências

Depois de todos os conjuntos de dados estarem representados através das características que consideramos relevantes para o humor, descritas anteriormente, passou-se à fase das experiências. Inicialmente, o modelo foi testado apenas com base nas características específicas do humor, recorrendo aos quatro diferentes algoritmos de aprendizagem já mencionados.

Depois, o modelo foi ainda testado com a junção das características lexicais com as características específicas do humor. O facto de se utilizar o conjunto de ambas as características fazia prever os melhores resultados obtidos pelo modelo para os diferentes conjuntos de dados.

5.2.1 Análise da Relevância das Características Específicas do Humor

Antes de passarmos para as experiências feitas ao modelo com base nas caraceterísticas específicas do humor, vamos primeiro analisar a sua relevância e tentar perceber que características são mais importantes nos diferentes conjuntos de dados. Da mesma forma do que foi feito com as características lexicais, foi também calculado o valor do teste do χ^2 neste caso, com base nas instâncias de treino de cada conjunto de dados. De seguida é apresentada a tabela para as 10 características mais relevantes para cada conjunto de dados:

D1		**D2**		**D3**	
Fora do Vocabulário	167.05	Aliteração (unigramas)	1573.33	Fora do Vocabulário	186.27
Total Entidades	161.01	Total Entidades	828.74	Entidade "Pessoa"	152.91
Aliteração (unigramas)	115.35	Entidade "Pessoa"	678.03	Ambiguidade (#2)	149.48
Polaridade (#1)	89.92	Ambiguidade (#2)	359.25	Polaridade (#1)	69.44
Ambiguidade (#2)	77.13	Aliteração (bigramas)	170.25	Total Entidades	56.27
Entidade "Local"	57.05	Entidade "Organização"	81.42	Aliteração (unigramas)	48.44
Entidade "Tempo"	40.69	Entidade "Local"	72.33	Entidade "Organização"	19.98
Aliteração (bigramas)	23.45	Imaginabilidade	46.82	Entidade "Obra"	16.11
Entidade "Organização"	23.27	Entidade "Valor"	45.83	Imaginabilidade	15.51
Polaridade (#3)	21.58	Concretude	41.49	Concretude	13.36

Tabela 5.3: Características Específicas do Humor com maior valor no teste do χ^2 para os três conjuntos de dados

Em D1 o número de palavras fora do vocabulário foi a característica que se mostrou mais importante para a classificação. Muitas das *one-liners* recolhidas contêm palavras que são "inventadas" com o propósito de tornar uma frase humorística, sendo este resultado

algo expectável. Destacou-se também o número total de entidades mencionadas numa frase, assim como o número de entidades do tipo "Local", "Tempo" e "Organização". Em D1 metade dos exemplos não humorísticos são perguntas e respostas de cultura geral, sendo que este tipo de instâncias contêm muitas vezes entidades de vários tipos. Mencionar ainda a aliteração, representada pelos unigramas e pelos bigramas de caracteres, que também se revelou importante para a classificação final. Foram também identificadas bastantes mais palavras com polaridade positiva nos exemplos não humorísticos e, não por tanta margem, mais palavras com polaridade negativa nos exemplos humorísticos, o que reforça a ideia de que é comum que as piadas sejam compostas por palavras com uma conotação mais negativa. Devido a este facto, a Polaridade, quando representada pelo número de palavras com polaridade positiva e pela diferença entre palavras com polaridade positiva e negativa, acabou por ser também essencial para o problema.

Já em D2, e como já era expectável, o número total de entidades assim como o número de entidades do tipo "Pessoa", "Local" e "Valor" aparecem nas dez características mais relevantes. O facto de, em títulos de notícias, serem bastantes vezes mencionadas entidades de vários tipos fez com que se obtivessem estes resultados. Uma das principais diferenças de D2 para D1 foi o facto de o número de palavras fora do vocabulário ter deixado de ser relevante sendo que em D1 foi a característica com maior valor no teste do χ^2. Este facto é facilmente justificável já que nos títulos de notícia presentes em D2 é muito raro que existam palavras desse género. Referir ainda a aliteração (unigramas e bigramas) e a ambiguidade (representada pelo valor de número de sentidos da palavra mais ambígua) que, como aconteceu em D1, voltaram a ser muito relevantes.

Finalmente, em D3, os resultados mostram uma junção de quase todas as características que apareceram nos dois anteriores conjuntos de dados, o que seria de esperar, porque D3 é composto por exemplos das fontes que constituem D1 e D2.

5.2.2 Reconhecimento de Humor com Base em Caraterísticas Específicas

Cada documento de cada conjunto de dados passou então a ser representado por 27 características, que corresponde ao número total daquelas que foram extraídas. Para cada conjunto de dados, o modelo foi treinado e testado usando os quatro algoritmos de aprendizagem já mencionados no capítulo anterior: SVM, *Naive Bayes*, *Decision Tree* e *Random Forest*. As tabelas 5.4, 5.5 e 5.6 apresentam os resultados das experiências:

	Treino				**Teste**			
	SVM	**NB**	**DT**	**RF**	**SVM**	**NB**	**DT**	**RF**
Accuracy	0.75 ± 0.03	0.66 ± 0.04	0.77 ± 0.04	$\mathbf{0.80 \pm 0.03}$	0.74	0.63	0.73	**0.80**
Precision	0.73 ± 0.03	0.61 ± 0.03	0.78 ± 0.05	$\mathbf{0.82 \pm 0.03}$	0.73	0.58	0.74	**0.83**
Recall	0.79 ± 0.09	$\mathbf{0.92 \pm 0.04}$	0.75 ± 0.04	0.77 ± 0.05	0.77	**0.91**	0.71	0.76
F1	0.76 ± 0.04	0.73 ± 0.02	0.76 ± 0.03	$\mathbf{0.79 \pm 0.04}$	0.75	0.71	0.73	**0.80**

Tabela 5.4: Resultados de treino e teste para D1 considerando apenas as características específicas do humor

	Treino				Teste			
	SVM	**NB**	**DT**	**RF**	**SVM**	**NB**	**DT**	**RF**
Accuracy	$\mathbf{0.83 \pm 0.02}$	0.77 ± 0.02	0.76 ± 0.02	0.80 ± 0.02	**0.82**	0.76	0.72	0.78
Precision	$\mathbf{0.82 \pm 0.02}$	0.78 ± 0.02	0.76 ± 0.03	0.81 ± 0.02	**0.83**	0.78	0.73	0.82
Recall	$\mathbf{0.83 \pm 0.03}$	0.76 ± 0.04	0.75 ± 0.03	0.77 ± 0.04	**0.79**	0.73	0.69	0.72
F1	$\mathbf{0.83 \pm 0.02}$	0.77 ± 0.03	0.76 ± 0.02	0.79 ± 0.02	**0.81**	0.75	0.71	0.76

Tabela 5.5: Resultados de treino e teste para D2 considerando apenas as características específicas do humor

	Treino				Teste			
	SVM	**NB**	**DT**	**RF**	**SVM**	**NB**	**DT**	**RF**
Accuracy	0.69 ± 0.02	0.69 ± 0.04	0.69 ± 0.03	$\mathbf{0.73 \pm 0.02}$	0.60	0.63	0.63	**0.69**
Precision	0.74 ± 0.03	0.74 ± 0.05	0.68 ± 0.03	$\mathbf{0.77 \pm 0.03}$	0.64	0.68	0.65	**0.75**
Recall	0.60 ± 0.04	0.60 ± 0.06	$\mathbf{0.71 \pm 0.05}$	0.67 ± 0.04	0.45	0.49	**0.57**	0.56
F1	0.66 ± 0.03	0.66 ± 0.05	0.69 ± 0.03	$\mathbf{0.72 \pm 0.02}$	0.53	0.56	0.61	**0.64**

Tabela 5.6: Resultados de treino e teste para D3 considerando apenas as características específicas do humor

Olhando para os resultados obtidos para o conjunto de dados D1 (tabela 5.4) reparamos que os resultados baixaram relativamente aos testes feitos apenas com as características lexicais. Ainda assim temos de considerar que o modelo está a ser representado apenas por 27 características, ao contrário do que acontece quando foi testado com características lexicais onde estava representado por mais de 1000. Podemos então afirmar que estes resultados confirmam que a maior parte das características específicas do humor que foram extraídas estão relacionadas com o humor ou, pelo menos, com o tipo de humor usado em D1. Já em relação aos algoritmos de aprendizagem, obtiveram todos resultados muito semelhantes sendo que a única surpresa foi o baixo resultado de *accuracy* obtido pelo *Naive Bayes* quando comparado com os restantes. O *Random Forest* conseguiu mesmo valores de 80% para a *accuracy* e para F1 ficando apenas a 7 pontos do máximo conseguido com características lexicais, o que é um resultado muito bom.

Já relativamente aos resultados para D2, quando foi utilizado o SVM como algoritmo de aprendizagem obtiveram-se resultados muito semelhantes aos conseguidos com as características lexicais. Conseguiram-se valores de 82% para *accuracy* e 81% de F1, que foram muito semelhantes aos alcançados depois da fase da seleção de características no capítulo anterior, sendo que o valor de *accuracy* manteve-se igual tendo o valor de F1 baixado apenas 1 ponto. Com estes resultados voltamos a comprovar a elevada relevância das características extraídas também no caso de D2. De resto, todos os outros algoritmos de aprendizagem apresentaram valores significativamente piores para este conjunto de dados ainda que, pelo facto de representarmos as frases por muito menos características, não tenham tido uma má performance.

Por último, como aconteceu no capítulo anterior, os resultados para D3 baixaram relativamente aos outros conjuntos de dados, o que era expectácvel. Aliado ao facto do modelo ter acesso a muito menos informação, visto que estamos a representar as instâncias apenas com 27 características, está também a maior diversidade de exemplos existentes em D3. Isto faz com que seja normal que em D3 se obtenham resultados um pouco mais baixos. Os valores máximos de *accuracy* e F1 foram conseguidos com o *Random Forest* com 69% e 64% respetivamente, ficando na ordem dos 10 pontos abaixo dos máximos obtidos com características lexicais.

5.2.3 Análise da Relevância do Conjunto de Todas as Características

Depois de feitos os testes ao modelo apenas com as características específicas do humor que foram extraídas, o próximo passo seria perceber qual o impacto de combinar essas mesmas caraceterísticas com as características lexicais. Mais uma vez fazemos primeiro uma análise à relevância das características, neste caso ao conjunto das características lexicais com as características específicas do humor.

D1		**D2**		**D3**	
Fora do Vocabulário	167.05	Aliteração (unigramas)	1573.33	Fora do Vocabulário	186.27
Total Entidades	161.01	Total Entidades	828.74	Entidade "Pessoa"	152.91
Aliteração (unigramas)	115.35	Entidade "Pessoa"	678.03	Ambiguidade (#2)	149.48
Polaridade (#1)	84.92	Ambiguidade (#2)	359.25	Polaridade (#1)	69.44
Ambiguidade (#2)	77.13	Aliteração (bigramas)	170.25	Total Entidades	56.27
Entidade "Local"	57.05	Entidade "Organização"	81.42	Aliteração (unigramas)	48.44
Entidade "Tempo"	40.69	Entidade "Local"	72.33	quem	25.62
Aliteração (bigramas)	23.45	Imaginabilidade	46.82	porque	23.45
Entidade "Organização"	23.27	Entidade "Valor"	45.83	Entidade "Organização"	19.98
Polaridade (#3)	21.58	Concretude	41.49	Entidade "Obra"	16.11

Tabela 5.7: Características Específicas do Humor com maior valor no teste do qui-quadrado para os três conjuntos de dados

Olhando para a tabela 5.7, reparamos que nos conjuntos de dados D1 e D2 as 10 características com mais relevância são exatamente as mesmas aquando do teste apenas com as específicas do humor.

Em D1 a palavra "um", que obteve um valor de 16.80 no teste do χ^2, é a primeira característica lexical a aparecer, sendo que ainda fica atrás do número de entidades do tipo "Obra" (20.43), além das já representadas na tabela. Já em D2, "quem" é a primeira característica lexical a surgir (37.21) ficando atrás da Concretude (41.49).

Em D3 apenas as palavras "quem" e "porque" aparecem representadas na tabela, existindo mais uma vez uma dominância por parte das características específicas do humor. De uma maneira geral podemos afirmar que as características específicas do humor têm maior preponderância no contexto do nosso problema, relativamente às características lexicais. No entanto, como vimos pelos resultados obtidos, isto não significa que estas caraterísticas levem a melhores resultados. Isto poderá dever-se ao facto do número de caraterísticas lexicais ser muito superior. Por outro lado, sugere que pode haver um impacto positivo ao explorar ambos os tipos de caraterísticas.

5.2.4 Características Lexicais e Específicas do Humor

Em relação ao ambiente experimental, e como já tinha sido referido no capítulo anterior, chegou-se a valores de pârametros que resultariam num modelo genérico que nos garantia bons resultados. Para as seguintes experiências serão então usados os algoritmos SVM e TF-IDF como algoritmo de aprendizagem e algoritmo para a representação das características lexicais, respetivamente. Visto que o *Random Forest* foi o algoritmo que, no geral, apresentou melhores resultados nos testes feitos apenas com características do humor e já que estas características revelaram ser mais relevantes, será também usado nesta experiência. Serão consideradas 1000 características lexicais que irão ser compostas por unigramas, bigramas e trigramas de palavras. A estas características lexicais juntam-se as 27 características específicas do humor. Cada documento dos conjuntos de dados está representado por 1027 características. As tabelas 5.8, 5.9 e 5.10 apresentam o desempenho do classificador nestas experiências.

	Treino		Teste	
	SVM	**RF**	**SVM**	**RF**
Accuracy	$\mathbf{0.95 \pm 0.02}$	0.90 ± 0.03	**0.88**	0.86
Precision	$\mathbf{0.96 \pm 0.03}$	0.94 ± 0.03	0.88	**0.90**
Recall	$\mathbf{0.94 \pm 0.04}$	0.85 ± 0.05	**0.88**	0.80
F1	$\mathbf{0.95 \pm 0.02}$	0.90 ± 0.03	**0.88**	0.85

Tabela 5.8: Resultados de treino e teste para D1 com a combinação das características lexicais e específicas do humor

	Treino		Teste	
	SVM	**RF**	**SVM**	**RF**
Accuracy	$\mathbf{0.89 \pm 0.01}$	0.82 ± 0.01	**0.88**	0.77
Precision	$\mathbf{0.89 \pm 0.02}$	0.87 ± 0.01	0.87	**0.88**
Recall	$\mathbf{0.89 \pm 0.02}$	0.75 ± 0.03	**0.88**	0.63
F1	$\mathbf{0.89 \pm 0.01}$	0.80 ± 0.02	**0.88**	0.73

Tabela 5.9: Resultados de treino e teste para D2 com a combinação das características lexicais e específicas do humor

	Treino		Teste	
	SVM	**RF**	**SVM**	**RF**
Accuracy	$\mathbf{0.89 \pm 0.02}$	0.81 ± 0.02	0.79	**0.80**
Precision	$\mathbf{0.92 \pm 0.02}$	0.88 ± 0.03	0.83	**0.91**
Recall	$\mathbf{0.86 \pm 0.03}$	0.72 ± 0.04	**0.73**	0.66
F1	$\mathbf{0.89 \pm 0.02}$	0.80 ± 0.03	**0.78**	0.76

Tabela 5.10: Resultados de treino e teste para D3 com a combinação das características lexicais e específicas do humor

Com a combinação das características específicas do humor com as características lexicais esperava-se que o modelo atingisse os melhores resultados. O modelo não só estaria a tirar partido das 1000 melhores características lexicais mas também das 27 características específicas do humor que foram extraídas e que se revelaram muito importantes para o problema.

Apesar de não muito significativa, em D1 acabou mesmo por existir uma subida na performance quando foi usado o SVM como algoritmo de aprendizagem. Os valores de *accuracy* e de F1 aumentaram 1 ponto passando ambos de 87% para 88%. No caso do *Random Forest* os resultados ficaram ligeiramente abaixo, com 86% para a *accuracy* e 85% para o F1.

Em D2, obtiveram-se resultados bastantes diferentes para os dois algoritmos. Com o SVM existiu uma subida relativamente aos resultados obtidos nas experiências anteriores, com os valores de *accuracy* e de F1 a atingirem 88% que foi o máximo obtido para D2. Já com o *Random Forest* os resultados foram claramente piores com o modelo a atingir apenas 77% de *accuracy* e 73% de F1.

Por último, em D3 os dois algoritmos tiveram performances muito semelhantes. Os resultados alcançados foram, mais uma vez, melhores que nas experiências anteriores. O *Random Forest* conseguiu chegar a um máximo de *accuracy* com 80%, ficando pelos 76% para F1. Já o SVM ficou 1 ponto abaixo na *accuracy*, no entanto, atingiu o máximo de F1 com 78%.

Depois de concluídas as experiências com o conjunto de todas as características ficou bem patente a importância da inclusão de características específicas para o reconhecimento de humor. Já se tinham revelado mais relevantes anteriormente e acabaram por confirmar essa importância visto que, para todos os conjuntos de dados, houve uma subida na performance do modelo.

5.2.5 Testes em Adivinhas Geradas Automaticamente

Um dos objetivos inicias com o desenvolvimento de um modelo para o reconhecimento de humor seria a sua integração num sistema de geração automática de humor, o que permitiria ordenar o texto gerado, de acordo com o seu potencial humorístico. Foram então utilizadas 300 piadas geradas automaticamente por um modelo de geração de adivinhas em português (Gonçalo Oliveira e Rodrigues, 2018), sendo que todas as adivinhas geradas teriam, idealmente, algum potencial humorístico. Todas as adivinhas geradas foram criadas através de um conjunto de regras pelo que a sua estrutura é sempre muito semelhante, consistindo em perguntas e respostas. Dentro dos conjuntos de dados utilizados no nosso trabalho, as *One-Liners* são as instâncias mais parecidas às adivinhas geradas pelo sistema.

Todas as 300 adivinhas passaram por um processo de classificação manual, através de uma plataforma de *crowdsourcing*. Mais propriamente, colaboradores humanos classificaram cada adivinha de acordo com três aspectos: coerência, novidade e ainda potencial humorístico. O nosso objetivo aqui seria aplicar o modelo de reconhecimento de humor desenvolvido a estas adivinhas, analisando depois se existia alguma correlação entre o potencial humorístico da adivinha, atribuído pelos colaboradores, e a probabilidade da adivinha ser humorística, atribuída pelo nosso modelo. Visto que cada adivinha foi avaliada por três pessoas, era necessário usar um valor de potencial humorístico proveniente do conjunto dessas três classificações. Considerou-se então que o potencial humorístico de cada adivinha seria o valor da mediana das três avaliações. Na tabela 5.11 são apresentados alguns exemplos das instâncias que foram geradas.

Exemplo	**Potencial Humorístico**
Qual é o contrário de malbarato? bem-caro.	5
Qual é o contrário de enfrentarei? evitar-rainha.	4
Que resulta do cruzamento entre um local e habitual? lugar-comum.	3
O que significa condecorado? um soberano que é vermelho.	3
O que significa reipesado? um soberano que é lento.	2
Qual é o contrário de atingir? atinandar.	1

Tabela 5.11: Exemplos de adivinhas geradas pelo sistema e o seu valor de potencial humorístico

A primeira experiência passou por treinar o modelo nos três diferentes conjuntos de dados que usámos e, posteriormente, testar com recurso às adivinhas. O modelo iria retornar o valor da probabilidade que cada uma delas tinha de ser humorística. Recorrendo à correlação de Pearson, foi depois testado se entre esse valor de probabilidade e o valor de potencial humorístico de cada adivinha existia alguma correlação. Referir ainda que foi usado o modelo genérico a que chegámos no capítulo anterior, recorrendo ao conjunto de características lexicais e das específicas do humor. A tabela 5.12 mostra os resultados da experiência. Além dos valores da correlação de Pearson é também apresentado o número de piadas que foram classificadas como humorísticas pelo modelo.

	D1	D2	D3
Nº Humorísticas	286	5	209
Correlação de Pearson	0.10	0.07	0.16

Tabela 5.12: Valores da Correlação de Pearson e número de adivinhas classificadas como humorísticas para o conjunto das 300

Os resultados da primeira experiência mostram valores de correlação baixos, sendo que o melhor acaba por ser obtido quando treinamos o modelo no conjunto de dados D3 com 0.16.

Em D1, 286 adivinhas foram classificadas como humorísticas (95.33%) e o valor de correlação foi 0.10. Foi o conjunto de dados onde mais adivinhas foram classificadas como humorísticas, também pelo facto de ser o conjunto de dados em que os exemplos humorísticos mais se assemelhavam às adivinhas geradas automaticamente. Já em D2 apenas 5 das 300 adivinhas foram consideradas humorísticas e obteve-se um valor de correlação de 0.07. Este resultado acaba por não surpreender visto que a estrutura das instâncias contidas em D2 é extremamente diferente da estrutura das adivinhas. Por último, em D3 209 adivinhas (69.67%) foram classificadas como humorísticas e conseguiu-se o valor de correlação mais alto com 0.16. Ainda que este valor tenha sido o melhor, acaba por não revelar uma correlação muito forte.

De forma a perceber se a correlação obtida estava a ser prejudicada pela subjetividade das avaliações e consequente discordância entre anotadores, foi feita uma segunda experiência. Visto que avaliar se uma frase é, ou não, humorística pode ser uma tarefa bastante subjetiva onde nem toda a gente segue o mesmo critério, decidiu-se limitar o número de adivinhas a serem testadas. Pegando nas três diferentes classificações de cada adivinha, calculou-se o desvio-padrão para cada uma. Foi definido que só consideraríamos piadas em que o valor do desvio-padrão das três avaliações fosse menor que 0.5, de modo a considerar aquelas em que mais concordância houve acerca do valor de potencial humorístico. Das 300 adivinhas ficaram apenas 124. A 5.13 mostra os resultados obtidos.

	D1	D2	D3
Nº Humorísticas	119	1	88
Correlação de Pearson	0.19	0.03	0.20

Tabela 5.13: Valores da Correlação de Pearson e número de adivinhas classificadas como humorísticas para o conjunto das 124

Os resultados são bastante parecidos aos iniciais sendo que o valor de correlação aumentou quando se treinou em D1 e D3 e diminui quando se treinou em D2. Ainda que não sejam valores que indiquem uma correlação muito forte, os resultados obtidos para D1 e D3 com 0.19 e 0.20 respetivamente mostram uma correlação positiva e que dá a entender que existe potencial do modelo na classificação de piadas. Referir mais uma vez que nestas experiências estamos também sujeitos a classificações de humanos, que podem não ter sido dadas com muito critério além de que, classificar humor, é sempre uma tarefa extremamente subjetiva.

5.3 Discussão

Neste capítulo foi feita uma exploração de características específicas que fossem relevantes para o reconhecimento de humor. Com base em trabalho relacionado para a língua inglesa e considerando algumas características propostas no âmbito deste trabalho foram identificadas e extraídas 27 características específicas do humor. Foi dada uma explicação acerca de cada característica e de que forma poderia estar associada ao humor, assim como exemplificada a sua representação e as ferramentas que se utilizaram para a sua extração.

A análise da relevância das características extraídas, feita com recurso ao teste do χ^2, aliada às experiências feitas apenas com acesso a estas características, revelaram uma grande importância das mesmas para o problema do reconhecimento de humor. Foram obtidos resultados muito interessantes, ainda que piores que os obtidos no capítulo anterior, uma vez que o modelo tinha acesso a muito menos informação nesta fase.

A grande importância das características extraídas ficou ainda mais vincada aquando da análise de relevância das características quando, às específicas, se juntaram as lexicais. Para todos os conjuntos de dados usados existiu uma clara superioridade das características específicas do humor.

Foi ainda feita a experiência de juntar todas as características e testar novamente o modelo. Aqui, como era esperado, foram obtidos os melhores resultados havendo um aumento na performance em todos os conjuntos de dados. Para D1 e D2 o modelo chegou a valores muito parecidos aos relatados em trabalho relacionado para inglês. Em D3, apesar dos valores obtidos serem mais baixos, acabou por existir uma boa performance tendo em conta a variade de exemplos existentes.

Por fim, foram feitas também experiências na classificação de adivinhas potencialmente humorísticas geradas automaticamente. O objetivo foi avaliar o nível de correlação que existia entre o valor de probabilidade que uma adivnha tinha de ser humorística (atribuído pelo nosso modelo) e o valor de potencial humorístico da mesma adivinha. Apesar de não termos obtido valores da correlação de Pearson que nos indiquem que exista uma correlação muito forte, conseguimos um valor positivo que nos revela algum potencial na classificação de piadas por parte do modelo.

Capítulo 6

Conclusão

Neste capítulo são apresentadas as principais conclusões relativas ao trabalho que foi desenvolvido. Serão mencionadas ainda as contribuições que resultaram do trabalho assim como discutido o trabalho futuro que poderá ser feito no âmbito do reconhecimento de humor em português.

O objetivo principal desta dissertação era o desenvolvimento de um modelo computacional que conseguisse reconhecer humor, algo que para a língua portuguesa nunca tinha sido feito. Esse objetivo foi conseguido na medida em que se chegou a um modelo computacional que pode ser usado para reconhecer humor escrito em português, ainda que com um desempenho variável, dependendo dos dados usados para treino e das caraterísticas usadas. Durante todo o processo foram testados diferentes algoritmos de classificação automática de texto assim como exploradas um grande número de características.

A fase inicial passou pela construção de conjuntos de dados que nos permitissem treinar e testar os modelos criados. Cada conjunto de dados continha instâncias humorísticas e não humorísticas retiradas de diferentes fontes. Relativamente aos exemplos humorísticos, foram recolhidas frases relativas a diferentes tipos de humor: *one-liners* e títulos humorísticos de notícias. Já em relação aos exemplos não humorísticos, optou-se por perguntas e respostas de cultura geral, títulos de notícia ou provérbios. Existiu sempre o cuidado de recolher exemplos positivos e negativos muito semelhantes, para que os resultados obtidos pelo modelo avaliassem realmente se existia, ou não, humor de modo a que não se baseasse em características irrelevantes para o problema.

Depois dos conjuntos de dados estarem criados foram feitas as primeiras experiências ao modelo. Inicialmente foram usadas apenas características lexicais. Os primeira testes consistiram na comparação da performance entre diferentes algoritmos de aprendizagem (SVM, *Naive Bayes*, *Decision Tree* e *Random Forest*) assim como diferentes representações das características lexicais (Contagem ou TF-IDF). Foram ainda testadas diferentes combinações de n-gramas de tokens como características lexicais, onde se definiram diferentes combinações (entre unigramas, bigramas e trigramas). Foi também feita uma seleção de características de modo a combater o *overfitting* do modelo. Finalmente chegou-se a um modelo genérico que garantia bons resultados no reconhecimento de humor onde se usou o SVM como algoritmo de aprendizagem, TF-IDF como representação das características onde se usaram apenas as 1000 características lexicais mais relevantes. Estas características lexicais eram constituídas não só por unigramas de tokens, mas também por bigramas e trigramas. Com este modelo foram obtidos bons resultados, que estão na linha do que foi conseguido para a língua inglesa.

Seguidamente foram exploradas características específicas do humor. Foram extraídas 27 características específicas sendo que além de nos termos baseado no trabalho relacionado para a escolhas das mesmas, foram introduzidas características que não tinha sido usadas em nenhum modelo de reconhecimento de humor. Introduzimos então o número de palavras fora do vocabulário, a Imaginabilidade, a Concretude e ainda todas as características relacionadas com as Entidades Mencionadas. Fizeram-se experiências com base apenas nas características extraídas e, apesar da performance do modelo ter baixado um pouco, obtiveram-se bons resultados que comprovaram a relevância das características específicas do humor. Foi também feita a experiência com o conjunto das características lexicais e das específicas do humor onde se obtiveram os melhores resultados do modelo comprovando mais uma vez a importância da adição de características específicas. Por último foi feita uma experiência em que o modelo classficou adivinhas potencialmente humorísticas geradas automaticamente, onde se obtiveram valores de correlação promissores.

Concluindo, conseguimos chegar a um modelo computacional que consegue, com bons resultados, identificar diferentes estilos de humor. Foram ainda identificadas características específicas do humor e explicado todo o seu processo de extração e as ferramentas usadas, assim como criados diferentes conjuntos de dados o que pode ser bastante útil para quem quiser abordar o mesmo problema.

6.1 Contribuições

Como principais contribuições resultantes deste trabalho estão:

- A criação de diferentes conjuntos de dados com diferentes tipos de humor. Encontrar um grande número de exemplos de frases humorísticas escritas em português não é propriamente fácil (principalmente *One-Liners*). Ter conjuntos de dados criados e organizados em que cada um contém exemplos de instâncias humorísticas e não humorísticas pode ser muito útil para quem quiser abordar o mesmo problema.

- Foram identificadas características que podem estar associadas à presença de humor com base em trabalho relacionado para a língua inglesa. Foram também introduzidas mais algumas características que podiam também ser relevantes para o problema, como o número de palavras fora do vocabulário, a Imaginabilidade ou a Concretude. Para todas elas foi explicado o processo de extração assim como todas as ferramentas usadas nesse processo. Foi também explicada a maneira como cada uma estava representada.

- No fim do trabalho conseguimos chegar a um modelo que, como comprovam os resultados, pode ser usado para reconhecer humor escrito em português baseado em diferentes características, este que era o objetivo principal da dissertação.

- Foi ainda escrito e publicado um artigo científico onde estão documentadas algumas das experiências que se realizaram durante o desenvolvimento do trabalho.

6.2 Trabalho Futuro

Os resultados obtidos com o modelo desenvolvido são promissores e mostram que é possível, através de diferentes características, conseguir reconhecer humor em português. No en-

tanto, ainda existe trabalho a fazer para que se consiga chegar a um sistema que garanta ainda melhores resultados e que seja mais genérico.

A recolha de mais dados será uma parte fundamental. Apesar de termos uma quantidade aceitável de dados, o treino de modelos com um maior número de exemplos (tanto positivos como negativos) permitiria a aplicação de classificadores mais recentes, por exemplo, baseados em redes neuronais profundas, e, possivelmente, obter ainda melhores resultados. Além disso, e caso se consiga ter mais dados com mais tipos de humor, poderá também ser possível, não só identificar uma frase humorística, mas também agrupar frases por tipos de humor, ou seja, treinar a classificação multinomial em vez de apenas binária ou binomial.

A exploração de mais características específicas pode também ser muito útil para o problema. Durante este trabalho identificámos e introduzimos algumas características que nos poderiam indicar a presença de humor em frases que acabaram por se revelar muito importantes. A identificação de mais características específicas ou até diferentes representações das características que introduzimos pode também ajudar na performance de modelos deste tipo.

Poderá ainda ser feita a integração do modelo implementado em agentes conversacionais de modo a que estes consigam identificar interações humorísticas, num filtro de publicações ou mesmo num sistema de geração de humor.

Bibliografia

Agarwal, Apoorv, Boyi Xie, Ilia Vovsha, Owen Rambow e Rebecca Passonneau (2011). "Sentiment analysis of twitter data". Em: *Proceedings of the workshop on languages in social media*. Association for Computational Linguistics, pp. 30–38.

Aggarwal, Charu C e ChengXiang Zhai (2012). "A survey of text classification algorithms". Em: *Mining text data*. Springer, pp. 163–222.

Barbieri, Francesco e Horacio Saggion (2014). "Automatic Detection of Irony and Humour in Twitter." Em: *Proceedings of 5th International Conference on Computational Creativity (ICCC)*, pp. 155–162.

Binsted, Kim, Anton Nijholt, Oliviero Stock, Carlo Strapparava, G Ritchie, R Manurung, H Pain, Annalu Waller e D O'Mara (2006). "Computational humor". Em: *IEEE Intelligent Systems* 21.2, pp. 59–69.

Burnard, Lou (1995). "Users Reference Guide British National Corpus Version 1.0". Em:

Buschmeier, Konstantin, Philipp Cimiano e Roman Klinger (2014). "An impact analysis of features in a classification approach to irony detection in product reviews". Em: *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 42–49.

Carvalho, Paula e Mário J. Silva (2015). "Sentilex-PT: principais características e potencialidades". Em: *Linguística, Informática e Tradução: Mundos que se Cruzam*. Ed. por Alberto Simões, Anabela Barreiro, Diana Santos, Rui Sousa-Silva e Stella E. O. Tagnin. Vol. 7(1). OSLa: Oslo Studies in Language 1. University of Oslo, pp. 425–438.

Carvalho, Paula, Luís Sarmento, Mário J Silva e Eugénio De Oliveira (2009). "Clues for detecting irony in user-generated contents: oh...!! it's so easy;-". Em: *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*. ACM, pp. 53–56.

Clemêncio, André, Ana Alves e Hugo Gonçalo Oliveira (2019). "Recognizing Humor in Portuguese: First Steps". Em: *Proceedings of 19th EPIA Conference on Artificial Intelligence, EPIA 2019, Vila Real, Portugal, September 3-6, 2019, Part II*. Vol. 11805. LNCS/LNAI. Springer, pp. 744–756.

Dash, Manoranjan e Huan Liu (1997). "Feature selection for classification". Em: *Intelligent data analysis* 1.1-4, pp. 131–156.

Ferreira, João, Hugo Gonçalo Oliveira e Ricardo Rodrigues (2019). "Improving NLTK for Processing Portuguese". Em: *Symposium on Languages, Applications and Technologies (SLATE 2019)*, In press.

Freitas, Cláudia, Paula Carvalho, Hugo Gonçalo Oliveira, Cristina Mota e Diana Santos (2010). "Second HAREM: advancing the state of the art of named entity recognition in Portuguese". Em: *Proceedings of 7th International Conference on Language Resources and Evaluation*. LREC 2010. La Valleta, Malta: ELRA.

Freitas, Larissa A de, Aline A Vanin, Denise N Hogetop, Marco N Bochernitsan e Renata Vieira (2014). "Pathways for irony detection in tweets". Em: *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. ACM, pp. 628–633.

Gonçalo Oliveira, Hugo (2018). "A Survey on Portuguese Lexical Knowledge Bases: Contents, Comparison and Combination". Em: *Information* 9.2. ISSN: 2078-2489. DOI: 10.3390/info9020034. URL: https://doi.org/10.3390/info9020034.

Gonçalo Oliveira, Hugo e Ricardo Rodrigues (2018). "Explorando a Geração Automática de Adivinhas em Português". Em: *Linguamática* 10.1, pp. 3–18. URL: http://www.linguamatica.com/index.php/linguamatica/article/view/268.

Hartmann, Nathan, Erick Fonseca, Christopher Shulby, Marcos Treviso, Jéssica Silva e Sandra Aluísio (2017). "Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks". Em: *Proceedings of the 11th Brazilian Symposium in Information and Human Language Technology*. Uberlândia, Brazil: Sociedade Brasileira de Computação, pp. 122–131. URL: http://aclweb.org/anthology/W17-6615.

Houvardas, John e Efstathios Stamatatos (2006). "N-gram feature selection for authorship identification". Em: *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*. Springer, pp. 77–86.

Jindal, Nitin e Bing Liu (2007). "Review spam detection". Em: *Proceedings of the 16th international conference on World Wide Web*. ACM, pp. 1189–1190.

Joachims, Thorsten (1998). "Text categorization with support vector machines: Learning with many relevant features". Em: *European conference on machine learning*. Springer, pp. 137–142.

Jurafsky, Daniel e James H. Martin (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. 2nd. Prentice Hall series in artificial intelligence. Englewood Cliffs, NJ: Prentice Hall, Pearson Education International. ISBN: 0-13-504196-1 ; 978-0-13-504196-3.

Lewis, David D, Yiming Yang, Tony G Rose e Fan Li (2004). "Rcv1: A new benchmark collection for text categorization research". Em: *Journal of machine learning research* 5.Apr, pp. 361–397.

Magnini, Bernardo, Alessandro Vallin, Christelle Ayache, Gregor Erbach, Anselmo Peñas, Maarten de Rijke, Paulo Rocha, Kiril Ivanov Simov e Richard F. E. Sutcliffe (2004). "Overview of the CLEF 2004 Multilingual Question Answering Track". Em: *Multilingual Information Access for Text, Speech and Images, 5th Workshop of the Cross-Language Evaluation Forum (CLEF), Revised Selected Papers*. Vol. 3491. LNCS. Springer, pp. 371–391.

McCallum, Andrew, Kamal Nigam et al. (1998). "A comparison of event models for naive bayes text classification". Em: *AAAI-98 workshop on learning for text categorization*. Vol. 752. 1. Citeseer, pp. 41–48.

Mihalcea, Rada e Stephen Pulman (2007). "Characterizing humour: An exploration of features in humorous texts". Em: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, pp. 337–347.

Mihalcea, Rada e Carlo Strapparava (2005). "Making computers laugh: Investigations in automatic humor recognition". Em: *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pp. 531–538.

— (2006). "Learning to laugh (automatically): Computational models for humor recognition". Em: *Computational Intelligence* 22.2, pp. 126–142.

Miller, George A (1995). "WordNet: a lexical database for English". Em: *Communications of the ACM* 38.11, pp. 39–41.

Paiva, Valeria, Alexandre Rademaker e Gerard Melo (2012). "OpenWordNet-PT: An Open Brazilian WordNet for Reasoning". Em: *Proceedings of 24th International Conference on Computational Linguistics*. COLING (Demo Paper).

Pang, Bo e Lillian Lee (2004). "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts". Em: *Proceedings of the 42nd annual*

meeting on Association for Computational Linguistics. Association for Computational Linguistics, p. 271.

Raskin, Victor (2008). *The primer of humor research*. Vol. 8. Walter de Gruyter.

Reyes, Antonio, Paolo Rosso e Davide Buscaldi (2009). "Humor in the blogosphere: First clues for a verbal humor taxonomy". Em: *Journal of Intelligent Systems* 18.4, pp. 311–332.

— (2012). "From humor recognition to irony detection: The figurative language of social media". Em: *Data & Knowledge Engineering* 74, pp. 1–12.

Ruch, Willibald (2002). "Computers with a personality? lessons to be learned from studies of the psychology of humor". Em: *Proceeding of The April Fools Day Workshop on Computational Humor*, pp. 57–70.

Salton, Gerard e Christopher Buckley (1988). "Term-weighting Approaches in Automatic Text Retrieval". Em: *Inf. Process. Manage.* 24.5, pp. 513–523. ISSN: 0306-4573. DOI: 10.1016/0306-4573(88)90021-0. URL: http://dx.doi.org/10.1016/0306-4573(88)90021-0.

Singh, Push et al. (2002). "The public acquisition of commonsense knowledge". Em: *Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*.

Sjöbergh, Jonas e Kenji Araki (2007). "Recognizing humor without recognizing meaning". Em: *International Workshop on Fuzzy Logic and Applications*. Springer, pp. 469–476.

Soares, Ana Paula, Ana Santos Costa, João Machado, Montserrat Comesaña e Helena Mendes Oliveira (2017). "The Minho Word Pool: Norms for imageability, concreteness, and subjective frequency for 3,800 Portuguese words". Em: *Behavior Research Methods* 49.3, pp. 1065–1081.

Song, Ge, Yunming Ye, Xiaolin Du, Xiaohui Huang e Shifu Bie (2014). "Short text classification: A survey". Em: *Journal of Multimedia* 9.5, p. 635.

Tagnin, Stella EO (2005). "O humor como quebra da convencionalidade". Em: *Revista brasileira de linguística aplicada* 5.1, pp. 247–257.

Turing, Alan (1950). "Computing intelligence and machinery". Em: *Mind* 59.2236, pp. 433–460.

Yang, Diyi, Alon Lavie, Chris Dyer e Eduard Hovy (2015). "Humor recognition and humor anchor extraction". Em: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2367–2376.

Yang, Yiming e Jan O Pedersen (1997). "A comparative study on feature selection in text categorization". Em: *Icml*. Vol. 97. 412-420, p. 35.