Instituto Superior de Engenharia do Porto

Classificação e agregação automática de notícias desportivas

André Pinho de Almeida

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática, Área de Especialização em Arquiteturas, Sistemas e Redes

Orientador: Dr. Nuno Escudeiro

Júri:

Presidente:

Doutora Maria de Fátima Coutinho Rodrigues

Vogais:

Doutor Fernando Jorge Ferreira Duarte

Doutor Nuno Filipe Fonseca Vasconcelos Escudeiro

Porto, Outubro 2014

«Em memória do avô Alcides»

Resumo

Este relatório foi elaborado no âmbito da dissertação para obtenção do Grau de Mestre em Engenharia Informática do Instituto Superior de Engenharia do Porto

Foi desenvolvido com vista o auxílio da implementação de um módulo de classificação e agregação (*clustering*) automática de notícias desportivas. Este módulo será implementado numa aplicação web relacionada com o desporto a ser desenvolvida futuramente.

O principal objetivo do trabalho desenvolvido é perceber entre inúmeras possibilidades existentes para classificação e *clustering* de documentos quais as que melhor se adequam face às exigências necessárias. Aqueles que apresentaram melhores resultados foram os escolhidos para a fase de implementação do módulo de classificação e *clustering* de notícias.

Em primeiro lugar foi realizado um levantamento do estado da arte de forma a se ter conhecimento de todas as possibilidades existentes. Face a essas possibilidades, foram selecionados dois algoritmos para cada um dos temas a abordar. Os algoritmos escolhidos foram aquelas que se verificaram os mais adequados.

Para a classificação foram selecionados o *Support Vector Machine* (SVM) e *K-Nearest Neighbors*. Para o *clustering*, algoritmos hierárquicos e o *K-means* adaptável. Cada uma dessas possibilidades foi devidamente avaliada de forma a perceber quais as melhores soluções face aos problemas propostos.

Foi também feita uma breve abordagem à sumarização de documentos, contudo, este é um tema secundário. O principal foco do trabalho desenvolvido é a classificação e *clustering* de texto.

Este trabalho foi feito em cooperação com LIAAD/INESC TEC - Laboratório de Inteligência Artificial e Apoio à Decisão sob a supervisão do Dr. Nuno Escudeiro

Palavras-chave: Texto, Classificação, *Clustering*, Notícias, Desporto

Abstract

This report has been made as part of the Computer Engineering Master's dissertation from School of Engineering – Polytechnic of Porto.

The report has been developed in order to aid the implementation of an automatic process for sports news classification and clustering. That module will be implemented in a web application related with sports.

The main goal for this research is to understand among various possibilities which ones fit best given the necessary requirements of the module to be developed. Those who present the best evaluations will be chosen to be implemented in the classification and *clustering* module.

Firstly has been made a survey of the state of the art in order to have knowledge of all possibilities. Given those possibilities, for each topic were selected two algorithms. The chosen algorithms were those that found to be the most suitable.

For text categorization were selected the Support Vector Machine (SVM) and the K-Nearest Neighbors (KNN) algorithms. For document *clustering*, were selected hierarchical algorithms and the adaptable *k-means* algorithm. Then, each of these possibilities have been properly evaluated in order to understand which are the best solutions.

Was also made a brief approach to the documents summarization, however, this is a secondary topic. The main focus of this report is document classification and *clustering*.

This work was made in cooperation with LIAAD/INESC TEC – "Laboratório de Inteligência Artificial e Apoio à Decisão" with supervision of Dr. Nuno Escudeiro

Keywords: Text, Classification, *Clustering*, News, Sports

Agradecimentos

O meu maior agradecimento vai para os meus pais, Amadeu Almeida e Olívia Oliveira. Sempre me incentivaram para que nunca desistisse e desse sempre o meu máximo nesta jornada académica. Reconheço também o grande esforço a nível financeiro feito pelos meus pais. Um especial agradecimento aos meus irmãos, Susana Almeida e Nuno Almeida.

Um grande agradecimento também a alguns grandes amigos que tive o prazer de conhecer nesta nobre casa. Bruno Costa, Tomás Sotomaior, Nelson Sousa, Joaquim Silva, Diogo Garcia, entre outros. Devo também muito aos grandes e velhos amigos de infância, João Ruela, Tiago Queiroz e Bruno Pinho.

Devo também agradecer aos amigos e colegas de curso que dispensaram um pouco do seu tempo para me ajudar em algumas avaliações e questionários. São eles Daniel Sousa, Emanuel Sousa, André Rebelo, Tiago Pina, Samuel Rodrigues, Pedro Henriques, entre outros.

Por fim, e não menos importante, um especial agradecimento ao Dr. Nuno Escudeiro que prontamente aceitou ser meu orientador. Foi sempre uma pessoa muito acessível e sempre disponível para ajudar em qualquer tipo de questão.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Equações

Acrónimos

RSS	*Rich Site Summary*
TFxIDF	*Term Frequency x Inverse Document Frequency*
SVM	*Support Vector Machine*
KNN	*K-Nearest Neighbors*
PLN	Processamento de Linguagem Natural
GNU GPL	GNU General Public License
POS	Part-of-Speech
NER	Named Entity Recognition

1 Introdução

1.1 Apresentação do problema

Este trabalho de investigação surge após a ideia de um conjunto de amigos para o desenvolvimento de um projeto relacionado com o desporto. Este relatório servirá como referência para a viabilidade da implementação de alguns dos módulos do projeto. Caso seja possível a sua implementação, quais as melhores hipóteses e soluções a usar na fase de implementação.

A quantidade existente de notícias desportivas disponível hoje em dia é muito grande. Hoje em dia praticamente toda a gente tem um *smartphone* ou *tablet* e qualquer tipo de conteúdo *online* pode ser consultado praticamente em tempo real. Quem conseguir fornecer esse conteúdo em primeiro lugar parte desde logo em vantagem. O problema proposto vai de encontro a este ponto, a organização automática de vários RSS *feeds.*

Esse problema pode ser dividido em três questões mais pequenas. A categorização automática de notícias de acordo o seu conteúdo, o *clustering* de notícias relacionadas e ainda identificação das principais entidades (pessoas, organizações, cidades, etc.) das notícias. O último ponto terá uma relevância secundária. O principal foco de abordagem neste trabalho é a classificação e o *clustering*.

Assim, face a estes problemas serão analisadas e avaliadas as melhores e possíveis soluções para resolver os problemas referidos.

1.2 Motivações

A principal motivação é a de criar um produto que junte dois mundos pelo qual existe um gosto especial, os sistemas de informação e o desporto, mais concretamente o Futebol. Relacionado, está também a vontade de criar um produto ao qual possa constar no meu portefólio para possível benefício profissional futuro.

Hoje em dia, a cada dia surgem milhares de novos artigos desportivos, sendo humanamente impossível processar tal quantidade de informação manualmente. Uma das motivações é perceber como esse problema pode ser solucionado de forma automática e como

apresentar aos utilizadores informação bem organizada tendo em conta alguns critérios. Apresentar apenas a informação que o utilizador tenha realmente interesse, evitando assim que procure alternativas para a consulta da mesma informação.

1.3 Objetivos e abordagem ao problema

Neste trabalho será avaliada a possibilidade de se obter uma boa classificação automática de notícias desportivas para um determinado número de categorias pré-definidas (ver capítulo 5.1.3). A principal diferença com outros trabalhos de investigação que focam o mesmo tema é que estas categorias são, no fundo, subcategorias muito específicas e próprias de um desporto. A maior parte dos trabalhos relacionados realizados abordam categorias mais genéricas, como por exemplo, Cinema, Desporto, Política, Economia ou, mesmo dentro do desporto a categorização em diversas modalidades – Futebol, Ténis, Andebol, entre outros.

Essas categorias mais abrangentes possuem um maior número de termos comuns, exemplificando, termos como “atleta”, “equipa”, “venceu”, “perdeu”. Neste trabalho foi avaliada a capacidade de classificação automática quando as categorias são muito específicas. Neste caso as categorias pré-definidas podem ser consideradas como subcategorias de uma subcategoria (Modalidade: Futebol) de uma categoria (Desporto).

O segundo problema é uma consequência direta do primeiro. Avaliou-se a capacidade e possibilidade de agrupar de forma rápida e eficaz um conjunto de vários documentos que partilhem um evento em comum. Por evento entenda-se, por exemplo, assuntos relacionados com uma determinada partida de futebol entre duas equipas.

O terceiro problema está diretamente relacionada com os resultados obtidos na tarefa de agrupamento de notícias. Consiste na extração das principais palavras que identifiquem um conjunto de notícias.

A abordagem a estes problemas começa, obrigatoriamente, pelo pré-processamento da coleção (corpus) obtida. Foram aplicadas algumas transformações ao texto como remoção de espaçamentos múltiplos, remoção de pontuação, remoção de *stopwords*, *stemming*, entre outros. O modelo de representação escolhido foi o *bag-of-words*, principalmente devido à sua simplicidade de representação. Para o corpus foi depois criada a respetiva matriz TFxIDF – *Term Frequency x Inverse Document Frequency*.

Para a tarefa de classificação, foram criados diversos modelos de forma a serem avaliados diversos cenários. Foram testados vários níveis de remoção das palavras menos frequentes e vários classificadores. Neste trabalho foram utilizados o *support vector machines* (SVM) e o *k-nearest neighbors* (KNN). O KNN e SVM são frequentemente dos mais utilizados e que melhores resultados obtêm, daí a sua escolha. A avaliação dos resultados foi feita com auxílio à técnica de validação cruzada e às métricas adequadas.

A tarefa de agregação de notícias foi realizada com recurso a algoritmos de *clustering*. O *clustering* é uma técnica para fazer agrupamentos automáticos de informação tendo em conta o seu grau de semelhança. Os algoritmos desta técnica podem ser divididos em particionais e hierárquicos. Face à natureza do problema, os algoritmos particionais foram automaticamente excluídos. O seu algoritmo mais conhecido, o *k-means* necessita que seja indicado um valor para o número de *clusters* a serem criados. O problema apresentado não permite a determinação de qual esse número.

Os algoritmos hierárquicos não necessitam que seja indicado um valor para o número de *clusters* a serem criados, pelo que é uma solução potencialmente válida para a resolução do problema. O *k-means* adaptável é uma solução híbrida que tem por base o tradicional algoritmo *k-means*. Organiza a informação pelos *clusters* da mesma forma que o *k-means* tradicional mas não necessita da especificação do número de *clusters* a serem criados.

A avaliação deste processo foi feita com recurso às métricas propostas pelos investigadores da área e, também com auxílio a avaliações manuais por voluntários.

Foi também feita uma breve introdução à técnica de nuvem de palavras como forma de extração das palavras que melhor identifiquem um conjunto de documentos.

1.4 Contributos deste trabalho

Com este trabalho, foram produzidos os seguintes contributos:

1. Um documento que aborda e avalia algumas técnicas de classificação e *clustering* para documentos de idioma português.

2. Um estudo acerca da classificação automática de documentos tendo por base categorias muito específicas (ver capítulo 1.3).

3. Um estudo acerca do agrupamento (*clustering*) de documentos com base em eventos e acontecimentos relacionados.

4. Código reutilizável para outros desportos ou idiomas. Sendo necessária apenas pequenas alterações do código e/ou criação de novos *datasets* para outros desporto e idiomas.

5. Os algoritmos que melhor se adequam face às exigências necessárias para a implementação do módulo de classificação automática e *clustering* de notícias desportivas.

6. Criação de um *dataset*[1] de documentos em português classificados tendo em conta oito categorias. O *dataset* foi doado ao repositório de aprendizagem máquina da Universidade da California – Invine. (À data de publicação do presente relatório o *dataset* estava pendente de aprovação).

1.5 Tecnologias usadas

Este trabalho teve como base o *text mining*. O *text mining* ou descoberta de conhecimento em texto é o processo de descoberta de informação nova e útil de fontes de texto não-estruturadas.

Para o desenvolvimento deste trabalho foi utilizado **R**. O R é uma linguagem de programação bastante utilizada para cálculos estatísticos e gráficos e o seu código fonte está disponível sob a licença GNU GPL. Pode ser utilizado tanto para *data mining* como *text mining.*

O R é um ambiente bastante expansível, existindo diversos pacotes – bibliotecas – que auxiliam na execução de determinadas funcionalidades. O principal pacote para o *text mining* é a **biblioteca TM** (CRAN 2014). O R está disponível em http://www.r-project.org/.

Foi utilizado também o **RStudio**, um ambiente de desenvolvimento integrado que oferece um ambiente mais agradável para o utilizador. Pode ser descarregado em http://www.rstudio.com/.

Para a gestão bibliográfica de referências foi utilizada a aplicação **Docear**, a qual está disponível em http://www.docear.org/.

[1] http://tinyurl.com/md9rrm2

Foi também utilizado o **Microsoft Word** e o **Microsoft Excel**. O primeiro como ferramenta de escrita e o último para manipulação dos dados gerados, como por exemplo, criação de tabelas, gráficos, etc.

1.6 Estrutura do relatório

O presente relatório está dividido em nove capítulos, podendo cada um deles dividir-se em vários subcapítulos.

1. **Introdução –** neste capítulo é dado a conhecer ao leitor alguns aspetos acerca do projeto de forma a poder perceber em traços gerais o que será abordado durante todo o restante relatório. É apresentado o problema, as motivações para a sua resolução bem como os objetivos e a abordagem ao problema, entre outros.

2. ***Text mining*** – este capítulo descreve o estado-da-arte da área do *text mining* em geral. É apresentado o que é aprendizagem máquina, o *text mining*, e alguns aspetos relacionados com o pré-processamento de texto, entre outros.

3. **Categorização de texto** – capítulo que diz respeito ao estado-da-arte na área específica da classificação de texto ou documentos. Aqui são apresentados vários dos algoritmos existentes bem como as formas de avaliar este processo.

4. **Agregação/sumarização de documentos** – capítulo também referente ao levantamento do estado-da-arte. Aqui são explorados os algoritmos de *clustering* bem como as suas formas de avaliação. É também explorada a sumarização de documentos.

5. **Abordagem ao problema –** este capítulo introduz o problema com maior detalhe. Aqui, o leitor vai poder perceber, passo-a-passo, como foi encarado o problema.

6. **Classificação automática** – neste capítulo são apresentados e interpretados todos os resultados obtidos durante a fase de experimentação dos processos de classificação de texto automática.

7. **Clustering** – neste capítulo são apresentados e interpretados todos os resultados obtidos durante a fase de experimentação dos processos de agregação de notícias por eventos.

8. **Tagging** – neste capítulo são retiradas algumas conclusões sobre a melhor forma de identificar *clusters*.

9. **Conclusões** – neste capítulo são apresentadas as conclusões obtidas face ao problema e objetivos iniciais propostos. Caso existam, serão referenciadas as limitações e dificuldades que existiram ao longo do trabalho desenvolvido e também o trabalho futuro a desenvolver.

2 Text mining

Este capítulo tem como objetivo a introdução ao *text mining* e principais conceitos associados. Serão abordados alguns aspetos essenciais do *text mining*, como por exemplo, o pré-processamento, a seleção de características, os modelos de representação de documentos, entre outros.

Será também feita uma introdução aos vários tipos de aprendizagem máquina existentes.

2.1 O que é o *text mining*?

O *text mining* ou descoberta de conhecimento em texto é o processo de descoberta de informação nova e útil de fontes de texto não-estruturadas. A descoberta de padrões é geralmente o processo mais utilizado na aquisição dessa informação.

Assim como o d*ata mining*, o *text mining* surgiu da necessidade de descoberta de informação de forma automática, neste caso em textos. Estima-se que mais de 90% da informação presente nas organizações está na forma de informação não-estruturada, sejam cartas, correio eletrónico, contratos, informações técnicas ou até mesmo notícias relevantes para o funcionamento de organização (Gomes et al. 2013).

Existem várias áreas em que o *text mining* pode ser aplicado:

- Classificação de documentos – também conhecida por categorização, é o processo de etiquetar documentos com uma categoria. Utiliza documentos existentes e previamente classificados para calcular a categoria mais adequada para cada novo documento. A classificação de um artigo com base no seu tópico (desporto, política, saúde, etc.) pode ser uma das suas aplicações.

- Extração de Informação (*Information Extraction*) – é o processo de pesquisa de respostas num documento de uma determinada frase de pesquisa. Os motores de pesquisa são uma aplicação muito frequente nesta área.

- *Clustering* – é um processo que tenta, para um conjunto de documentos, agrupá-los da melhor forma possível tendo em conta semelhança entre si. Um caso bem real

para uma aplicação desta área pode ser o agrupamento de notícias todas relacionadas com um evento ou entidades.

- Resumo de documentos – é um processo mais recente e ainda muito em estudo. Engloba um pouco de todas as áreas acima mencionada. A apresentação de um sumário que represente um artigo ou um conjunto de artigos é, obviamente, a sua principal aplicação.

2.2 *Text mining* vs *data mining*

O *text mining* e o *data mining* não são assim tão diferentes. Os principais métodos de aprendizagem máquina (capítulo 2.3) para texto requerem que este seja transformado para uma representação numérica. Numa fase intermédia a informação é apresentada no formato tradicional do *data mining* (Weiss et al. 2010).

Para representar um documento, o normal é recorrer-se a tabelas. Por exemplo, a presença de uma palavra num documento pode ser representada sob a forma binária, uns (1) e zeros (0). Sendo "um" indicador da presença de uma palavra, "zero" o caso contrário. A tabela seguinte representa um conjunto de documentos e palavras.

Tabela 1 – Representação de documentos

Documento	Porto	Benfica	Sporting	Boavista
"FC Porto vence Boavista no Estádio do Dragão"	1	0	0	1
"Benfica contrata novo médio-centro"	0	1	0	0
"Benfica, Sporting e FC Porto lutam pelo título"	1	1	1	0
"Cristiano Ronaldo marca mais três"	0	0	0	0
"Sporting empata a zero bolas"	0	0	1	0

A representação dos documentos sob a forma de uma matriz esparsa leva a que por muitas vezes sejam armazenados muitos zeros. Isto acontece porque são representados na matriz a presença ou não de todas as palavras em todos os documentos. Como quase nunca existem frases iguais, é natural que o número de palavras aumente à medida que são introduzidos novos documentos. Para simplificar esta representação, pode-se optar por uma representação em vetores. Para além de oferecer uma representação mais simples, ainda que possa parecer mais difícil de compreender, permite poupar alguns recursos máquina. Na tabela seguinte é representada a Tabela 1 sob a forma vetorial.

Tabela 2 - Representação de documentos em forma vetorial

Documento	Vetor
"FC Porto vence Boavista no Estádio do Dragão"	(1,4)
"Benfica contrata novo médio-centro"	(2)
"Benfica, Sporting e FC Porto lutam pelo título"	(1,2,3)
"Cristiano Ronaldo marca mais três"	()
"Sporting empata a zero bolas"	(3)

Contudo, a utilização de uma representação binária não é obrigatória. Podem ser optadas formas alternativas da representação da frequência de uma palavra em um documento. Uma alternativa é a utilização de um sistema ternário, um sistema baseados em três valores.

- 0 – Palavra não ocorreu.
- 1 – Palavra ocorreu uma vez.
- 2 – Palavra ocorreu duas ou mais vezes.

Outras alternativas são a frequência dos termos (*term frequency)*, ou *term frequency-inverse document frequncy* (TFxIDF). No capítulo 2.7 serão apresentados com maior atenção algumas formas de representação de documentos.

2.3 Aprendizagem máquina

A aprendizagem máquina é uma área da inteligência artificial cujo objetivo é a construção de sistemas e modelos que permitam os computadores "aprender" através da análise dos seus dados e exemplos (Hotho et al. 2005). Tipicamente existem três tipos de aprendizagem: supervisionada, não-supervisionada e semi-supervisionada. Mais recentemente, tem sido introduzido pela comunidade de investigação o conceito de aprendizagem ativa, ou *active learning*.

A diferença entre os vários tipos de aprendizagem assentam basicamente na informação existente nos dados iniciais. Na aprendizagem supervisionada existe informação previamente classificada, na não-supervisionada não existe informação classificada e a aprendizagem semi-supervisionada situa-se entre a supervisionada e não-supervisionada. A aprendizagem ativa é semelhante à aprendizagem semi-supervisionada mas prossupõe a intervenção humana de um especialista.

2.3.1 Aprendizagem supervisionada, não-supervisionada e semi-supervisionada

A aprendizagem máquina **supervisionada** assume que existem exemplos previamente classificados, isto é, a classe que gerou cada padrão no modelo de classificação é conhecida. O modelo é então utilizado para tomar a decisão correta sempre que existem novas amostras a serem processadas. Uma das aplicações para este tipo de aprendizagem é a classificação de documentos.

Na aprendizagem **não-supervisionada** não existem exemplos previamente classificados, não existem padrões previamente descobertos. Assim, os algoritmos têm que encontrar uma estrutura nos dados que permita dividi-los em grupos que partilhem características semelhantes entre os seus constituintes.

A aprendizagem **semi-supervisionada**, como o próprio nome indica, utiliza os dois paradigmas referidos anteriormente. Combina informação classificada e não classificada para construir o conjunto de dados de treino (Gabriel 2009). A ideia por trás deste paradigma é a utilização de uma reduzida quantidade de exemplos classificados ao invés de se classificar um largo conjunto de exemplos. Isto ajuda a diminuir o esforço necessário na classificação manual de documentos pelos especialistas. A utilização de dados não-classificados através de métodos semi-supervisionados permitem ter uma precisão aceitável no que toca às tarefas de classificação (Prakash & Dr. Nithya 2014).

2.3.2 Aprendizagem ativa

O processo de **aprendizagem ativa** (Escudeiro 2012; Gabriel 2009; Settles 2009), ou *active learning*, está relacionado com os métodos de aprendizagem semi-supervisionada. A ideia por detrás destes métodos é que os algoritmos de aprendizagem máquina podem atingir uma maior precisão com uma menor quantidade de informação classificada. A possibilidade de escolher quais os dados que serão utilizados para treino é o passo diferenciador relativamente a todos os outros tipos de aprendizagem.

Aqui, durante a fase de aprendizagem, o algoritmo tem a permissão de perguntar a um utilizador, geralmente humano, para classificar alguns exemplos quando existem dúvidas acerca da classe a atribuir. Estes pedidos são denominados de *queries*. É um processo útil para quando existe imensa informação não-classificada e o custo da sua classificação é muito alto.

Neste relatório, será introduzido brevemente um algoritmo de aprendizagem ativa desenvolvido por Escudeiro (Escudeiro 2012), o *d-confidence*. A Figura 1 esquematiza, sumariamente, o processo de execução dos algoritmos de aprendizagem ativa em tarefas de classificação.

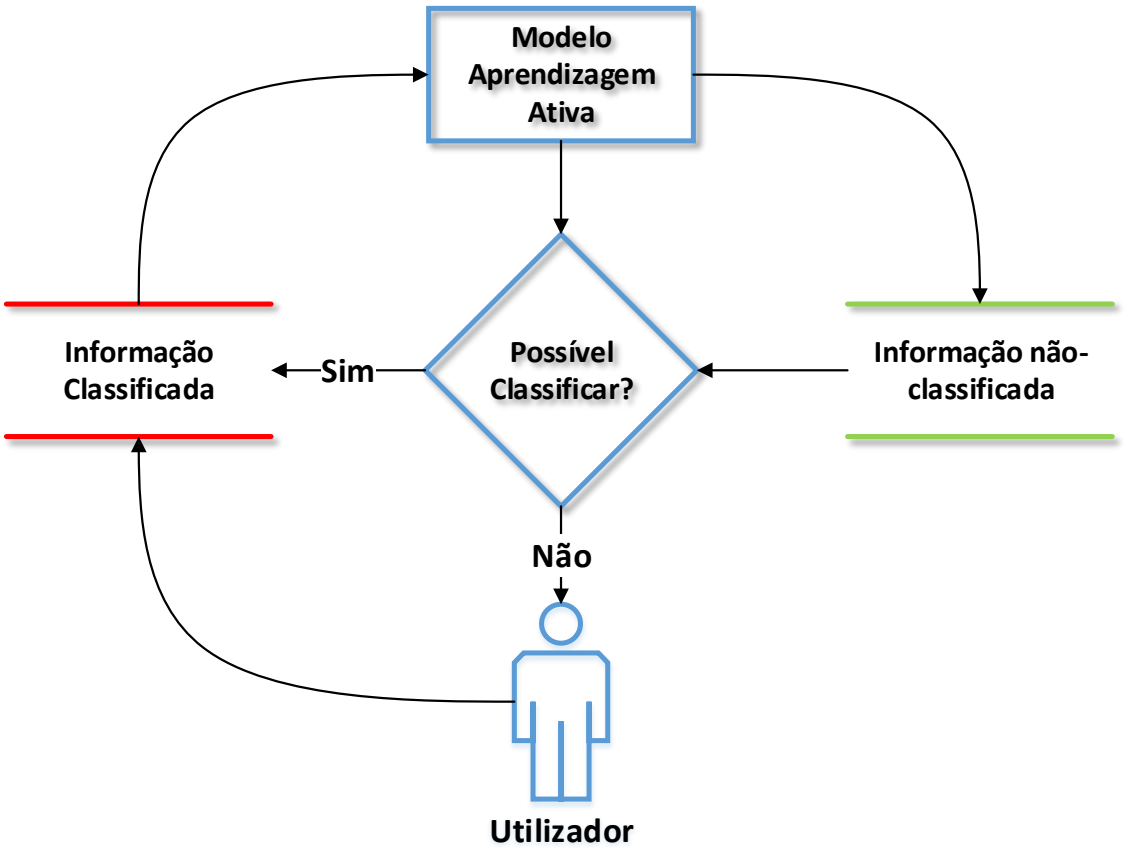


Figura 1 - Ciclo da aprendizagem ativa

2.4 Processamento de linguagem natural

O processamento de linguagem natural (PLN) é um ramo da ciência da computação que se foca no desenvolvimento de sistemas que permitam a comunicação com as pessoas através da sua língua do dia-a-dia. A sua grande preocupação é perceber como podem os métodos computacionais ajudar no entendimento da linguagem humana (Mooney 2014). Por outras palavras, como converter linguagem humana em uma representação que seja de fácil manipulação máquina. Desempenha um papel fundamental para o *text mining* uma vez que é utilizado na etapa de pré-processamento, permitindo um primeiro nível de estruturação de dados (Briscoe 2011; Gomes et al. 2013; Liu 2014).

O Processamento de Linguagem Natural pode ser resumido a um total de sete níveis de linguagem (Liu 2014):

- Fonético – lida com a interpretação sonora do discurso.

- Morfológico – está relacionado com a estrutura e formação das palavras. A ideia é olhar para as palavras isoladamente, retirá-las completamente do contexto da frase onde estava inserida e estudá-la. Uma forma de aplicação deste nível é através da divisão de palavras em prefixos, radicais e sufixos – a cada um desses elementos dá-se o nome de morfema – e assim tentar reconhecer significados nos morfemas. Por exemplo, em palavras que o sufixo é "s", é bastante provável que sejam palavras no plural.

- Léxico – aqui as palavras e o seu significado é interpretado de forma independente da frase onde está integrado. Cada uma das palavras é classificada de acordo a categoria gramatical que assume na frase - substantivos, adjetivos, numerais, pronomes, verbos, entre outros.

- Sintático – procura analisar as palavras de uma frase de forma a descobrir e validar a sua estrutura gramatical. Para isso, é necessário uma gramática e ainda um analisador sintático. Atualmente, já existem redes léxico-conceptuais tanto para o inglês como para o português, elas são a WordNet e WordNet.PT (University of Princeton; Centro de Linguística da Universidade de Lisboa). O resultado final é a perceção das dependências relacionais das palavras.

- Semântico – é frequentemente confundido com o nível sintático e, apesar de estarem relacionados, não são a mesma coisa. No nível sintático o objetivo é perceber apenas as relações entre palavras e validá-la conforme a gramática utilizada, se uma frase faz sentido ou não, é neste nível que tal é verificado. Aqui é feito, por exemplo, a desambiguação de palavras com múltiplos sentidos.

- Discurso e pragmático – perceber os significados de uma frase ou então entre frases. Deixa-se de trabalhar ao nível das palavras para trabalhar ao nível frásico.

O PLN possibilita diversas tarefas úteis para o pré-processamento de texto. No capítulo seguinte serão explicadas, com algum detalhe, aquelas que se perfilam como as mais importantes para o trabalho a desenvolver.

2.5 Pré-processamento

O pré-processamento (Escudeiro 2012; Gabriel 2009; Nogueira et al. 2008; Ramasubramanian & Ramya 2013; Silva et al. 2005) é um processo muito importante no *text mining*. O texto é informação não-estruturada, por isso, o pré-processamento é necessário para se proceder à transformação e representação do texto para uma forma minimamente estruturada. As tarefas de tratamento de erros, limpeza de ruído e redução de informação são as mais usuais nesta fase.

É uma fase crucial e crítica, um bom pré-processamento da informação permite distinguir a informação importante da pouco relevante. Um bom pré-processamento permite uma maior qualidade dos resultados finais quando aplicados processos de classificação, *clustering*, entre outros (Ramasubramanian & Ramya 2013; Weiss et al. 2010).

Algumas das tarefas realizadas neste processo podem ser: A tokenização, um processo da fase de pré-processamento onde as frases são "partidas" em palavras individuais. A remoção de s*topwords* remove as palavras que não apresentam qualquer valor linguístico. O *stemming* que passa todas as palavras à sua raiz semântica. O *POS (Part-of-Speech) tagging* identifica as partes do discurso e o reconhecimento de entidades é como o próprio nome indica, a identificação de entidades (pessoas, organizações, cidades, países, etc.).

2.5.1 Tokenização

A Tokenização (Gabriel 2009; Sajjanhar & Zhao 2012; Sanwaliya et al. 2010; Weiss et al. 2010) é o processo de partir uma frase nos seus componentes mais elementares, palavras, e assim permitir uma manipulação mais fácil do texto. A estas palavras dá-se o nome de *tokens*.

Este processo é parte integrante da análise sintática de um texto, pelo que está muito dependente de cada língua, pois cada uma tem as suas particularidades (Hassler & Fliedl 2006; Weiss et al. 2010). O único ponto que por vezes é comum é no que diz respeito ao tratamento de delimitadores ou sinais de pontuação. Estes são geralmente ignorados pois não acrescentam

qualquer valor aos métodos de aprendizagem. Uma exceção é, por exemplo, o apóstrofo que é bastante utilizado na língua inglesa.

Este processo apresenta algumas limitações, principalmente ao nível da pontuação, por exemplo, em palavras com hífen ou apóstrofo (Witten 2005). Luiza Gabriel (Gabriel 2009) refere um exemplo bem evidente deste problema, as palavras *doesn't* e *does not* apesar de gramaticalmente diferentes, têm o mesmo significado, por isso deve ser tomada uma decisão acerca desta questão sob pena de *does not* ser considerado duas palavras diferentes.

Segundo Hassler e Fliedl (Hassler & Fliedl 2006), que por sua vez citam outros autores, este passo é geralmente o primeiro a ser aplicado durante a fase de pré-processamento.

2.5.2 Remoção de stopwords

As s*topwords* são palavras que não apresentam qualquer valor linguístico. Geralmente estas palavras têm a única e exclusiva tarefa de fazer a conexão entre frases, auxiliando na estruturação da linguagem. As s*topwords* englobam proposições, artigos, conjunções, pronomes, advérbios, entre outros.

Este processo requer uma lista das palavras a serem removidas. Obviamente que estas palavras diferem de língua para língua. No ranks.nl[2] podem ser consultadas listas de *stopwords* de várias línguas, inclusive português e inglês.

A prévia tokenização da informação traz vantagens a este processo, pois poderá comparar individualmente cada palavra (Sajjanhar & Zhao 2012; Silva et al. 2005) e assim remover as palavras desnecessárias de forma rápida e simples.

2.5.3 Stemming

Em poucas palavras, o processo de *stemming* consiste em passar todas as palavras à sua raiz semântica. Caso existam, prefixos e/ou sufixos são removidos. Seguem-se alguns exemplos para perceber melhor o que é a forma raiz de uma palavra.

- Aconselhamento → Aconselh
- Regular → Regul

[2] http://www.ranks.nl/stopwords

- Irregular → Irregul

É um processo que tem as suas vantagens e desvantagens (Rodrigues 2013). Se por um lado o mapeamento de palavras em um único termo permite uma maior abrangência das consultas, as pesquisas irão obviamente tornar-se menos precisas. Ainda que o índice de palavras possa ser reduzido até 50%, isto pode originar alguma perda de informação.

Este processo está muito dependente da sua aplicação. Weiss, Indurkhya, Zang e Damerau (Weiss et al. 2010) aconselham aquando da dúvida da sua utilização que se teste as duas formas, com e sem *stemming* e, assim, determinar se há vantagens ou não em utilizar este processo.

O algoritmo de Porter (Porter 1980) é, provavelmente, o algoritmo de *stemming* mais conhecido para o inglês. Cada língua requer o seu próprio algoritmo devido às diferenças gramaticais na formação das palavras. Orengo e Huyck (Orengo & Huyck 2001) propuseram um algoritmo específico para a língua portuguesa.

2.5.4 Part-of-speech tagging

A identificação de partes do discurso, ou *part-of-speech (POS) tagging*, é o processo de atribuir uma parte do discurso (nome, verbo, pronome, preposição, advérbio, adjetivo, etc.) a cada palavra de uma frase sob a forma de uma etiqueta (*tag)*.

Uma das principais aplicações deste processo é a desambiguação do sentido de palavras. Contudo, existe o problema de uma palavra poder pertencer a mais que uma categoria gramatical. Por exemplo, nas frases "Eu gosto do João" e "O João tem bom gosto", a palavra "gosto" apesar de estar escrita de igual forma, a sua categoria gramatical é diferente. A primeira frase seria internamente representada da seguinte forma.

"Eu/PRO gosto/VER do/PRE João/SUB".

Onde "PRO" indica um pronome, "VER" um verbo, "PRE" uma preposição e "SUB" um substantivo. Hoje em dia, é possível ter POS *taggers* com uma precisão superior a 95% (Sogaard 2010; Spoustova et al. 2009; Toutaniva et al. 2003; Tsai & Chen 2003). O mais conhecido e utilizado nos dias de hoje para a língua inglesa é o *Penn Treebank POS Tagger*[3]. O Grupo de Fala

[3] http://www.cis.upenn.edu/~treebank/home.html

e Linguagem Natural do Departamento de Informática da Universidade de Lisboa tem vindo a desenvolver um *POS Tagger* para a língua Portuguesa[4].

2.5.5 Named entity recognition

O reconhecimento de nomes de entidades, em inglês n*amed entity recognition* (Tkachenko & Simanovsky 2012), foca-se na identificação de entidades no texto. As entidades podem ser nomes de pessoas, cidades, países, organizações, entre outros. É um processo que está bastante relacionado com o *POS Tagging*, sendo possível a sua integração em *POS taggers* (Tkachenko & Simanovsky 2012).

Como o contexto deste projeto se relaciona com notícias desportivas, esta etapa do pré-processamento pode assumir um papel de grande relevância. Isto porque, tradicionalmente, as notícias desportivas referem atletas, clubes, competições, organizações, entre outros.

2.6 Seleção de características

Durante a fase de preparação de dados, este processo é provavelmente o mais relevante, pois vai preparar o conjunto de dados para a próxima fase. Até aqui, o objetivo era remover palavras que não acrescentassem valor. Agora, o objetivo é o completo oposto. Selecionar as palavras que apresentem maior relevância na representação do documento e, assim, melhorar a precisão dos modelos de aprendizagem máquina. Além disso, permite que seja evitado sobreajustamento, em inglês *overfitting* (Escudeiro 2012). O sobreajustamento aparece quando se tem um modelo que atua muito bem nos dados de treino mas quando lida com o conjunto de teste, isso já não acontece.

A dimensão das características é algo que deve ser tida em conta, uma vez que quanto maior for o conjunto de dados, a precisão dos modelos de aprendizagem tenderá a ser menor (Sajjanhar & Zhao 2012). Existem diversas formas de selecionar as características de um documento e assim reduzir a sua dimensão (Escudeiro 2012):

- *Document frequency threshold* – depende da frequência do documento inversa, o número de documentos onde a característica está presente. Elimina as características que estão a abaixo de um certo limite pré-definido. É uma técnica

[4] http://lxcenter.di.fc.ul.pt/tools/pt/LXTaggerPT.html

que apresenta alguma simplicidade e um reduzido custo computacional. Xu et al. (Xu et al. 2008) refere que tem sido provado este método ser um dos melhores para língua chinesa e inglesa.

- *Information Gain* – ou ganho de informação, ordena de forma decrescente as características pelo seu ganho de informação, as características mais informativas são mantidas enquanto as restantes são removidas do conjunto de dados.

- *Mutual information* – ou informação mútua, mede a associação entre características e classes com base numa tabela de contingência de duplo sentido. As características com mais informação mútua são as selecionadas.

- *Chi-square* – ou chi-quadrado, utiliza a mesma tabela de contingência que o método de informação mútua, mas realiza o teste estatístico chi-quadrado para verificar a independência entre as características e categorias. Comparativamente ao método de informação mútua, este método tem a vantagem da normalização realizada durante o cálculo estatístico, pelo que permite comparações entre característica para a mesma classe.

- *Term strength* – ou força do termo, é um método com características bastante distintas do anterior. Este método calcula o peso de cada termo independentemente da categoria do documento. Assume-se que documentos que partilhem várias palavras são semelhantes e que palavras em comuns são informativas. Este método estima a importância de um termo com base na probabilidade condicional de aparecer num determinado documento.

- *The Markov blanket criterion* – reduz o conjunto de características excluindo incrementalmente as características menos relevantes até que o subconjunto de características seja satisfatório.

- *Latent semantic indexing* – ou indexação semântica latente, é uma técnica de reparametrização que usa a decomposição de valores singulares de uma matriz documento-termo para reduzir a dimensão do conjunto de característica. As suas principais aplicações são a identificação de padrões nas relações entre termos.

2.7 Modelos de representação de documentos

Após o processo de pré-processamento e seleção de características estar concluído é necessário apresentar devidamente os dados. Permitir que a sua interpretação nas fases seguintes seja mais fácil (Gabriel 2009). Assim, o texto é convertido para um formato que permita o seu processamento automático.

Existem dois modelos de representação de documentos que devem ser tidos em conta, o *bag-of-words* (saco-de-palavras) e o *n-gram* (n-grama).

2.7.1 Bag-of-words

Os documentos são encarados como um saco de palavras. Dentro deles existem palavras e frases que descrevem o documento. As palavras, frases ou associação de palavras podem ser também denominados por termos. É comum ver este método representado sob a forma de matriz, chamada de matriz documento-termo (ou termo-documento), onde estão representados os documentos d_i, os termos t_j e o respetivo peso w_{ij}.

Tabela 3 - Matriz documento-termo

	t_1	t_2	t_3	t_4
d_1	$w_{1,1}$	$w_{1,2}$	$w_{1,3}$	$w_{1,4}$
d_2	$w_{2,1}$	$w_{2,2}$	$w_{2,3}$	$w_{2,4}$
d_3	$w_{3,1}$	$w_{3,2}$	$w_{3,3}$	$w_{3,4}$
d_4	$w_{4,1}$	$w_{4,2}$	$w_{4,3}$	$w_{4,4}$
d_5	$w_{5,1}$	$w_{5,2}$	$w_{5,3}$	$w_{5,4}$

A principal vantagem deste modelo é o seu custo computacional, uma vez que mesmo com grandes dimensões consegue ter uma performance razoável. Porém, apresenta algumas limitações que podem ser problemáticas (Agarwal et al. 2012; Alves 2010). Ignora as relações entre palavras e não considera a ordem das palavras nos documentos. Isto pode trazer alguns problemas na análise semântica do documento.

O saco de palavras possuiu três modelos de representação do texto, o modelo booleano (Bezerra & Goldschmidt 2010; Escudeiro 2012; Lee 1994), o probabilístico (Alves 2010; Bezerra & Goldschmidt 2010; Robertson & Jones 1976) e o modelo vetorial (*vector space model*) (Bezerra & Goldschmidt 2010; Escudeiro 2012; Salton et al. 1975; Yuan et al. 2013).

Este último é, geralmente, a forma mais utilizada para a representação de documentos. Aqui, cada documento é representado por um vetor de termos.

A cada termo do vetor é atribuído um peso, podendo este ter vários formatos. Binário se apenas representar a presença ou não de uma palavra no documento, inteiro se indicar o número de ocorrências de uma palavra no documento, ou real, representando o peso calculado no processo de seleção de características (Alves 2010; Gabriel 2009).

Tradicionalmente, o vetor de um documento apresenta a seguinte notação (Bezerra & Goldschmidt 2010):

$$\overrightarrow{d_j} = (\omega_{1j}, \omega_{2j}, \dots, \omega_{mj}) \quad (1)$$

Onde:

- d_j representa o documento j.
- ω_{mj} o peso de cada termo m no documento j.

Como cada documento é representado por um vetor, este pode ser desenhado no espaço, sendo por isso possível calcular o grau de semelhança entre dois documentos. A semelhança entre dois documentos é feita através do cálculo do cosseno do ângulo entre os dois vetores, ou *cosine similarity*. No capítulo 3.3.2 podem ser encontrados detalhes sobre a forma de cálculo da similaridade do cosseno.

Para cada palavra de um documento podem ser efetuados alguns cálculos de forma a ser calculada a sua relevância, vulgo peso, no documento. A escolha da abordagem de pesagem é bastante importante. Referenciando outros estudos (Cingiz & Diri 2012), este processo tem mais impacto no resultado final do que a escolha do próprio algoritmo.

A relevância dos termos em um documento pode ser feita com recurso a vários tipos de representações:

1. Peso booleano

É uma abordagem bastante simples (Escudeiro 2012), o peso de um termo varia sempre entre 1, em que um termo está presente no documento e, 0, o termo não está presente no documento. Neste caso, o modelo de espaço vetorial é um caso particular do modelo booleano.

$$w_{ij} = 1, \forall i,j : F_{ij} > 0; w_{ij} = 0, \forall i,j : F_{ij} = 0 \quad (2)$$

2. Frequência de palavras

Outra abordagem muito simples é o uso da frequência de um termo no documento. É uma abordagem que apenas usa a informação acerca do número de vezes que uma palavra aparece no documento.

$$w_{ij} = F_{ij} \quad (3)$$

3. TFxIDF (Weiss et al. 2010)

Este método consiste em multiplicar a frequência de uma palavra, em inglês *term frequency* (TF) pelo inverso da frequência do documento, em inglês *inverse document frequency* (IDF). Simplificando, consiste na multiplicação do número de vezes que um termo aparece num documento pelo inverso da frequência do documento. Neste método é tido em conta a frequência do termo ao longo do corpus. Esta função verifica o número de documentos em que ocorre a palavra e inverte a sua escala.

Este método considera uma palavra com pouca importância quando está presente em muitos documentos, sendo, portanto, o valor de IDF mais baixo. As equações 4 e 5 representam como o cálculo do IDF e TFxIDF são realizados:

$$IDF(t_j) = \log(\frac{N}{df(j)}) \quad (4)$$

$$w_{ij} = TF(d_i, t_j) * IDF(t_j) \quad (5)$$

Onde N é o número de documentos na coleção e $df(j)$ o número de documentos onde o termo t_j aparece. A fórmula apresentada para o cálculo do IDF não é a única, existem várias alternativas (Escudeiro 2012).

4. Normalização do TFxIDF (Chang & Hsu 2005)

O cálculo do TFxIDF não tem em conta o tamanho dos documentos. A equação 6 surge na tentativa de normalizar esse fator:

$$|\sum_{n=i}^{N}(tf(d_i, d_j) * idf(t_j))^2|^{\frac{1}{2}} \quad (6)$$

5. Entropia (Escudeiro 2012)

Baseia-se na teoria da informação (equação 7)

$$w_{ij} = \log(f_{ij} + 1) * \left(1 + \frac{1}{\log(N)} * \sum_{i=i}^{N}\left(\frac{f_{ij}}{N_j} * \log(\frac{f_{ij}}{N_j})\right)\right) \quad (7)$$

Em que $\frac{1}{\log(N)} * \sum_{i=i}^{N}\left(\frac{f_{ij}}{N_j} * \log(\frac{f_{ij}}{N_j})\right)$ é a entropia média do termo t_j. Esta é igual a 1 se a palavra é igualmente distribuída sobre todos os documentos e 0 se o termo ocorre em apenas "um" documento.

Os métodos de cálculo de peso acima mencionados são os mais frequentes, porém, têm vindo a ser apresentadas novas abordagens. Algumas novidades, outras refinamentos ou variações das já existentes. Agarwal et al. (Agarwal et al. 2012) apresentam a medida CFxIDF, que tenta considerar relações entre conceitos e propriedades, sendo verificadas algumas melhorias na precisão da classificação utilizando este método.

2.7.2 N-grams

Os n-grams, ou n-gramas em português, é uma outra forma de representação dos documentos. É um método puramente estatístico e capaz de substituir palavras e frases por n-gramas, não sendo necessários dicionários e regras (Liu et al. 2012). Os n-gramas consistem na sequência de n palavras consecutivas. Segue-se um exemplo da aplicação de n-gramas à palavra "Estados Unidos da América".

- $n = 1$ – Unigrama ("Estados").
- $n = 2$ – Bigrama ("Estados Unidos").
- $n = 3$ – Trigrama ("Estados Unidos da").

No caso anterior, caso se optasse pela utilização de $n = 4$, seria possível reconhecer a palavra Estados Unidos da América. Contudo, a sua utilização pode provocar um aumento da dimensão do conjunto de dados, uma vez que os unigramas, bigramas e trigramas fazem a tarefa de reconhecimento de padrões muito mais complexa para o sistema (Dilrukshi et al. 2013).

Alexandra Alves (Alves 2010) refere que a utilização de n-gramas permite manter a ordem das palavras, permitindo assim conservar o contexto das palavras, o que não acontece no *Bag-of-Words*.

3 Categorização de Texto

Neste capítulo é introduzido o processo de categorização de texto e as suas etapas envolventes. Nessas etapas incluem-se a aprendizagem do modelo (treino), aplicação do modelo a um conjunto de teste e respetiva avaliação.

Serão apresentados alguns dos classificadores que existem para a classificação de documentos.

3.1 Classificação de Texto

A classificação de texto (ou categorização) é a tarefa de atribuir categorias (ou classes, tópicos) pré-definidas a um conjunto de documentos (Sebastiani 2005). É a tarefa de atribuir um valor a cada par $\{d_j, c_i\} \in D\ X\ C$, onde D é o domínio dos documentos e $C = \{c_1, c_2, \dots, c_{|c|}\}$ um conjunto de categorias pré-definidas (Sebastiani 2002).

O processo de classificar documentos possui três etapas bem definidas. A primeira é a aprendizagem do modelo (ou treino). Para um dado conjunto de documentos (documentos de treino) é-lhe aplicado um classificador que constrói um modelo que represente esses dados. Esse modelo é depois aplicado a documentos que ainda não tenham sido classificados (documentos de teste). O resultado final é a atribuição de uma categoria a esse novo documento. No final é realizada uma avaliação à aplicação do modelo nos novos documentos, existindo para isso um conjunto de métricas adequadas.

Existem vários métodos para treino, os quais serão abordados com algum detalhe no subcapítulo seguinte. A Figura 2 esquematiza o processo de treino e classificação de um novo documento.

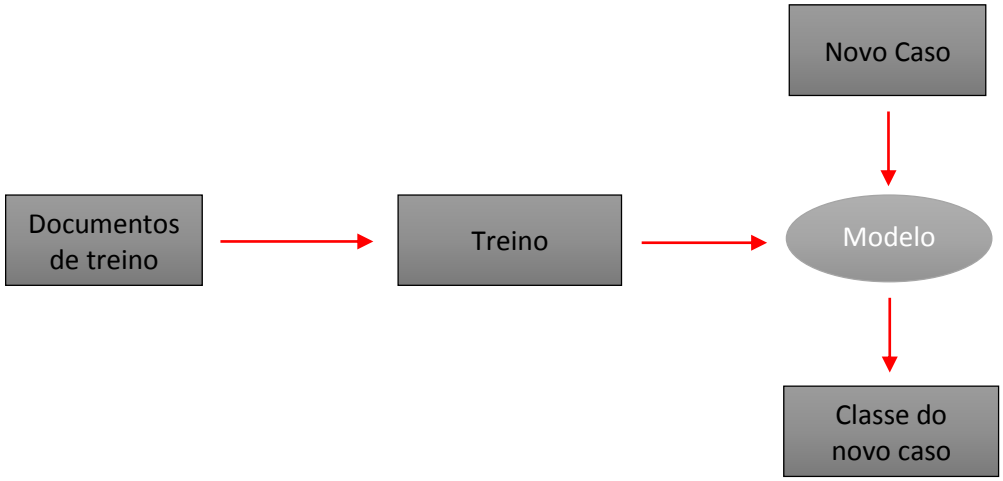


Figura 2 - Classificação novo documento

3.2 Classificadores

Um classificador é um algoritmo que, dada uma ou mais classes de entrada, decide automaticamente a que classe pertence um dado documento com base na análise do seu conteúdo (Alves 2010).

Existem diversos classificadores propostos para a tarefa de classificação de texto. Neste capítulo serão abordados os mais conhecidos e utilizados, sendo que a maioria desses classificadores pertencem à categoriza da aprendizagem supervisionada. Será também introduzido um algoritmo de aprendizagem ativa.

3.2.1 Naïve Bayes

O *Naïve Bayes* (Bayes 1763; Escudeiro 2012; Sebastiani 2002) é um classificador que pertence à família dos modelos probabilísticos e é baseado no teorema de Bayes. De uma forma simplificada podemos referir o teorema de Bayes como a probabilidade de um documento representado pelo vetor $\overrightarrow{d_j} = (\omega_{1j}, \omega_{2j}, \dots, \omega_{ij})$ pertencer à classe c_i. Pode ser expressa pela equação 8:

$$p(c_i|\overrightarrow{d_j}) = \frac{p(\overrightarrow{d_j}|c_i)p(c_j)}{p(\overrightarrow{d_j})} \quad (8)$$

Onde:

- $p(c_i|\overrightarrow{d_j})$ é a probabilidade à posteriori[5] de $\overrightarrow{d_j}$ pertencer à classe c_j.
- $p(c_j)$ é a probabilidade de observar a classe c_j.
- $p(\overrightarrow{d_j}|c_i)$ é a probabilidade de observar $\overrightarrow{d_j}$ dada a classe c_j.
- $p(\overrightarrow{d_j})$ é a probabilidade de se observar $\overrightarrow{d_j}$.

Neste classificador a dependência entre termos é ignorada. Assume que a probabilidade condicional de um termo de uma dada categoria é independente da probabilidade condicional de outro qualquer termo de uma dada categoria, daí a assunção *naïve* (ingénuo).

Quando assumida a independência de termos, a probabilidade condicional de um documento d_j, dada a classe c_i, pode ser obtida com recurso à equação 9:

$$p(\overrightarrow{d_j}|c_i) = \prod_m p(t_m|c_i) \quad (9)$$

Onde t_m representa os termos do vetor que representa o documento d_j.

Assim, para um documento, o algoritmo calcula as probabilidades *à posteriori* (Escudeiro 2012) de cada uma das classes e aquele que for a mais provável é atribuída ao documento.

Apesar da sua simplicidade, que é a sua principal vantagem, pode apresentar alguns problemas e desvantagens (Pinto 2005). A sua simplicidade pode também ter o reverso da medalha e transformar-se em uma desvantagem. Em situações mais complexas, pode ter alguns problemas.. Apesar de ter um bom desempenho com conjuntos de treino pequenos, o mesmo pode não ser verdade quando estes aumentam.

3.2.2 Árvores de decisão

As árvores de decisão são representações gráficas, geralmente diagramas, e que apresentam uma estrutura em árvore. É representada como uma partição recursiva descendente no espaço. A construção destas árvores começam num nó raiz (*root*) e terminam, eventualmente, em um conjunto de nós terminais, os quais não têm nenhuma ligação de saída.

[5] http://en.wikipedia.org/wiki/Posterior_probability

A esses nós dá-se o nome de folhas ou nós de decisão. Todos os nós intermediários denominam-se de nós de teste ou nós internos.

Cada nó tem apenas um ramo de entrada mas pode dividir-se em dois ou mais ramos de saída (Rokach & Maimon 2005). A divisão é feita com o objetivo de minimizar a diversidade das categorias presentes em cada nó e é feita até não ser possível mais nenhuma melhoria (Escudeiro 2012).

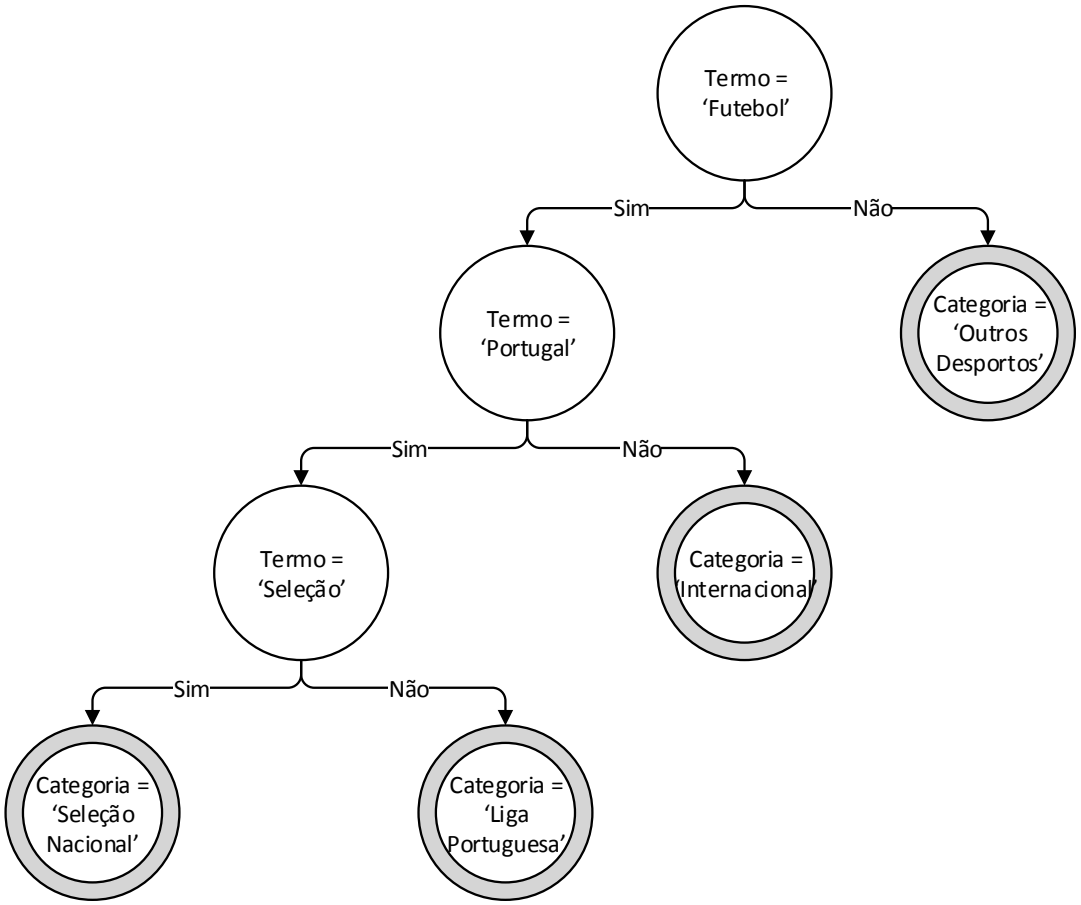


Figura 3 - Árvore de decisão

Os nós internos incluem o teste a um atributo, frequentemente um termo, que por sua vez, pode ser dividido em dois ou mais ramos e que representam o resultado do teste. Duas ou mais categorias, por exemplo. Os nós de decisão indicam, por exemplo, uma categoria. Na classificação de documentos, os documentos de teste são então sucessivamente divididos em grupos, até que estes apenas contenham apenas documentos pertencentes a uma determinada categoria (Reis 2012). A Figura 3 apresenta uma representação gráfica e simplificada do processo de classificação de um documento usando árvores de decisão.

Segundo Escudeiro (Escudeiro 2012), a forma de decisão sobre qual o atributo que irá permitir a melhor divisão, a que gera a partição mais homogénea, é uma questão muito importante. A entropia é uma das unidades de medida mais utilizada para efetuar essa decisão. A entropia de um dado nó, L, é dada pela equação 10.

$$-\sum_{j}^{K} p(c_j|L)\log(p(c_j|L))$$

(10)

Onde $\mathrm{p}(c_j|L)$ é a probabilidade de uma amostra de treino que está no nó L pertencer à classe c_j, a qual pode ser calculada através da seguinte equação:

$$p(c_j|L) = \frac{N_j(L)}{N(L)}$$

(11)

Em que:

- $N_j(L)$ é o número de instâncias da classe c_j no nó L.
- $N(L)$ o número total de instâncias no nó L.

De acordo com Rokach e Maimon (Rokach & Maimon 2005), a complexidade de uma árvore tem um grande efeito na precisão. Tipicamente, a complexidade de uma árvore é feita através da medição das seguintes medidas:

- O número total de nós.
- Número total de folhas.
- Profundidade da árvore (número de níveis).
- Número de atributos (termos) usados.

De forma a evitar uma grande complexidade da árvore, podem ser utilizados critérios de paragem ou então realizar-se um processo de poda da árvore. A poda, ou *pruning*, é o processo de reduzir o tamanho da árvore de decisão através da remoção de secções que forneçam pouca capacidade de decisão. Existem duas abordagens comuns de poda: *post* ou *backward pruning* e *pre* ou *forward pruning*. Escudeiro (Escudeiro 2012) dá uma abordagem simplificada das duas abordagens. Eibe Frank (Frank 2000) dá uma análise mais detalhada das técnicas de poda.

Alguns dos algoritmos mais conhecidos e populares desta categoria são o CART(Breiman et al. 1984), ID3(Quinlan 1986), CHAID(Kass 1980) e C4.5(Quinlan 1993)

As árvores de decisão estão estreitamente relacionadas com as regras de decisão (Weiss et al. 2010), seguem inclusive a mesmo modo de funcionamento, a divisão dos dados em grupos. Frequentemente as regras de decisão podem ser extraídas de árvores de decisão.

3.2.3 Redes neuronais

As redes neuronais (Han & Kamber 2006) começaram por ser originalmente desenvolvidas nas áreas da psicologia e neurobiologia. O objetivo era desenvolver e testar análogos computacionais de neurónios. Uma rede neuronal é um conjunto de unidades interligadas entre si e com conexões de entrada e saída. Cada conexão tem um peso associado. Durante a fase de treino a rede aprende ajustando os pesos das suas ligações de forma a encontrar a classe correta para o conjunto de teste.

Uma das grandes vantagens das redes neuronais é que conseguem apresentar uma precisão bastante alta, mesmo para os problemas mais complexos. Contudo, envolvem muito tempo de treino. Ainda assim, como o conhecimento é distribuído por várias ligações, existe a possibilidade de distribuir o processamento através de técnicas de computação paralela/distribuída e assim reduzir o tempo de processamento. Apresentam uma grande tolerância a ruídos bem como boa capacidade de deteção de padrões que ainda não foram treinados.

As grandes desvantagens (Alves 2010; Han & Kamber 2006) das redes neuronais são a sua difícil utilização. São frequentemente criticadas devido à sua fraca interpretabilidade, muitas vezes é difícil para humanos perceber o seu funcionamento. O facto de requererem a configuração de muitos parâmetros e um pré-processamento muito específico dos dados provoca muitas vezes o utilizador a optar por outros algoritmos.

Os algoritmos de redes neuronais implementam frequentemente mecanismos de *backpropagation* (Rumelhart et al. 1986). Em poucas palavras, estes mecanismos permitem que em caso de erro seja identificado o neurónio responsável e, então, reajustar os seus parâmetros.

3.2.4 Rocchio

O algoritmo de Rocchio (Rocchio 1971) é um algoritmo que utiliza os vetores dos documentos de treino para construir um vetor protótipo para cada classe/categoria. Cada vetor é calculado como o vetor médio de todos os vetores dos documentos que pertencem a uma classe. Um novo documento é assim classificado de acordo a cosine *similatiry* entre o vetor dos documentos e os vetores das classes.

É um método de bastante simples implementação e o seu custo computacional tende para ser também reduzido.

3.2.5 K-nearest neighbors

O *k-nearest* neighbors (KNN) ou em português, método dos vizinhos mais próximos, é um dos métodos de classificação mais simples e deve ser encarado como uma séria primeira escolha para um estudo de classificação onde há pouco conhecimento acerca da distribuição da informação (Yi et al. 2012). É um método que apresenta frequentemente um bom desempenho em problemas relacionados com o reconhecimento de padrões e classificação de padrões.

Na classificação de um novo documento, os k vizinhos mais próximos (documentos de treino) de um documento de teste são tidos em conta e vão ter um papel determinante na atribuição da classe do novo documento. Como os documentos são representados sob a forma de vetores, é calculada a distância entre o documento de teste e os de treino. Esse cálculo é feito com algumas das seguintes métricas: distância euclidiana, similaridade cosseno (*cosine similarity*), *city block*, entre outros. A classificação é feita com base nas características dos k documentos mais próximos. Mais à frente, será explicada a forma de cálculo de alguns dessas métricas.

O k é um número inteiro e que corresponde ao número máximo de vizinhos mais próximos a ter em conta. Esta é uma tarefa essencialmente empírica, o k deve ser determinado e avaliado por tentativa e erro, podendo este variar de caso para caso. Quando $k = 1$ é atribuído ao documento de teste a classe que se encontra mais próxima no espaço. Quando $k > 1$ é realizada uma votação majoritária entre os seus vizinhos, sendo a classe mais comum entre os k vizinhos mais próximos atribuída ao documento de teste.

Uma das grandes desvantagens deste método é o seu custo computacional. Como cada documento de teste é comparado com todos os documentos de treino, em conjuntos de dados

grandes esta tarefa pode ser morosa e computacionalmente cara. Ainda que este método não precise de aprendizagem prévia, amostras previamente classificadas são necessárias (Escudeiro 2012). É um método de fácil implementação e de fácil entendimento dos resultados obtidos.

3.2.6 Support vector machine

O *support vector machine* (SVM) utiliza o modelo de espaço vetorial como forma de representação dos dados (Joachims 1998). A sua forma de funcionamento consiste na pesquisa de um hiperplano que separe duas classes distintas e que maximize a margem que separa os exemplos positivos e negativos de cada classe (Gabriel 2009). A margem é a distância entre o hiperplano e o ponto mais próximo de cada classe.

O principal objetivo é a definição de uma função discriminante a partir dos exemplos de treino mais próximos e que definam o hiperplano. Para isso utiliza vetores de suporte (vetores que representam os pontos mais próximos do hiperplano de separação) e as margens, que são definidas pelos vetores de suporte. A Figura 4 esquematiza o funcionamento do SVM.

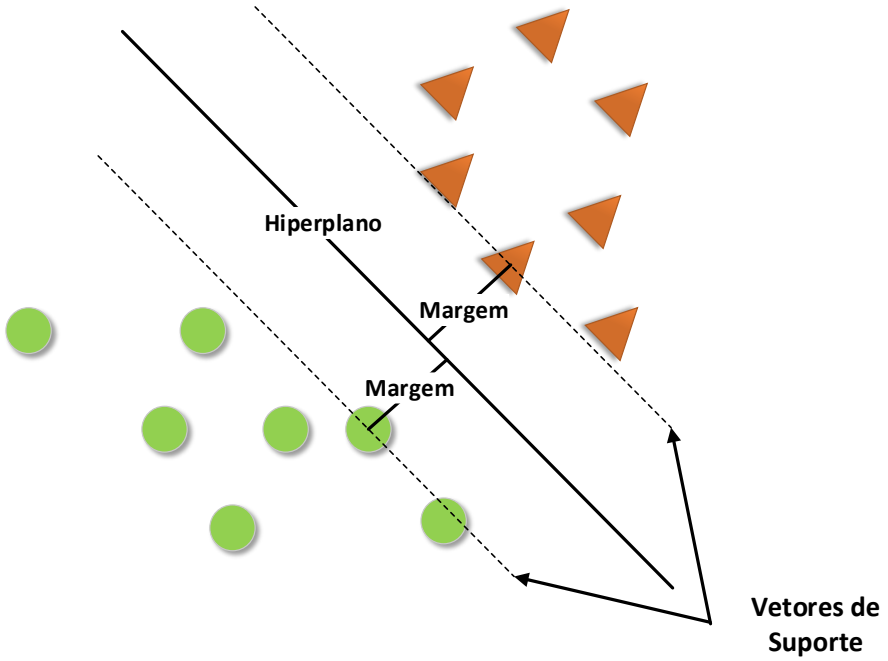


Figura 4 - SVM

A vantagem do SVM é que requere pequenos conjuntos de treinos e são menos propensos a sobreajustamento em relação a outros métodos. É também considerado um dos métodos mais eficientes de aprendizagem máquina supervisionada (Han & Kamber 2006; Sajjanhar & Zhao 2012).

Este método tem sido bastante estudado, tendo frequentemente boas performances na categorização de informação, mesmo quando existe um elevado número de atributos (Alves 2010; Reis 2012). Ainda assim, Han e Kamber (Han & Kamber 2006) referem que o tempo de aprendizagem pode ser elevado, mesmo para conjuntos mais pequenos. Porém, a sua elevada precisão anula um pouco essa desvantagem.

Sebastiani (Sebastiani 2002) cita Joachims dando duas grandes vantagens deste modelo:

- Frequentemente, não é necessário selecionar termos uma vez que o SVM tende a ser robusto no que diz respeito a sobreajustamento. São escaláveis até um considerável número de dimensões.

- Não é necessário esforço humano ou máquina no ajuste dos parâmetros iniciais. Uma escolha *default* dos parâmetros de configuração tem-se demonstrado como uma escolha com boa eficácia.

3.2.7 d-Confidence

O *d-Confidence* é um algoritmo proposto por Nuno Escudeiro (Escudeiro 2012) que funciona sob o paradigma de aprendizagem ativa. Em cada iteração, caso seja necessário, é selecionado um documento, o considerado mais informativo com base no critério *d-Confidence* e, é pedido ao utilizador para o classificar. Em cada iteração aplica-se um classificador base.

O classificador base pode ser qualquer um, contudo, na implementação do *d-Confidence* é usado o SVM. Na Figura 5 é possível visualizar de forma esquematizada o funcionamento do algoritmo *d-Confidence*.

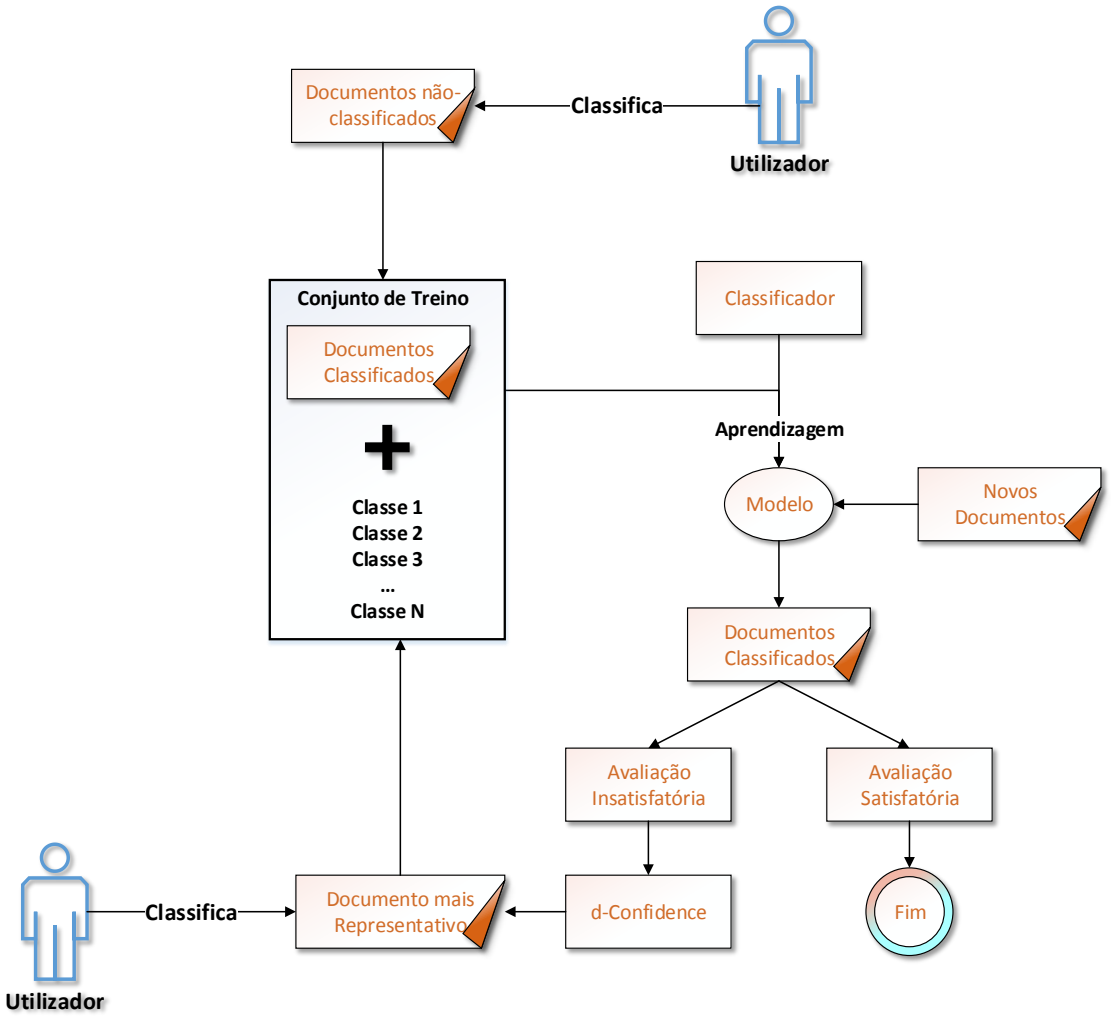


Figura 5 – Aprendizagem ativa

A ideia por detrás da proposta deste algoritmo é que as *queries* não devem ser selecionadas com base exclusivamente na confiança de um classificador na classificação de um novo exemplo. Gabriel na sua Tese de Mestrado (Gabriel 2009) apresenta um resumo do funcionamento do algoritmo.

O algoritmo seleciona as *queries* que serão apresentadas ao utilizador com base num critério que agrega a confiança posterior do classificador, a distância entre casos não classificados e as classes conhecidas. O critério seleciona os casos com baixa confiança e grande distância para as classes conhecidas. Atua em casos onde: no espaço alvo, existem poucas instâncias classificadas e, pelo contrário, existem muitas instâncias não-classificadas. Tem como objetivo a identificação de instâncias representativas para todas as classes com o menor número de *queries* possíveis.

A "próxima" *querie* a ser selecionada é aquela que tem o *d-Confidence* mais baixo. Um *d-Confidence* baixo combina a baixa confiança do classificador (provavelmente indicando as instâncias das classes desconhecidas) com a grande distância para as classes conhecidas (apontando para as regiões invisíveis no espaço). É obtido através do rácio entre a confiança do classificador e a distância entre as instâncias não classificadas e as classes conhecidas. Pode ser calculada da seguinte forma (Escudeiro 2012):

$$dConf(x_j) = \max_k(\frac{conf(x_j, c_k)}{d_j^k}) \quad (12)$$

Onde para uma dada instância não classificada, $x_j \in U_i$, o classificador gera a confiança posterior em relação às classes conhecidas. As distâncias entre um documento não classificado x_j e todos os documentos classificados pertencentes à classe $c_k \in C_i$, d_j^k, são calculadas pela seguinte equação.

$$ClassDist_i(x_j, c_k) = d_j^k = mediana(dist(x_j, L_i^k)) \quad (13)$$

Em que:

- x_j é um documento.
- L_i^k todas as instancias classificadas em L_i pertencentes a c_k.

A Figura 6 representa a escolha do documento que será utilizado como *querie* ao utilizador. De forma a simplificar o entendimento do processo, assume-se que todas as instâncias não classificadas têm a mesma confiança (Escudeiro 2012).

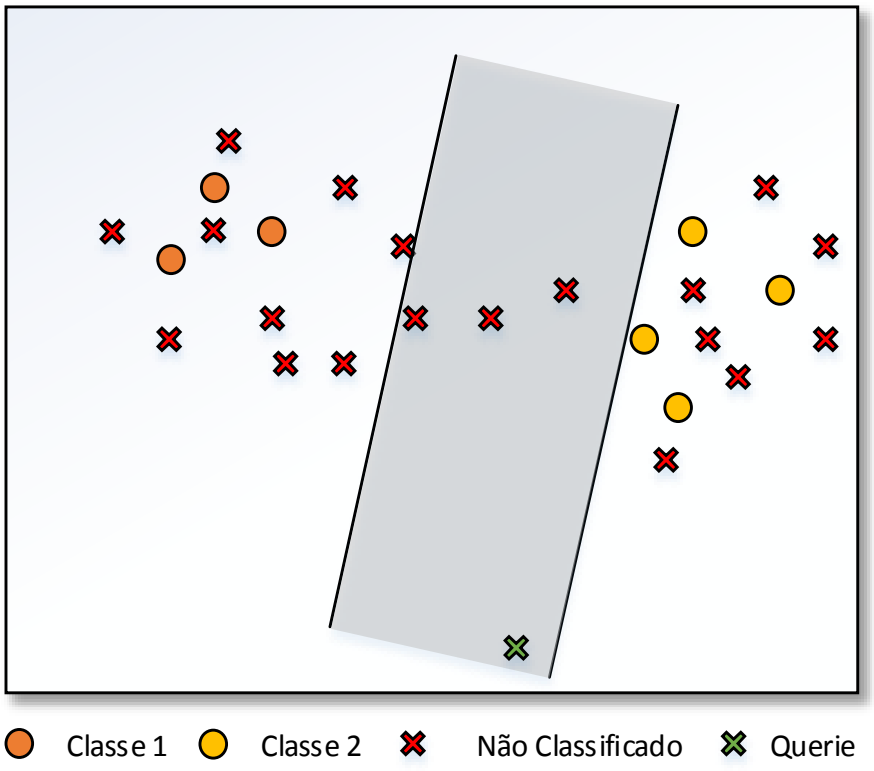


Figura 6 - *d-Confidence* – Seleção de instância para querie

3.3 Semelhança entre documentos

Como referido anteriormente, no problema da classificação de texto os documentos podem ser representados sob a forma de um vetor, podendo serem desenhados no espaço. Isto permite calcular o grau de semelhança entre dois documentos. Existem várias formas de o fazer, sendo as mais conhecidas a distância euclidiana e a similaridade do cosseno.

3.3.1 Distância euclidiana

A distância euclidiana é a distância entre dois pontos. Suponha-se que se tem os seguintes pontos do documento $d_1 = (w_{11}, w_{12}, \dots, w_{1j})$ e do documento $d_2 = (w_{21}, w_{22}, \dots, w_{2j})$. A distância é calculada através da equação 12:

$$d(d_1, d_2) = \sqrt{\sum_{j=1}^{n} \left(w_{1j} - w_{2j}\right)^2} \quad (14)$$

Onde:

- n é o número total de atributos do documento de teste
- w_{1j} e w_{2j} são os vários atributos dos documentos d_1 e d_2.

3.3.2 Similaridade do cosseno

A similaridade do cosseno, ou *cosine similarity* em inglês, é medida através do cosseno dos vetores que representam os documentos em questão. É feito através do cálculo da seguinte fórmula:

$$simcosseno(d_1, d_2) = \frac{\sum(w_{d1}(j) * w_{d2}(j))}{(norm(d1) * norm(d2))} \quad (15)$$

Onde:

- d_1 representa o documento 1;
- d_2 representa o documento 2;
- $w_{d_1}(j)$ são os pesos dos termos do documento 1;
- $w_{d_2}(j)$ são os pesos dos termos do documento 2;

Devido aos documentos variarem no seu tamanho, a informação da frequência das palavras pode ser, por vezes, enganadora. Como tal, é frequente os pesos dos termos serem normalizados (Chang & Hsu 2005; Weiss et al. 2010) de forma a amenizar esse efeito. Essa normalização pode ser feita através do cálculo da função $norm(D)$:

$$norm(D) = \sqrt{\sum w(j)^2}, \quad (16)$$

3.4 Avaliação

Os modelos dos classificadores oferecem potenciais soluções. Não garantem que são sempre as melhores soluções, pelo que a escolha do algoritmo de classificação deve ser encarada com seriedade. Um classificador que apresente bons resultados para uma tarefa não irá ter necessariamente os mesmos resultados para uma outra tarefa.

Para avaliar um modelo, geralmente dividem-se dados em duas amostras, uma para treino e outra para teste. A amostra de treino é utilizada pelos classificadores para a construção dos modelos de classificação. A de teste é utilizada para avaliar a qualidade do modelo criado.

Existem várias formas de divisão dos dados, sendo as mais comuns o método de *holdout* (Refaeilzadeh et al. 2009) e a validação cruzada (Han & Kamber 2006). A validação cruzada consiste na divisão dos dados em k subconjuntos (D_k) de tamanho aproximado, sendo o treino e teste realizado k vezes. Assim, em cada iteração *i*, uma partição D_i é reservada para teste. É geralmente recomendado um valor de $k = 10$ devido à sua relativa baixa tendência e variância.

O método *holdout* consiste em dividir aleatoriamente $\frac{2}{3}$ das amostras para treino e os restantes $\frac{1}{3}$ para teste. Pode ser considerado como uma forma mais simples da validação cruzada. Ainda que seja um método computacionalmente menos pesado, tem a desvantagem da sua avaliação poder apresentar alta variância, uma vez que depende muito de onde os dados foram colocados. Na validação cruzada, isso não acontece pois todos os pontos de dados pertencerão exatamente uma vez ao conjunto de teste e $k - 1$ vezes ao conjunto de treino. Contudo, a validação cruzada pode tornar-se computacionalmente mais pesada, pois o algoritmo tem de ser executado k vezes. À medida que k aumenta, a variância tende a diminuir (Schneider & Moore 2000).

Todos os documentos de uma amostra de teste são previamente classificados manualmente por especialistas. Assim pode-se comparar a categoria verdadeira de um documento àquela obtida através do processo de classificação automática. Se as duas classes forem iguais, então o modelo classificou corretamente o documento. A apresentação desta informação é frequentemente feita com recurso a matrizes de confusão, a qual será abordada no capítulo 3.4.1.

3.4.1 Matriz de confusão

A matriz de confusão é uma ferramenta muito útil de análise à performance dos classificadores (Han & Kamber 2006). Grande parte das unidades de medição de performance de um classificador têm por base matrizes de confusão.

A Tabela 4 representa uma matriz de confusão para duas classes, a qual fornece informação acerca do número de classificações corretas e incorretas. Quando um classificador apresenta uma boa qualidade, a maior parte dos documentos classificados situam-se na diagonal da matriz.

Tabela 4 - Matriz de confusão

		Classe Atribuída pelo Classificador	
		C	$\overline{C}$
Classe Real	C	Verdadeiros Positivos	Falsos Positivos
	$\overline{C}$	Falsos Negativos	Verdadeiros Negativos

Em que:

- Verdadeiros Positivos (VP) – número de documentos corretamente classificados como sendo da classe C.
- Verdadeiros Negativos (VN) – número de documentos corretamente classificados como não sendo da classe C.
- Falsos Positivos (FP) – número de documentos incorretamente classificados como sendo da classe C.
- Falsos Negativos (FN) – número de documentos incorretamente classificados como não sendo da classe C.

Através da matriz de confusão é possível determinar algumas das seguintes métricas de avaliação. A exatidão, ou *accuracy*, indica a proporção do número total de classificações corretas.

$$a = \frac{VP + VN}{VP + VN + FP + FN} \quad (17)$$

A taxa de erro, ou *error rate*, é o inverso da exatidão. Mede a proporção de documentos mal classificados.

$$e = 1 - a = \frac{FP + FN}{VP + VN + FP + FN} \quad (18)$$

A precisão, ou *precision*, indica a percentagem de documentos que foram corretamente classificados com a categoria C dentro daqueles que foram classificados como C, incluindo aqueles que foram incorretamente classificados com a categoria C. Isto traduz a capacidade de

excluir os falsos positivos, os documentos que foram incorretamente classificados com a categoria C. É calculada com recurso à equação 19:

$$p = \frac{VP}{VP + FP} \quad (19)$$

A abrangência, ou *recall*, indica a relação entre os documentos que foram corretamente classificados na categoria C sobre o total de documentos dessa mesma categoria, incluindo aqueles que foram incorretamente classificados na categoria $\bar{C}$. A equação 20 apresenta a forma de se calcular a abrangência.

$$r = \frac{VP}{VP + FN} \quad (20)$$

Uma vez que a precisão e a abrangência medem diferentes tipos de erros, se a taxa de erro geral se mantiver a mesma, o aumento da precisão (reduzir um tipo de erros) leva a que a abrangência (aumento o outro tipo de erros) diminua. A este fenómeno dá-se o nome de *precision-recall tradeoff*. Os dois valores estão sempre compreendidos entre 0 e 1. A *F-measure*, ou *F1 Score*, foi proposta como forma de maximizar ambas as unidades de medida. Combina ambas as medidas em um único indicador. É definida pela média harmónica das duas medidas.

$$F1 = \frac{2}{\frac{1}{p} + \frac{1}{r}} \quad (21)$$

4 Agregação/sumarização de documentos

Hoje em dia existe uma enorme quantidade de informação digital, onde se incluem documentos de texto. Por documentos considere-se notícias, correio eletrónico, relatórios, entre outros.

A *International Data Corporation* (IDC) no seu último relatório (IDC 2014) anual acerca do universo digital publicado em Abril de 2014 corrobora a afirmação anterior. Nos dias de hoje existem cerca de 4,4 zeta bytes de informação digital e estimam que esta cresça a uma média de 40% ao ano. A previsão para 2020 é que exista cerca de 44 zeta bytes de informação em formato digital. Dada esta quantidade de informação é natural que possa existir muita informação redundante e que aborde os mesmos tópicos. Surge então a necessidade da aplicação de mecanismos de *clustering* e sumarização de forma a se reduzir essa informação.

O *clustering* tem como objetivo o agrupamento de informação tendo em conta diversos fatores, por exemplo, temas relacionadas. As técnicas de sumarização permitem criação de versões resumidas de documentos.

As técnicas de sumarização e *clustering* estão algo relacionadas. Existem algumas situações onde a criação de resumos pode ser útil e deve ser seriamente encarada:

- Documentos únicos e extensos.
- Documentos múltiplos que abordam um tema em comum.

Contudo, os programas encarregues de gerar sumários não vão reescrever os documentos originais. A sumarização automática de textos pode ser classificada sob dois paradigmas (Sureka & Kong 2006), baseada em estatística e baseada em processamento de linguagem natural. A criação de resumos baseada no PLN é ainda muito limitada, pois requer a compreensão, interpretação e parafraseamento do texto original.

Assim, os investigadores optam por uma forma mais estável para a criação de resumos que consiste na extração de palavras e frases (paradigma estatístico). Este paradigma envolve o *ranking* de frases e inclusão das frases originais mais relevantes de um dado artigo no artigo final. Aqui, o grande desafio é determinar quais os melhores critérios para selecionar as frases

mais relevantes (Wu & Liu 2003). No capítulo 4.3 será abordado com mais detalhes o paradigma de sumarização estatística.

O tamanho do resumo pode ser um parâmetro configurável pelo utilizador. Ainda assim, a taxa de compressão dos sumários é um problema chave. Yu (Yu & Ren 2009) refere, com base em outras investigações, que uma taxa de compressão entre 5% e 30% é aceitável para humanos. Quando essa taxa é superior a 30%, é provável que exista já algum ruido na informação. Nenkova e Mckeown (Nenkova & McKeown 2011) referem que sumários com uma percentagem de compressão na ordem dos 17% podem ajudar a acelerar o processo de tomada de decisão.

O *clustering* é o processo de agrupamento automático de dados de acordo com o seu grau de semelhança (similaridade). A sua aplicação no *text mining* está relacionada com o agrupamento de documentos tendo em conta a sua semelhança. Quando o processo termina, o objetivo é ter-se grupos homogéneos e bem separados entre si (Han & Kamber 2006). O *clustering* é normalmente encarado como um problema de aprendizagem máquina não-supervisionada.

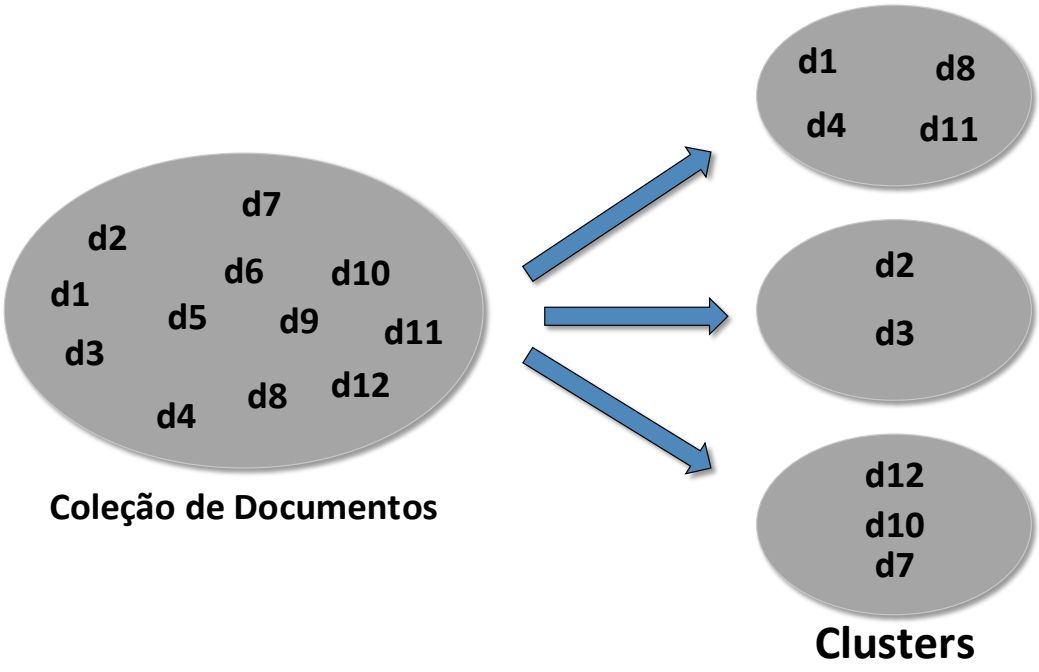


Figura 7 - *Clustering* de documentos

A similaridade é a principal característica tida em conta para o *clustering*. Esta é calculada com as medidas do capítulo 3.3. O capítulo seguinte introduz com mais detalhe algumas técnicas de agrupamento de documentos.

4.1 Algoritmos de *clustering*

Os algoritmos de *clustering* podem ser divididos em duas categorias (Bouras & Tsogkas 2010; Han & Kamber 2006; Manning et al. 2008) aglomerativos hierárquicos e particionais. São técnicas maioritariamente de aprendizagem não-supervisionada.

Os algoritmos hierárquicos geram uma série de partições na informação que pode ir de um apenas *cluster* contendo toda a informação, até n *clusters*, cada um contendo apenas um documento. São visualizáveis através de uma estrutura em árvore. Os algoritmos particionais prossupõem, na sua maioria, a indicação do número de *clusters* em que a informação irá ser dividida. Os algoritmos podem ainda ser divididos em outras duas categorias, *hard clustering* e *soft clustering*.

No hard clustering a informação é dividida em clusters distintos. Cada elemento pertence apenas e só a um cluster. Os algoritmos de soft clustering, também denominado por fuzzy clustering, permite que a informação possa pertencer a mais do que um cluster em simultâneo. Neste caso, um documento terá para cada cluster um diferente grau de adesão.

4.1.1 Algoritmos hierárquicos

Os algoritmos hierárquicos (Bouras & Tsogkas 2010) produzem um conjunto de *clusters* aninhados e que podem ser representados na forma de árvore hierárquica – dendrograma. São representadas as divisões realizadas em cada iteração. Estes algoritmos podem ser divididos em duas categorias:

- Aglomerativa (*bottom-up*) - o algoritmo começa por considerar cada instância como um *cluster* de si próprio e vai agregando informação de acordo um certo grau de similaridade entre si a cada iteração.

- Divisiva (*up-bottom*) - a informação é dividida a cada iteração. Começa-se com um único *cluster* que contém toda a informação no topo, e vários *clusters* individuais no fundo.

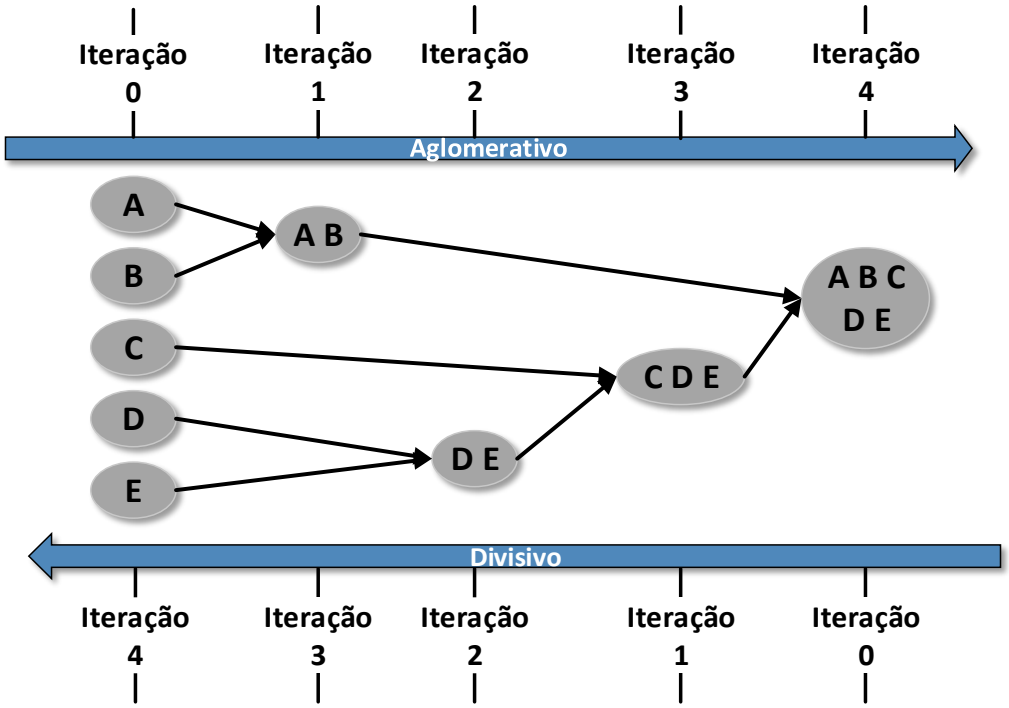


Figura 8 - Algortimos hierárquicos (aglomerativo e divisivo)

Existem várias formas de cálculo da similaridade entre *clusters* nesta categoria de algoritmos. As mais frequentemente usadas são o *single-link* (*s-link*), *complete-link* (*com-link*), *average-link* (*ave-link*). Apesar de existirem várias opções, no fim é sempre a menor distância (maior similaridade) que é tida em conta para a criação dos *clusters*. É importante perceber que estas opções apenas indicam quais os constituintes de um *cluster* que vão ser tidos em conta para o cálculo da semelhança. Não confundir com a distância euclidiana ou similaridade do cosseno. Obviamente estas serão as medidas utilizadas para o cálculo das semelhantes. A facilidade em utilizar diferentes tipos de formas para calcular a similaridade entre *clusters* acaba por ser uma das vantagens dos algoritmos hierárquicos.

No *single-link*, a similaridade entre dois *clusters* é dada pela distância entre os dois membros mais próximos. Tem a desvantagem de ser sensível a valores isolados e dificuldade em lidar com grandes diferenças de densidades entre *clusters* (Rafsanjani et al. 2012).

No *complete-link* a similaridade de dois *clusters* é dada pela distância dos seus membros mais distantes. Em relação ao *single-link* este método apresenta uma menor sensibilidade a ruídos e valores isolados (Rafsanjani et al. 2012).

No *average-link* a proximidade de dois *clusters* é dada pela distância média entre todos os membros. Surge como tentativa de amenizar as desvantagens do *single-link* e *complete-link* (Manning et al. 2008).

Existe ainda o método de Ward. Este método ao contrário dos anteriores não utiliza métricas de distância para formar os *clusters*. Encara o processo como um problema de variância dos *clusters*. Conhecido também por método da variância mínima de Ward, este faz uso da função objetivo proposta por Joe Ward (Ward 1963). Este método a cada iteração tenta encontrar o par de *clusters* em que a sua combinação leva ao menor incremento da variância do *cluster* após serem combinados. No capítulo 4.2 será introduzido o conceito de variância.

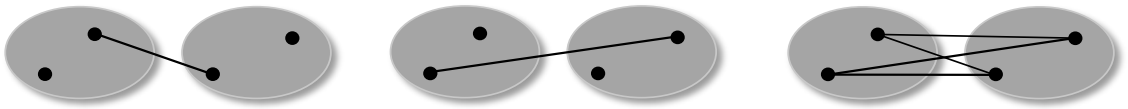


Figura 9 – (a) Single-link (b) Complete-link (c) Average-link

A maior desvantagem destes algoritmos é o tempo necessário para serem executados e a sua complexidade computacional, pelo que este método é mais recomendável para pequenas amostras de documentos.

Os algoritmos hierárquicos aglomerativos têm uma complexidade de $O(n^3)$ e os divisionistas $O(2^n)$. Alguns algoritmos particionais como o *k-means* apresentam uma complexidade linear (Manning et al. 2008). Para base dados muito grandes podem apresentar algumas dificuldades na construção dos dendrogramas.

Nos últimos anos têm sido propostos vários algoritmos aglomerativos: CURE (Guha & Shim 1998), BIRCH (Zhang et al. 1996), ROCK (Guha et al. 1999), Chameleon (Karypis et al. 1999), entre outros.

4.1.2 Algoritmos particionais

Ao contrário dos algoritmos hierárquicos, estas técnicas produzem uma divisão da informação em apenas um nível. Dado um número de *clusters* desejados, k, o algoritmo apresenta todos os k *clusters* de uma só vez. Ainda assim, estes algoritmos funcionam de forma iterativa na atribuição dos documentos aos *clusters*.

Para que os resultados de *clustering* sejam precisos privilegia-se uma baixa distância entre os elementos de um *cluster* e altas distâncias entre os *clusters*. Os algoritmos mais típicos são o *k-means*, *k-medians* e *k-medoids*.

São algoritmos que baseiam-se na noção de um centro no *cluster*, um ponto no espaço formado com ajuda dos vetores dos objetos de um *cluster*. As suas diferenças consistem na

forma como os centros são definidos (Bouras & Tsogkas 2010). O *means*, *medians* e *medoids* indica a forma como os centróides são construídos.

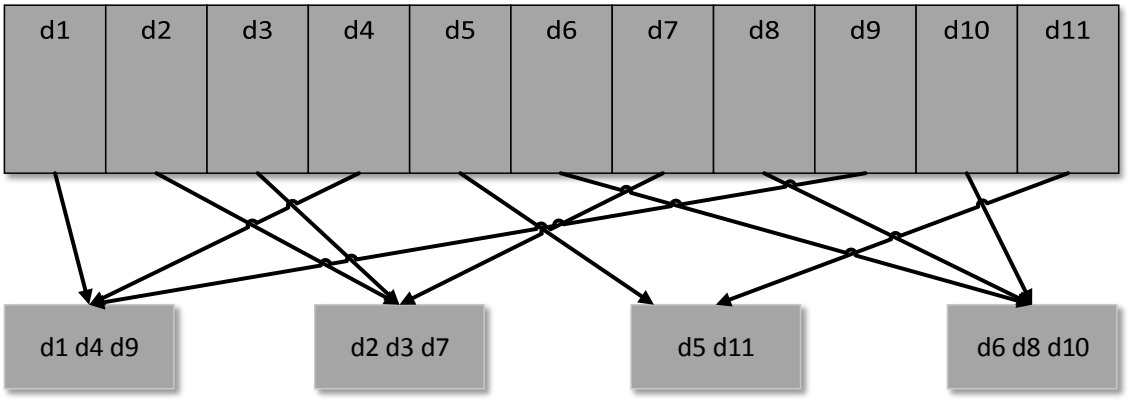


Figura 10 - Algoritmo particional

A baixa complexidade é um denominador comum nos algoritmos particionais, por isso são os ideais para *clustering* de grandes bases de dados de documentos (Bouras & Tsogkas 2010).

Apesar de existirem mais algoritmos particionais para além do *k-means*, apenas este será descrito (no próximo subcapítulo) com maior detalhe. É um dos mais utilizados no *clustering* de documentos (Steinbach et al. 2000). O algoritmo da maximização de expectativas (ME) (Dempster et al. 1977; Gupta & Chen 2011) é uma das principais alternativas, até porque o *k-means* é uma variante deste algoritmo. A diferença entre os dois é que ME é um algoritmo de *soft clustering* enquanto o *k-means* é de *hard clustering*.

4.1.2.1 K-means

O *k-means* é o algoritmo mais importante dos algoritmos particionais (Macqueen 1967). É um método clássico de *clustering* de dados numéricos que foi adotado para documentos. É amplamente usado e eficiente. O algoritmo recebe um parâmetro de entrada, k, e divide um conjunto de n documentos por k *clusters*. O seu objetivo é minimizar a média quadrada da distância euclidiana dos seus documentos em relação aos centros dos *clusters* (centróide). Tenta criar *clusters* compactos mas bem separados entre si.

No *k-means* o centro do *cluster* é definido pelo vetor de médias dos vetores que representam os documentos do *cluster*. O *k-means* funciona da seguinte forma (Weiss et al. 2010):

1. Distribui todos os documentos entre os k *clusters*. A primeira iteração atribui aleatoriamente os documentos pelos *clusters*.

2. Calcula o vetor de médias para cada *cluster*.

3. Compara o vetor de cada documento com os vetores de médias dos *clusters* e assinala o vetor cuja similaridade é maior.

4. Move os documentos para os *clusters* mais semelhantes.

5. Se nenhum documento foi movido para um novo *cluster* o processo termina, senão, volta ao passo 2. Isto significa que foi atingido um critério de convergência, um ponto em que não existe nenhuma troca, ou um valor mínimo da função objetivo entre iterações.

A função objetivo deste algoritmo é a soma do quadrado dos erros entre os objetos do *cluster* e o respetivo centro – SQE. Para cada ponto, o erro é a distância para o centro do *cluster* mais próximo. Pode ser calculado utilizando a equação 22:

$$SQE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x) \quad (22)$$

Onde:

- x é um ponto no *cluster*.
- m_i o centro do cluster C_i.

Dadas duas partições, escolhe-se aquela com o menor erro. Uma forma de reduzir o SQE é aumentando o valor de k.

A definição do valor do k gera sempre alguma confusão. À semelhança do algoritmo de classificação KNN, devem ser realizados conjuntos de testes com vários valores de k de forma a atestar qual o melhor valor para k.

Na próxima sequência de imagens, é apresentada a sequência de atribuição de *clusters* a vários documentos de um corpus. A cruz representa o centro do *cluster*, a linha mais clara o *cluster* antigo e a linha mais escura o novo *cluster*.

Nota: As distâncias são aproximadas. O objetivo é apenas fazer uma representação simbólica das várias iterações do *k-means*.

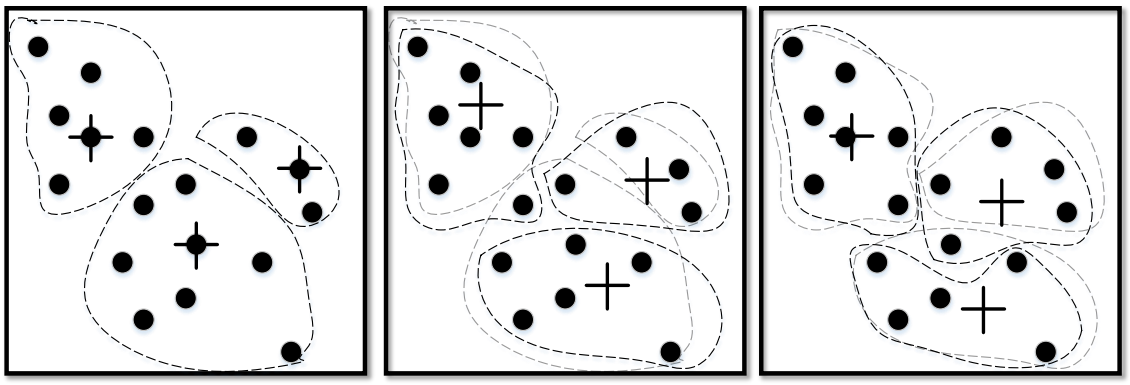


Figura 11 - Funcionamento *k-means*

É frequentemente uma das primeiras escolhas na investigação de novos *datasets*. A sua algoritmia simples, robustez e fácil programação são algumas das suas vantagens. Contudo, existem dois grandes inconvenientes (Kaur & Kaur 2013):

1. Como a cada iteração a distância entre cada ponto e os centros dos *clusters* têm que ser calculada, esta tarefa pode-se tornar lenta e dispendiosa em grandes volumes de dados.

2. Este método é bastante sensível aos *clusters* iniciais. Como na primeira iteração os *clusters* são gerados aleatoriamente, diferentes *clusters* iniciais podem gerar diferentes *clusters* finais.

4.1.2.2 K-means adaptável baseado em threshold

O k-means (Bhatia 2004) adaptável, ou *adaptative k-means* em inglês, é um algoritmo proposto por Sanjiv Bhatia em 2004. Como o próprio nome indica, é um algoritmo cuja base é o *k-means*. O seu funcionamento base é igual ao original. Começa por distribuir a informação em vários *clusters* e vai reajustando-a a cada iteração. Para o ajustamento dos *clusters* utiliza as mesmas formas de cálculo de distância.

A grande diferença para o *k-means* tradicional é que este não precisa que seja indicado um valor inicial de k. O algoritmo tenta encaixar novos pontos de entrada num dos *clusters* existentes, caso não seja possível cria um novo. Para este processo é necessário indicar um valor

limite (*threshold*) para o ajustamento dos *clusters*. Este valor corresponde à distância máxima que um ponto de entrada pode ter para poder pertencer a um *cluster*.

Bhatia (Bhatia 2004) resume o funcionamento do algoritmo passo a passo:

1. Dado um *dataset* inicial, é escolhido um elemento. É criado um *cluster* com esse elemento a ser sua semente.

2. Para cada novo elemento, é calculada a sua distância para os centróides dos *clusters* existentes. Se essa distância for menor que o valor de *threshold* especificado, o elemento é adicionado ao *cluster* mais próximo. Se após todos os *clusters* serem analisados não existir um *cluster* adequado é criado um novo *cluster* com esse elemento.

3. O centróide dos *clusters* são recalculados. A distância dos *clusters* afetados para os outros *clusters* é também recalculada.

4. Se, como resultado do passo anterior, a distância de um novo *cluster* para outro for menor que o valor de *threshold*, os dois *clusters* são combinados. As distâncias entre *clusters* são recalculadas.

5. O processo está concluído quando todos os elementos tiverem sido atribuídos a um *cluster*.

Quando o valor de *threshold* é baixo, a tendência é que exista um menor número de *clusters*. No caso contrário, quando esse valor é mais elevado, a tendência é que sejam criados mais *clusters*.

4.2 Avaliação c*lustering*

Ao longo do tempo têm vindo a ser propostas medidas para serem usadas na avaliação entre algoritmos de *clustering*. Essas medidas podem ser divididas em dois grupos:

- Avaliação interna – a avaliação é feita tendo em conta apenas a própria informação dos *clusters* criados. Neste tipo de avaliação é aplicado o conceito de similaridade intra-*clusters* e dissimilaridade inter-*cluster* para avaliar os *clusters*. Geralmente, as medidas desta categoria atribuem um índice (normalmente entre 0 e 1) para avaliar a qualidade do método utilizado.

- Avaliação externa – este tipo de avaliação requer que a informação esteja previamente etiquetada de forma a poder ser possível estabelecer uma comparação entre o algoritmo executado e a informação inicial. Utiliza algumas das mesmas medidas usadas no problema da classificação de texto, como por exemplo, precisão, abrangência, F-Measure, etc. (Rendón et al. 2011).

Têm sido propostas várias medidas para a avaliação interna de *clusters*. Seguem-se algumas das medidas mais conhecidas (Rendón et al. 2011):

1. Índice BIC – tem por base o teorema de Bayes, e é usado para determinar qual a mistura baseada em probabilidades é a mais apropriada. Foi pensado para evitar sobreajustamento e é representado pela seguinte função:

$$BIC = -\ln(L) + vln(n) \quad (23)$$

Onde:

- n é o número de objetos.
- L é a probabilidade de um parâmetro gerar os dados num modelo.
- v é o número de parâmetros livres no modelo Gauss.

O índice de BIC tem em conta o ajustamento do modelo aos dados e a complexidade do modelo. Um modelo com um menor índice BIC é um modelo melhor.

2. Índice de Davies-Bouldin - foi introduzido em 1979 por David L. Davies e Donald W. Bouldin (Davies & Bouldin 1979). Mede a similaridade média entre cada *cluster* e aquele que mais se assemelha. Tem o objetivo de identificar conjuntos de *clusters* que são compactos e bem separados. Um valor mais pequeno de DB indica uma melhor solução. É definido pela seguinte função:

$$DB = \frac{1}{c}\sum_{i=1}^{c} \max_{i<>j}\left\{\frac{d(X_i) + d(X_j)}{d(c_i, c_j)}\right\} \quad (24)$$

Onde:

- c é o número de *cluster.*
- i, j são as etiquetas dos *clusters*.
- $d(X_i)$ e $d(X_j)$ são todas as amostras nos *clusters* i e j para os respetivos centróides. $d(c_i, c_j)$ é a medida de similaridade entre dois grupos.

3. Índice de silhueta - mede a qualidade da estrutura dos *clusters* tendo em conta a compacidade dos *clusters* e a separação entre *clusters*. Foi introduzido por Pete Rousseeuw em 1987 (Rousseeuw 1987). O valor do índice situa-se no intervalo $[-1,1]$ e deve ser maximizado (Brock et al. 2008).

   Dado o *cluster* $X_j (j = 1, \dots c)$, este método atribui ao membro i de X_j uma medida de qualidade, $s(i) = (i = 1, \dots m)$, conhecida como largura da silhueta. Este valor é um indicador da confiança do membro i no *cluster* X_j. É representado através da seguinte função:

$$s(i) = \sum_{i=i} \frac{(b(i) - a(i))}{\max\{a(i), b(i)\}} \quad (25)$$

   Onde:

   - $a(i)$ é a distância média entre i e todas as amostras incluídas em X_j.
   - $b(i)$ é a distância média aos objetos do grupo mais próximo.

4. Índice de Dunn - proposto por J. C. Dunn em 1974 (Dunn 1974), o índice de Dunn mede o rácio entre a distância inter-*clusters* mais pequena e a maior distância intra-*clusters*. Quanto maior for o índice de Dunn, melhor a solução de *clustering*. É definido como:

$$Dunn = \min_{1 \le i \le c} \left\{ \min_{1 \le j \le c, j <> 1} \left\{ \frac{d(c_i, c_j)}{\max_{1 \le k \le c} (d(X_k))} \right\} \right\} \quad (26)$$

Em que:

- $d(c_i, c_j)$ define a distância inter-*cluster* entre o *cluster* X_i e X_j.
- $d(X_k)$ representa a distancia intra-*cluster* do *cluster* (X_k).
- c é o número de *clusters*.

4.3 Sumarização estatística

O processo de criação de sumários pode ser dividido em duas categorias. Sumarização individual, em que o objeto de sumarização é um texto único e sumarização multidocumento. Este tipo de sumarização consiste no resumo de um conjunto de documentos que partilham um assunto relacionado. Os documentos podem ser previamente agrupados com recurso a técnicas de *clustering*. A sumarização multidocumento é a chave para organizar e apresentar resultados de pesquisas mais rapidamente, especialmente quando o objetivo do utilizador é procurar o máximo de informação possível acerca de um tema (Nenkova & McKeown 2011).

Uma abordagem de aprendizagem máquina pode ser aplicada se existir um corpus com os seus respetivos sumários criados previamente de forma manual por especialistas. Esta tarefa pode ser vista como um problema de classificação de duas classes, onde uma frase é ou não é adequada para ser inserida no resumo final. É expectável que o sumarizador crie um modelo através do qual pode identificar os padrões com os quais o resumo foi construído. Assim, quando um novo documento tem que ser resumido, para cada frase são aplicados os padrões aprendidos pelo modelo e cada uma é classificada como adequada ou não adequada a ser introduzida no sumário (Neto et al. 2002). Contudo, este tipo de sumarização tem o inconveniente de necessitar de um grande conjunto de treino.

Atualmente, os métodos de criação automática de resumos utilizam essencialmente técnicas de extração de frases (Nenkova & McKeown 2011). Para cada frase é, por exemplo, quantificada a sua relevância. Nos seguintes subcapítulos serão abordadas algumas formas de calcular o peso das frases de um documento.

4.3.1 Documentos únicos

Várias têm sido as propostas acerca de como pontuar palavras e frases para a criação de sumários, ainda hoje, é um assunto muito discutido. Geralmente, são consideradas como relevantes para uma frase algumas das situações (Chen 2010; Fattah & Ren 2008; Lloret 2009; Neto et al. 2002; Yu & Ren 2009):

- Frequência das palavras no texto – palavras mais frequentes num documento (após eliminação de *stopwords*) tendem a ser importantes.

- Posição da palavra/frase em relação ao texto – as frases que se situam perto do início de um texto ou parágrafo tendem a ser mais importantes do que as restantes.

- Palavras de uma frase que coincidem com as do título tendem a ter algum valor.

- Palavras de sinalização como "concluindo", "importante", entre outras, podem ajudar a perceber a relevância ou irrelevância de frases.

Fattah e Ren (Fattah & Ren 2008) propõem um total de 10 métricas para avaliar a utilidade de uma frase para um sumário. Yen e Ren (Yu & Ren 2009) utilizam algumas dessas métricas. Serão apresentadas aquelas com maior relevância:

1. Posição da frase – assunção de que as primeiras frases de um parágrafo são as mais importantes. Por isso, pontuam as frases de um parágrafo de acordo as suas posições até um máximo de 5 posições. Por exemplo, a primeira recebe 5 pontos, a segunda de 4, e por aí em diante.

2. Semelhança com outras frases – a semelhança com outras frases consiste na sobreposição de vocabulário entre as frases s e p. É calculado da seguinte forma:

$$Score_{f_2} = |\frac{palavras\ em\ s \cap palavras\ em\ p}{palavras\ em\ s \cup palavras\ em\ p}|$$

(27)

3. Grau de semelhança entre a frase e o título – frases com alto grau de semelhança com o título são frases importantes. O seu cálculo é feito da seguinte forma:

$$Score_{f_3} = |\frac{palavras\ em\ s \cap palavras\ \text{no título}}{palavras\ em\ s \cup palavras\ \text{no título}}|$$

(28)

4. Presença de nomes de entidades – frases que contêm nomes de pessoas, organizações e outras entidades são importantes pelo que devem ter grande probabilidade de serem incluídas em resumos. Fattah e Ren(Fattah & Ren 2008) calculam esta medida da seguinte forma:

$$Score_{f_4} = \frac{número\ de\ entidades\ em\ s}{tamanho(s)}$$

(29)

5. Presença de valores numéricos – frases com caracteres numéricos têm algum valor.

$$Score_{f_5} = \frac{número\ de\ dados\ numéricos\ em\ s}{tamanho(s)} \quad (30)$$

6. Tamanho das frases – esta medida é aplicada de forma a penalizar frases que são muito curtas. É expectável que estas não acrescentem muita informação. Para isso, é calculado o tamanho relativo de uma frase, que é feito da seguinte forma:

$$Score_{f_6} = \frac{tamanho(s) * número\ frases\ no\ artigo}{tamanho(artigo)} \quad (31)$$

Em (Neto et al. 2002), SweSum[6](Dalianis 2000) e (Preradovic et al. 2010) podem ser consultadas mais algumas métricas utilizadas para a pontuação e seleção de frases na criação de resumos.

4.3.2 Multidocumento

A sumarização multidocumento consiste na extração de frases de vários textos relacionados e encaixá-las todas entre si de forma a criar um novo documento representativo de todos os restantes. Uma técnica que deve ser considerada devido à sua facilidade de implementação e simplicidades são as nuvens de palavras.

As nuvens de palavras (*word cloud*) são representações gráficas de um documento ou conjunto de documentos em que as palavras são apresentadas no formato de uma nuvem. Essas são representadas tendo em conta a sua importância ou frequência nos documentos. Quantas mais vezes uma palavra estiver presente no corpus, maior aparecerá a palavra na representação gráfica.

As nuvens de palavras podem ter a utilidade de permitir uma rápida identificação dos principais assuntos e/ou intervenientes de um determinado conjunto de documentos (Hollander & Marx 2011). A Figura 12 demonstra como uma nuvem de palavras é representada.

[6] http://swesum.nada.kth.se/index-eng.html

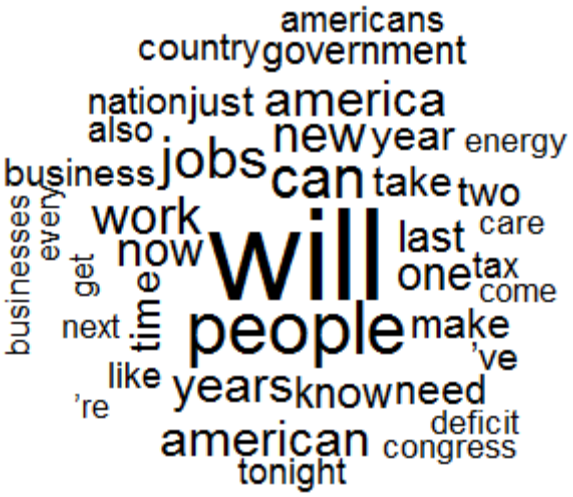


Figura 12 - Nuvem de palavras

Para mais conteúdo relacionado com sumarização multidocumento aconselha-se a leitura de algumas das seguintes fontes: (Chen 2010; Das & Martins 2007; Harabagiu & Lacatusu 2002; Lin & Hovy 2002; Mitkov & Hovy 2005; Nenkova & McKeown 2011; Radev, Blair-Goldensohn, et al. 2001; Radev, Fan, et al. 2001).

4.4 Sumarizadores desenvolvidos

Existem já algumas soluções que realizam a sumarização de notícias para posteriormente serem apresentadas aos utilizadores. O **Columbia Newsblaster**[7] (Evans & Klavans 2005), desenvolvido pela Universidade de Columbia, é um sistema para pesquisa de notícias que faz *crawling* a notícias da web, agrupa as notícias relacionadas em *clusters*, faz o resumo para os documentos do *cluster* e apresenta-os numa interface web.

Em 2001, a universidade do Michigan desenvolveu um sistema de sumarização quer para documentos únicos ou para vários, esse sistema tem o nome de **MEAD**[8](Radev, Blair-Goldensohn, et al. 2001). Utiliza as seguintes formas de pontuar uma frase:

- Pontuação centróide – mede o quão próximo uma frase está face ao centróide que representa o *cluster*.
- Posição – mede o quão longe está uma frase do início. Quanto mais perto está do início do texto, maior pontuação terá.

[7] http://newsblaster.cs.columbia.edu/
[8] http://www.summarization.com/mead/

- Sobreposição com a primeira frase ou título – é feito através da medição do TFxIDF entre a frase S_i e a primeira frase (ou título caso exista).

Também desenvolvido pela Universidade do Michigan é o **WebInEssence** (Radev, Fan, et al. 2001). É mais que um simples sistema de sumarização. E um motor de pesquisa para sumarizar *clusters* de páginas web relacionadas e oferecendo informação mais resumida de forma a ajudar os utilizadores a explorar os resultados de pesquisas mais facilmente (Lloret 2009). Este sistema pode ser dividido em dois módulos, um que recolhe a informação e outro que gera os sumários multidocumento através do MEAD.

O **NeATS** (Lin & Hovy 2002) é um sistema de sumarização multidocumento adaptado para artigos noticiosos e foi desenvolvido em 2001 pela Universidade do Sul da Califórnia. Para gerar os sumários utiliza técnicas relacionadas com a posição das frases, frequência do termos, assinaturas de tópicos e *clustering* de termos.

Outro sistema de geração de sumários é o **GISTexter** (Harabagiu & Lacatusu 2002). Desenvolvido em 2002 e produz sumários abstratos e extrativos para multidocumentos ou documentos únicos. Utiliza técnicas de extração de informação orientadas a modelos (*templates*). O sistema funciona de forma diferente para os dois casos (Lloret 2009). Para documentos únicos as frases mais relevantes são extraídas através de regras aprendidas a partir de sumários escritos por humanos. Cada sumário tem um limite de 100 palavras. Para sumarização multidocumento, o sistema utiliza *templates* criados quer a partir de um conjunto anterior (se o assunto é bem conhecido) ou por geração *ad-hoc*[9] (caso o tópico seja desconhecido).

Em 2007 surgiu o **NetSum** (Svore et al. 2007), desenvolvido pela Microsoft, este foca-se apenas na sumarização de documentos únicos. É um sistema que utiliza técnicas de aprendizagem máquina e faz extratos de notícias através de redes neuronais. O sistema gera uma lista de frases ordenadas de acordo um *ranking* de importância, para isso, é utilizado o algoritmo RankNet (Burges et al. 2005).

[9] Tarefa em que o utilizador especifica as suas necessidades de informação através de queries que iniciam a pesquisa por documentos provavelmente relevantes para o utilizador.

4.5 Avaliação sumarização

A avaliação da geração de sumários é um problema com difícil resolução uma vez que não existe um resumo ideal para um documento ou conjunto de documentos. Das e Martins (Das & Martins 2007) referem, com base em outras investigações, que não existe muito entendimento quanto à forma de criação de sumários (Lloret 2009), bem como à forma de se avaliar um sumário. O sumário ideal depende do utilizador, propósito e género (Chen 2010). Por isso, é que existem métricas de avaliação bastante dispersas (Das & Martins 2007; Mitkov & Hovy 2005; Nenkova & McKeown 2011).

Isto gera um problema crítico (Das & Martins 2007), a inexistência de um *standard* consensual na avaliação de sumários e, respetiva dificuldade em comparar diferentes sistemas. Têm sido propostas várias categorias de métodos de avaliações, incluindo análise humana, avaliação da similaridade, avaliação baseada em tarefas e questionários (Chen 2010).

A avaliação com base em análise humana é feita com base no convite a pessoas para ajuizar os sumários gerados de acordo várias escalas de avaliação. Ainda que este método seja dispendioso a nível de tempo e trabalho é o mais direto e eficaz caso o teste seja preparado corretamente (Chen 2010; Das & Martins 2007). A este tipo de avaliação dá-se o nome de avaliação extrínseca.

A maior parte das avaliações de sistemas de sumarização são intrínsecas. Os avaliadores criam um conjunto de sumários ideais e comparam com o *output* gerado pelo sumarizador. A sobreposição do conteúdo nos dois sumários é uma das formas de avaliação mais utilizadas (Mitkov & Hovy 2005).

4.5.1 ROUGE

ROUGE[10] ou *Recall-Oriented Understudy for Gisting Evaluation* é um conjunto de métricas propostas por Lin (Lin 2004) que se tornaram bastante utilizadas na avaliação automática de sumários (Das & Martins 2007; Nenkova & McKeown 2011) devido à sua forte orientação à unidade de medida *recall*. É bastante utilizada como métrica de qualidade na seleção de conteúdo em *papers* de investigação devido ao seu reduzido custo e rapidez (Nenkova & McKeown 2011).

[10] http://www.berouge.com/Pages/DownloadROUGE.aspx

Neste conjunto estão incluídas as seguintes cinco métricas:

1. ROUGE-N(Lin 2004)(Lin 2004)(Lin 2004)(Lin 2004)(Lin 2004)(Lin 2004)(Lin 2004)(Lin 2004)(Lin 2004)(Lin 2004)(Lin 2004) – baseia-se em estatísticas de coocorrências de n-gramas.

2. ROUGE-L – Aplica o conceito da *longest common subsequences* (LCS) (Bergroth et al. 2000). A ideia por de trás é que quanto maior o LCS entre as frases de dois sumários, mais semelhantes elas são.

3. ROUGE-W – é uma refinação da ROUGE-L que inclui pesos que penalizam correspondências de subsequências que não são consecutivas.

4. ROUGE-S – semelhante ao ROUGE-N mas com base em *skip-bigrams*. Os *skip-bigrams* são qualquer par de palavras na sua respetiva ordem frásica, permitindo assim intervalos entre elas. Por exemplo, a frase “Police killed the gunman” tem os seguintes *skip-bigrams*: “police killed”, “police the”, “police gunman”, “killed the”, “killed gunman”, “the gunman”.

5. ROUGE-SU – tenta suprimir algumas lacunas do ROUGE-S introduzindo uni-gramas.

5 Abordagem ao problema

Após o processo de fundamentação teórica das tarefas a serem realizadas face ao problema proposto, este capítulo tem como objetivo descrever como foi abordado o problema.

5.1 Apresentação do problema

O presente trabalho propõe-se a trabalhar com notícias desportivas em quatro fases diferentes que seguem uma cronologia específica. Essa cronologia é muito importante pois permitirá ir de encontro aos requisitos definidos no projeto que vai irá ser desenvolvido.

A primeira fase consistiu na construção e pré-processamento do corpus. Cada documento é uma notícia desportiva recolhida de várias fontes *online*.

Seguiu-se a fase de classificação das notícias. Depois foi construído um modelo representativo dos documentos do corpus. Este foi depois aplicado aos documentos de teste. Para terminar foi feita a avaliação do processo.

Um dos requisitos do projeto é a agregação de notícias de várias fontes que se relacionem com um evento ou acontecimento. Para cada categoria foram testadas técnicas de *clustering* na tentativa de separar os documentos consoante os requisitos definidos. Por eventos entenda-se, por exemplo: um grupo de notícias que referenciem a contratação de um jogador de futebol para um determinado clube

Realizar a classificação em primeiro lugar permitiu uma primeira divisão dos dados, reduzindo o número de documentos a serem considerados na construção dos *clusters*. À semelhança da classificação foi também realizada uma avaliação a este processo. Uma estatística e outra manual.

O passo final foi novamente de encontro a um dos requisitos do projeto em que este relatório se insere. A criação de *tags* que identifiquem os principais assuntos e/ou intervenientes do evento de um *cluster*. Este assunto foi abordado de forma muito superficial não sendo o principal foco do trabalho apresentado. A próxima esquematiza o processo referido.

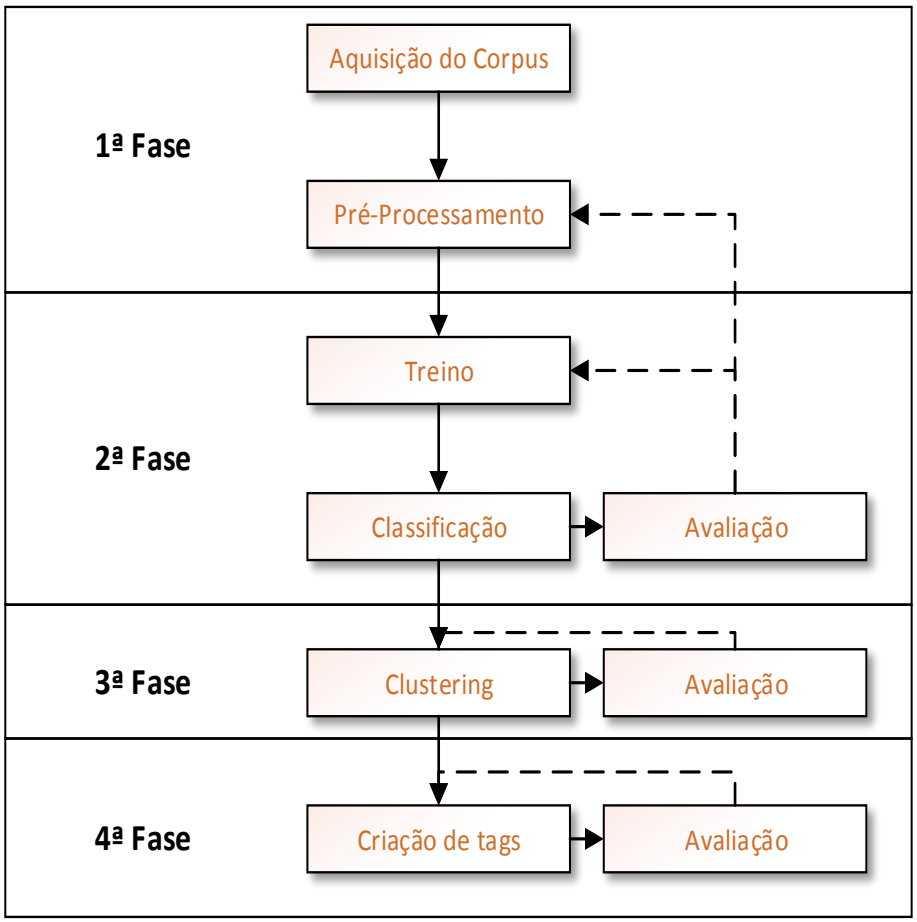


Figura 13 - Fluxo de desenvolvimento

É importante perceber que as etapas acima referidas tiveram abordagens à base da tentativa e erro de forma a tentar obter os melhores resultados possíveis. Nos próximos subcapítulos será descrita cada etapa com maior detalhe.

5.1.1 Aquisição do corpus

Esta etapa consistiu na recolha de várias notícias desportivas de forma a construir um corpus que represente suficientemente as várias categorias definidas. Foi uma das tarefas mais difíceis, pois necessitou de bastante trabalho manual, principalmente na classificação manual de todos os documentos.

O corpus foi construído através da recolha de notícias de cinco diferentes fontes exclusivamente portuguesas, quase todas especializadas em desporto (exceção é o Diário de Notícias) e das que mais destaque jornalístico têm em Portugal quando o assunto é desporto.

- Jornal Abola – www.abola.pt

- Jornal Record – www.record.xl.pt
- Jornal OJogo – www.ojogo.pt
- MaisFutebol – www.maisfutebol.iol.pt
- Diário de Notícias (secção de deporto) – www.dn.pt/desporto

Foi desenvolvido leitor de RSS *feeds* feito para armazenar todas as notícias.

5.1.2 Pré-processamento

A fase de pré-processamento consistiu na análise dos documentos e estruturá-los da melhor forma para a fase de treino e teste. Este processo foi decomposto em vários módulos de forma a agilizar e tornar o processo mais eficiente. Optou-se pela representação *bag-of-words* e o TFxIDF como como medida de pesagem da importância das palavras. A escolha justifica-se por serem representações e métricas que são frequentemente aceites pela comunidades como escolhas equilibradas e eficazes. A Figura 14 apresenta as etapas da fase de pré-processamento. A Figura 14 demonstra o principal fluxo de pré-processamento utilizado.

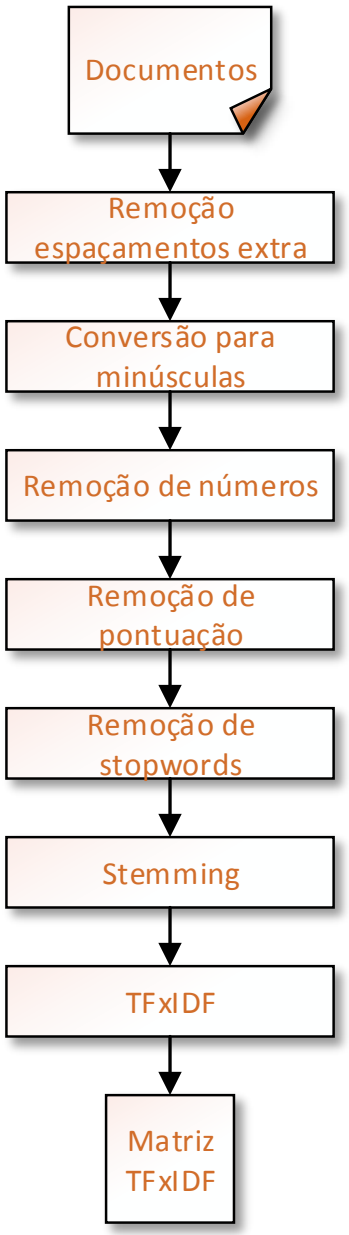


Figura 14 - Fases do pré-processamento

5.1.3 Treino, teste e avaliação

Após realizado o pré-processamento, o passo seguinte foi decidir quais os classificadores a utilizar. Para a realização deste projeto foram utilizados dois algoritmos de classificação. Para cada um foi feita a respetiva avaliação de forma a se perceber qual o melhor algoritmo. O classificador que apresentou melhores resultados será integrado no projeto que será desenvolvido futuramente. Após consideração dos algoritmos apresentados em 3.2, os algoritmos escolhidos foram o *k-nearest neighbors* e o *support vector machine*.

A principal justificação para a escolhas destes dois algoritmos é por serem reconhecidos pela comunidade como dois classificadores eficientes para texto. O levantamento do estado da arte permitiu perceber que são dois dos algoritmos mais utilizados. A avaliação dos resultados da classificação foi feita de com recurso à F-Measure.

Foram definidas as seguintes categorias para os documentos:

- Pré-Jogo - notícias relacionadas com partidas de futebol a serem realizadas num curto período de tempo. Inclui declarações prévias à partida, conferências de imprensa, jogadores em dúvidas para o jogo, convocatórias, nomeações de árbitros, etc.

- Resultados e Pós-Jogo – notícias relacionadas com partidas recentemente disputadas. Nesta categoria incluem-se principalmente comentários e reações às partidas, conferências de imprensa pós-jogo, listagem de resultados do dia, entre outros.

- Mercado – notícias relacionadas com rumores de transferências de jogadores, dispensas, renovações de contrato, entre outros.

- Castigos – notícias referentes a castigos ou suspensões de atletas, equipas, dirigentes, etc.

- Lesões – notícias que se referem a lesões de atletas.

- Treinadores – notícias relacionadas com a contratação e despedimento de treinadores.

- Óbitos – notícias relacionadas com falecimentos de pessoas ligadas ao desporto.

- Geral – é uma categoria generalista para as notícias onde nenhuma das categorias anteriores se adeque.

5.1.4 Clustering e avaliação

O objetivo desta etapa foi, para uma determinada categoria, dividi-la em grupos de eventos relacionados. Após análise ao estudo feito no capítulo 4.1, conclui-se que a utilização dos algoritmos particionais tradicionais não era o caminho a seguir. Estes requerem a definição

prévia do número *clusters*, o que não é possível determinar dado o problema em causa. A Figura 15 demonstra o processo.

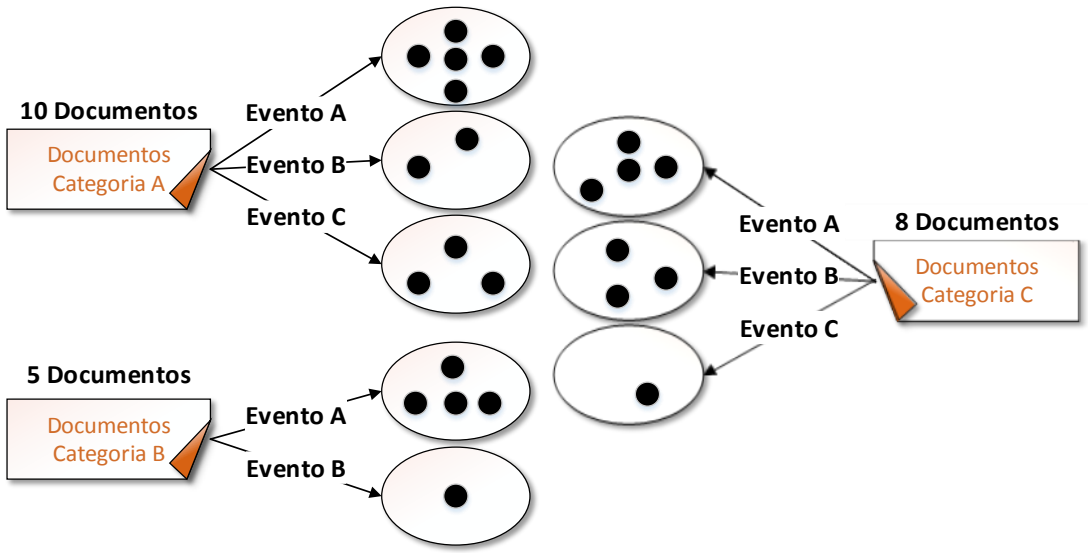


Figura 15 - Agrupamentos por eventos para cada categoria

Este processo foi realizado com recurso a algoritmos hierárquicos e ao *k-means* adaptável. Este último apesar de ter por base o tradicional *k-means* apresenta algumas condições que justificaram a sua escolha (capítulo 4.1.2.2). Foi realizada uma avaliação aos dois algoritmos e tomadas as devidas conclusões.

Um dos maiores problemas reside no aspeto temporal das notícias relacionadas com um evento. O seguinte exemplo reflete essa questão. Considere-se a transferência de um jogador de futebol para um clube. É muito frequente que surjam várias notícias acerca dessa transação. É normal a criação de muitos rumores em torno desse acontecimento. Por exemplo, um dia vai para esse clube, passados alguns dias afinal já não vai. Passado mais alguns dias o negócio foi realmente concluído mas pelo meio existiram outros clubes interessados no atleta. Tudo que rodeie esta transação desde os seus primeiros rumores até à confirmação oficial da mesma devem estar agrupadas no mesmo *cluster*.

Já notícias acerca da mesmo atleta mas que se refiram, por exemplo, à sua prestação menos conseguida numa determinada partida, ainda que estejam relacionadas com o atleta, são eventos diferentes. Pelo que devem ser agrupadas num *cluster* que já exista ou então num novo *cluster*.

Sempre que surge uma nova notícia, deve ser verificado se existem *clusters* relacionados com esse evento, em caso afirmativo esta é adicionada ao *cluster*, caso contrário, um novo *cluster* deve ser criado. Contudo, este processo não foi alvo de avaliação no presente trabalho.

É importante referir que para cada algoritmo o processo foi executado duas vezes. A primeira com as notícias com o seu texto integral mas com o devido pré-processamento (ver capítulo 5.1.2). Na segunda considerando apenas nomes de entidades (clubes, atletas, dirigentes, entre outros). Contudo existe uma limitação na segunda opção. Para a deteção das entidades preservam-se apenas as palavras com a primeira letra maiúscula. À exceção dos inícios das frases, estas geralmente representam entidade, até porque as notícias são geralmente escritas por profissionais com formação académica superior. Como tal, é esperado que exista esse cuidado sempre que existe uma referência a nomes de pessoas, clubes, competições, entre outros. Contudo existe sempre alguma possibilidade de existirem palavras não desejadas.

5.1.5 Criação de tags e avaliação

Um dos principais objetivos do projeto é que os utilizadores possam visualizar apenas a informação que é relevante para um determinado fim. Para isso, os utilizadores terão de definir previamente um conjunto de filtros. Assim, cada grupo de notícias deverá ter um conjunto de palavras (*tags*) que permitam identificar o assunto a que se referem e os intervenientes mais diretos. Assim será possível fazer uma correspondência entre os filtros do utilizador e as *tags* que identificam um *cluster*.

Para a criação das *tags* foi feito um estudo acerca da viabilidade da utilização de nuvens de palavras. Obviamente apenas serão tidas em conta as k palavras mais importantes ou mais frequentes. A avaliação da viabilidade deste processo foi feita através de avaliações manuais (humanas).

6 Classificação automática

Este capítulo irá apresentar com maior detalhe a metodologia utilizada a resolução do problema proposto para a classificação automática de notícias. Serão apresentados os respetivos resultados e feita a conclusão e análise dos mesmos.

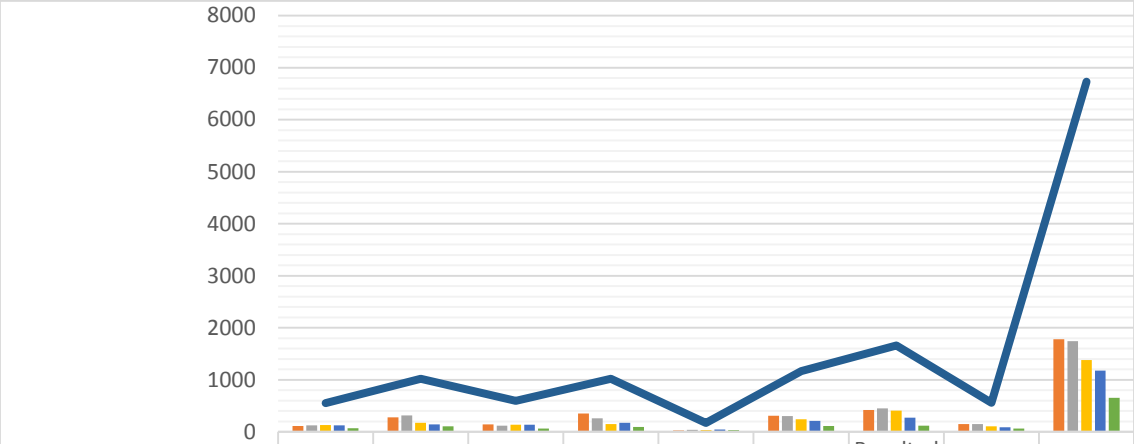
6.1 Constituição do corpus

Como referido no capítulo 5.1.1 os documentos obtidos para o desenvolvimento deste trabalho são artigos de notícias desportivas de algumas das fontes mais consultadas em Portugal. A escolha da utilização de documentos apenas em português deveu-se ao facto de para além de haver um maior domínio da língua, o projeto a ser desenvolvido, numa primeira fase, ter como público-alvo os países lusófonos. Atualmente existem já ferramentas de pré-processamento de texto para português.

Os artigos foram maioritariamente recolhidos durante o período de 07 de Julho de 2014 e 18 de Agosto de 2014, inclusive. Ainda assim, para algumas categorias o número de documentos era claramente insuficiente, pelo que se procedeu à pesquisa e recolha manual de algumas notícias. Foi definido o limite máximo de 1 ano desde o início da recolha automática de artigos. Assim, para as categorias de "Óbitos", "Lesões", "Castigos" e "Treinadores" foram recolhidas mais notícias até à data aproximada de Julho de 2013. A pesquisa foi feita apenas nas cinco fontes referidas anteriormente.

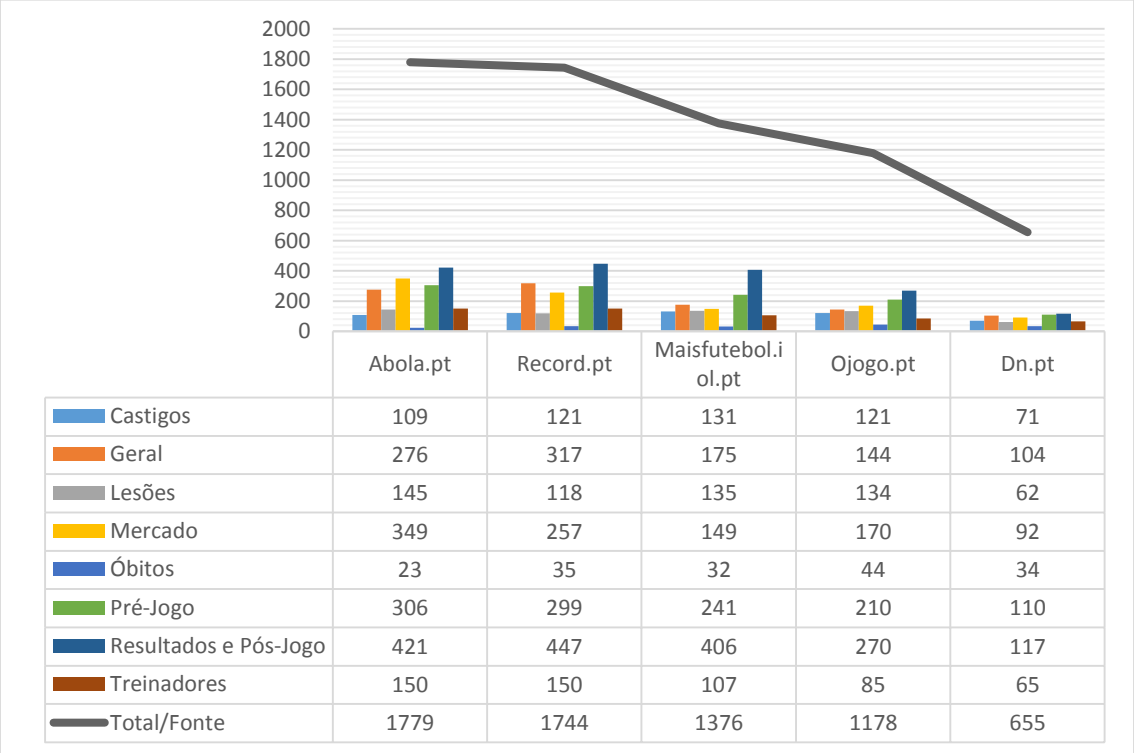
Foram armazenados um total de 6732 artigos, sendo todos eles classificados manualmente.

Nos gráficos seguintes pode-se observar alguns detalhes acerca da distribuição dos artigos no corpus. Vai ser possível observar que entre algumas categorias existe um grande desequilíbrio no número de artigos, o que pode criar problemas de *overfitting*. Foram utilizadas amostragens estratificadas como forma de tentar evitar esse problema.



	Castigos	Geral	Lesões	Mercado	Óbitos	Pré-Jogo	Resultados e Pós-Jogo	Treinadores	Total/Fonte
Abola.pt	109	276	145	349	23	306	421	150	1779
Record.pt	121	317	118	257	35	299	447	150	1744
Maisfutebol.iol.pt	131	175	135	149	32	241	406	107	1376
Ojogo.pt	121	144	134	170	44	210	270	85	1178
Dn.pt	71	104	62	92	34	110	117	65	655
Total/Categoria	553	1016	594	1017	168	1166	1661	557	6732

Figura 16 - Artigos por categoria



	Abola.pt	Record.pt	Maisfutebol.iol.pt	Ojogo.pt	Dn.pt
Castigos	109	121	131	121	71
Geral	276	317	175	144	104
Lesões	145	118	135	134	62
Mercado	349	257	149	170	92
Óbitos	23	35	32	44	34
Pré-Jogo	306	299	241	210	110
Resultados e Pós-Jogo	421	447	406	270	117
Treinadores	150	150	107	85	65
Total/Fonte	1779	1744	1376	1178	655

Figura 17 - Artigos por fonte

6.2 Plano experimental

O processo de experimentação começou pelo pré-processamento. É uma tarefa importante, pois dados com menor ruído tendem a favorecer a performance e eficácia dos classificadores. No capítulo 5.1.2 podem ser consultadas todas as fases de pré-processamento adotadas na realização deste trabalho.

Após construída a matriz TFxIDF, houve a necessidade de remover os termos com baixa frequência pois a matriz construída tinha um número exageradamente grande de termos, 18251 no total. A remoção dos termos foi feita através da indicação do *document frequency threshold* (Escudeiro 2012), em que são eliminados os termos cujo valor de IDF está acima desse valor. Na linguagem R, é um valor que se situa entre 0 e 1. Na tabela seguinte podem ser consultados os vários níveis limite para a remoção de termos e os respetivo número de termos da matriz. Por 1 entenda-se a matriz com todos os termos.

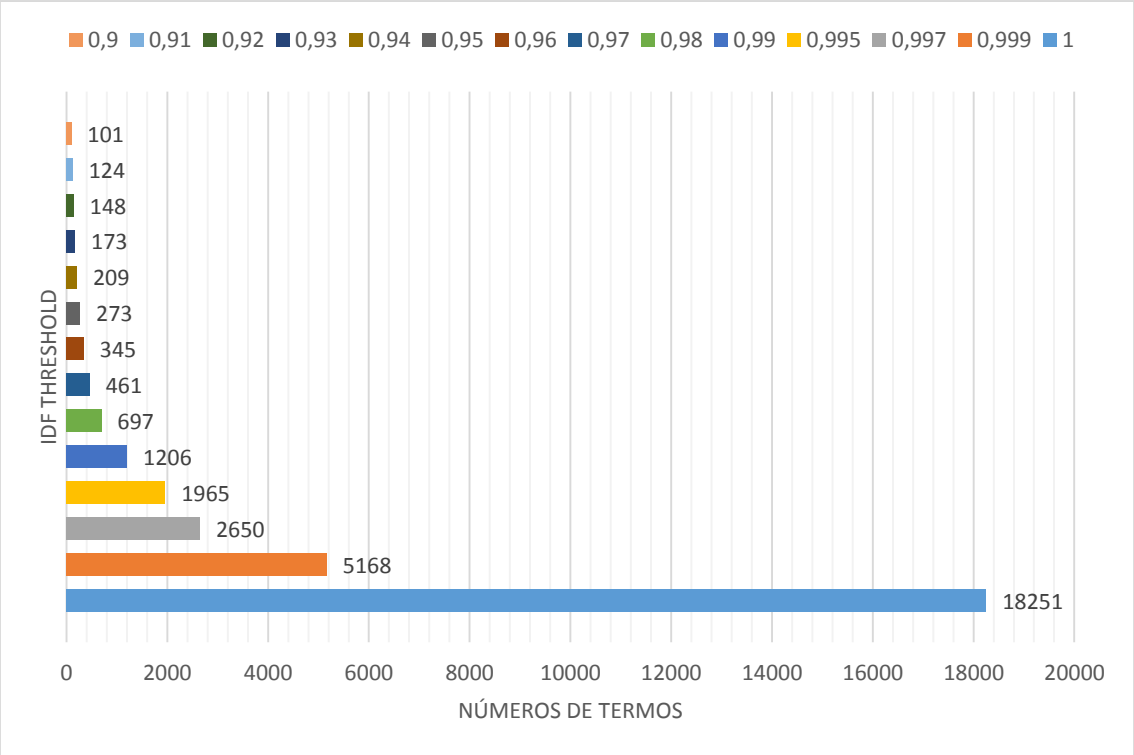


Figura 18 – Termos por IDF *threshold*

Para a construção dos modelos de classificação, foram utilizados dois dos algoritmos mais conhecidos e utilizados, o SVM e o KNN. Para o KNN foram testados três valores para $K = 3, 7 \; e \; 11$.

Foi utilizada a técnica de validação cruzada de forma a ser possível avaliar a generalização de um modelo. O valor de $K = 10$ é referido como uma escolha fiável (Alves 2010).

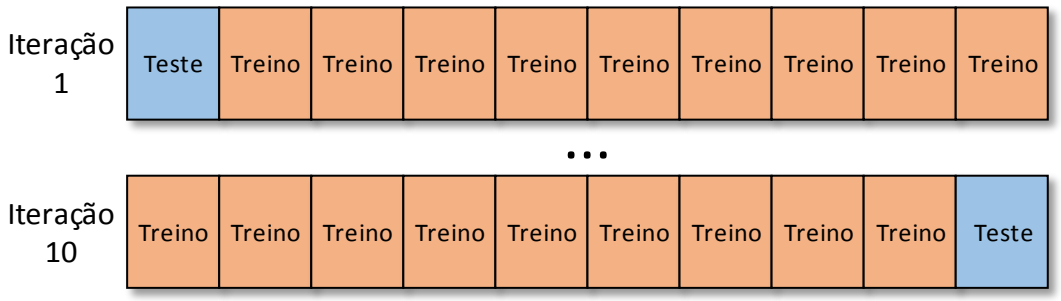


Figura 19 - 10-Fold cross validation

Contudo, e em virtude do número de categorias ser bastante desequilibrado em alguns casos e, para que em todas as 10 iterações da validação cruzada, existam dados de todas as categorias nos conjuntos de treino e teste, foi optada por fazer uma amostragem estratificada dos dados.

Resumidamente, a amostragem estratificada deve ser aplicada para quando numa população existem subgrupos com diferenças significativas no número de elementos de cada um. Garante que todos esses subgrupos (estratos) estarão representados na amostra final de forma proporcional ao seu número de elementos na amostra total. Exemplificando, numa amostra que reflete o número pessoas por grupo etário existem 700 adultos, 200 crianças e 100 idosos. Uma amostra estratificada de 10% da amostra seria 70 adultos, 20 crianças e 10 idosos.

Assim, os dados foram organizados da seguinte forma:

- Cada um dos 10 conjuntos terá tamanho igual ou praticamente igual de documentos. Na maior parte dos casos, os conjuntos de teste possuem sempre um total de 673 documentos, o que corresponde aproximadamente a 10% do número total de artigos presentes no corpus.

- Cada um dos conjuntos é criado de forma a ter uma porção de dados de cada categoria de cada fonte. Exemplificando, na primeira iteração, o conjunto de teste vai conter os primeiros 10% de documento de cada uma das categorias para cada uma das fontes. O conjunto de treino é construído com os restantes 90% de cada categoria de cada fonte. Obviamente, a cada iteração o intervalo de dados de teste é diferente, no caso da segunda iteração, será considerado o intervalo de 11% a 20% dos dados.

- De forma a facilitar este processo, os dados foram ordenados primeiro por fonte e dentro de cada fonte, ordenados por categoria.

Assim, para a fase da construção dos modelos, foram considerados um total de 520 modelos diferentes. Para o algoritmo SVM foi um total de 130 matrizes a serem consideradas para a construção do modelo pois foi aplicada a validação cruzada para cada matriz com diferentes *IDF threshold*. No total foram criadas 13 matrizes, cada uma com um diferente número de termos, totalizando 130 matrizes ($13 * 10 = 130$). A mesma situação aplica-se também ao algoritmo KNN. Como este algoritmo será avaliado com três diferentes valores de K, foram criados 390 modelos.

Partindo da base criada anteriormente, foram gerados todos os modelos de classificação considerados. Para permitir uma melhor avaliação e análise aos dados obtidos foi criada uma matriz de confusão para cada modelo. Esta permite avaliar a qualidade de um modelo.

	Reference							
Prediction	Castigos	Geral	Lesões	Mercado	Óbitos	Pré-Jogo	Resultados e Pós-Jogo	Treinadores
Castigos	34	3	0	0	0	0	1	0
Geral	12	58	3	13	6	4	6	8
Lesões	1	0	44	0	0	3	1	0
Mercado	1	11	2	85	0	1	0	0
Óbitos	0	0	0	1	10	0	0	1
Pré-Jogo	4	17	4	2	0	102	17	4
Resultados e Pós-Jogo	3	13	6	1	0	7	142	4
Treinadores	0	0	0	0	0	0	0	38

Figura 20 - Matriz de confusão de um modelo no R

A partir da matriz de confusão é possível avaliar cada classe. Para tal as matrizes tiveram que ser convertidas para uma série de matrizes binárias, uma para cada classe. Consideram como classes c e $\bar{c}$. Para cada modelo, foram criadas um total de oito, uma para cada categoria. As várias medidas globais do modelo, podem depois ser calculadas a partir da média ponderada de cada uma das medidas das categorias.

$Castigos

	c	~c
c	34	21
~c	4	614

Figura 21 - Matriz de confusão binária para a categoria castigos

6.3 Apresentação e análise aos resultados

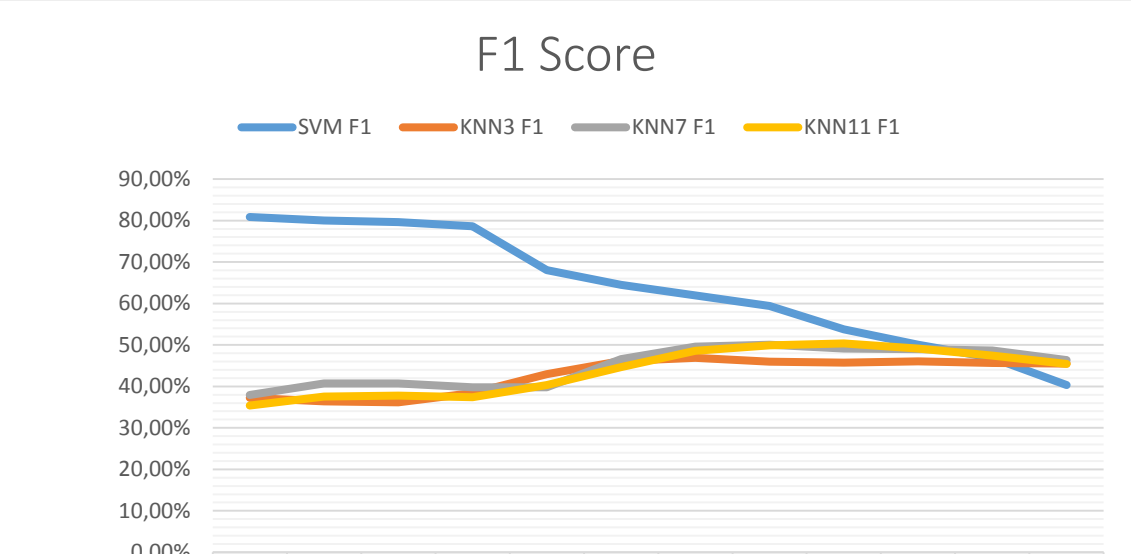
Esta secção tem como finalidade a apresentação dos resultados obtidos e a discussão dos mesmos. Aqui serão abordados vários aspetos dos dados, por exemplo, a performance por

classificador, a performance obtida para cada uma das classes, explorando aspetos como a relação entre a capacidade de uma correta classificação e o número de documentos por categoria. Todos os resultados apresentados incluem a aplicação da técnica de validação cruzada, tendo sido para isso calculada a média ponderada para cada modelo.

Os resultados apresentados incidirão principalmente na *F1 Score*, que como referido no capítulo 3.4.1 é uma medida que agrega a precisão e a abrangência.

6.3.1 Classificadores e IDF threshold

A forma como os classificadores constroem os seus modelos varia entre cada um. Cada um é implementado sob um algoritmo diferente. Assim, é expectável que os resultados entre si variem.



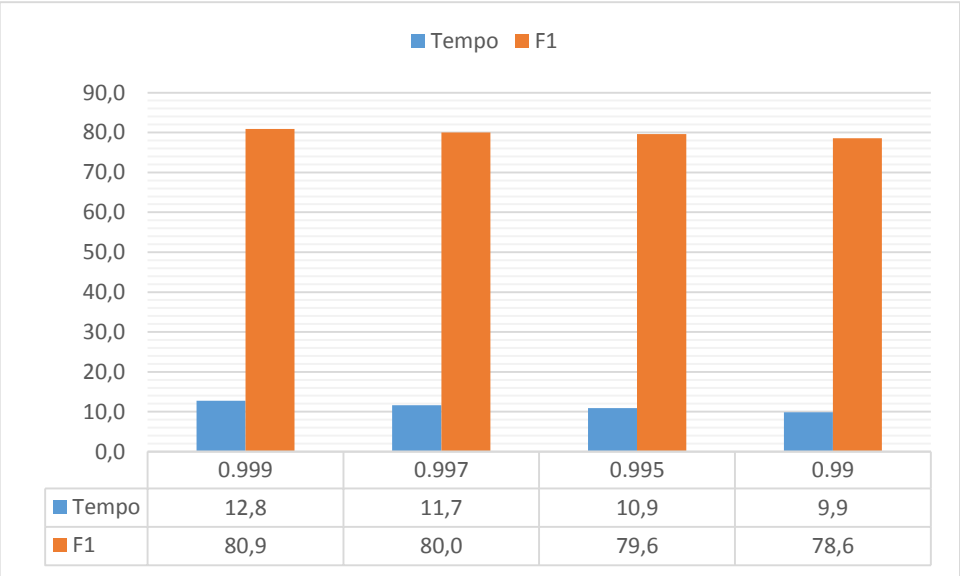
	0.999	0.997	0.995	0.99	0.98	0.97	0.96	0.95	0.94	0.93	0.92	0.91
SVM F1	80,90%	80,0%	79,6%	78,6%	68,1%	64,5%	62,0%	59,4%	53,8%	50,1%	46,9%	40,3%
KNN3 F1	37,2%	36,4%	36,2%	38,4%	42,9%	46,1%	46,9%	45,9%	45,8%	46,1%	45,7%	45,5%
KNN7 F1	37,9%	40,7%	40,7%	39,8%	39,8%	46,6%	49,6%	50,0%	49,1%	49,0%	48,7%	46,3%
KNN11 F1	35,4%	37,6%	37,8%	37,4%	40,3%	44,6%	48,6%	49,9%	50,4%	49,1%	47,4%	45,4%

Figura 22 - *F1 Score* de cada classificador por IDF *threshold*

O gráfico da Figura 22 apresenta a relação entre cada classificador e os diferentes valores do *IDF threshold*. Numa primeira análise, enquanto o IDF *threshold* é superior a 0,93, é notório que o SVM é de longe o classificador que apresenta melhores resultados. Contudo, a partir desses valores e, apesar de os dois algoritmos se equipararem, a performance apresentada é bastante baixa.

A escolha parece óbvia quanto ao melhor algoritmo a ser utilizado. As três variantes do algoritmo KNN apresentam resultados bastante medíocres, ultrapassando apenas em escassas vezes um valor de F1 superior a 50%. Ainda assim, é curioso observar que em contraste com o SVM, no KNN à medida que o número de termos diminui, a performance dos algoritmos KNN tende a aumentar, ainda que de forma muito ligeira. Isso acontece, provavelmente, devido ao efeito da "*curse of dimensionality*" (Donoho 2000). Em poucas palavras, a "*curse of dimensionality*" refere que à medida que as dimensões (termos) aumentam, o volume do espaço dimensional também aumenta, pelo que a pesquisa de pontos de dados no espaço torna-se mais difícil pois os dados encontram-se mais dispersos. Daí a qualidade do algoritmo aumentar à medida que o número de termos (dimensões) diminui.

O classificador SVM apesenta uma performance bastante aceitável entre os quatro valores mais elevados do IDF *threshold*, havendo apenas um ligeiro decréscimo à medida que esse valor diminui até 0,99. A partir daí, verifica-se uma quebra acentuada da performance do classificador.



	0.999	0.997	0.995	0.99
Tempo	12,8	11,7	10,9	9,9
F1	80,9	80,0	79,6	78,6

Figura 23 - Tempo de construção dos modelos vs *F1 Score*

No gráfico anterior, pode-se comparar o tempo de construção do modelo e a performance de cada um dos modelos em que o IDF *threshold* ainda apresenta uma performance aceitável. Para eventuais reaprendizagens do modelo, de forma a criar um modelo mais atualizado, justifica-se a utilização do modelo com IDF *threshold* de 0,99. Comparativamente com o primeiro modelo, existe apenas uma diferença de 2,3% na qualidade do modelo, porém, esse modelo demora quase mais 3 minutos a ser construído. Uma vez que o projeto a ser desenvolvido tem

o objetivo de devolver informação aos utilizadores o mais rapidamente possível, a utilização do modelo cujo valor de IDF *Threshold* é 0,99 é aceitável.

A partir deste momento, todos os dados apresentados serão referentes ao modelo construído com o algoritmo SVM e o IDF *threshold* de 0,99.

6.3.2 Categorias

É importante perceber o comportamento da classificação em cada classe de forma a ser possível melhorar os modelos futuramente. A taxa de erro pode ser um bom indicador relativamente à capacidade do modelo classificar corretamente novos documentos. A Figura 24 apresenta a taxa de erro por cada categoria.

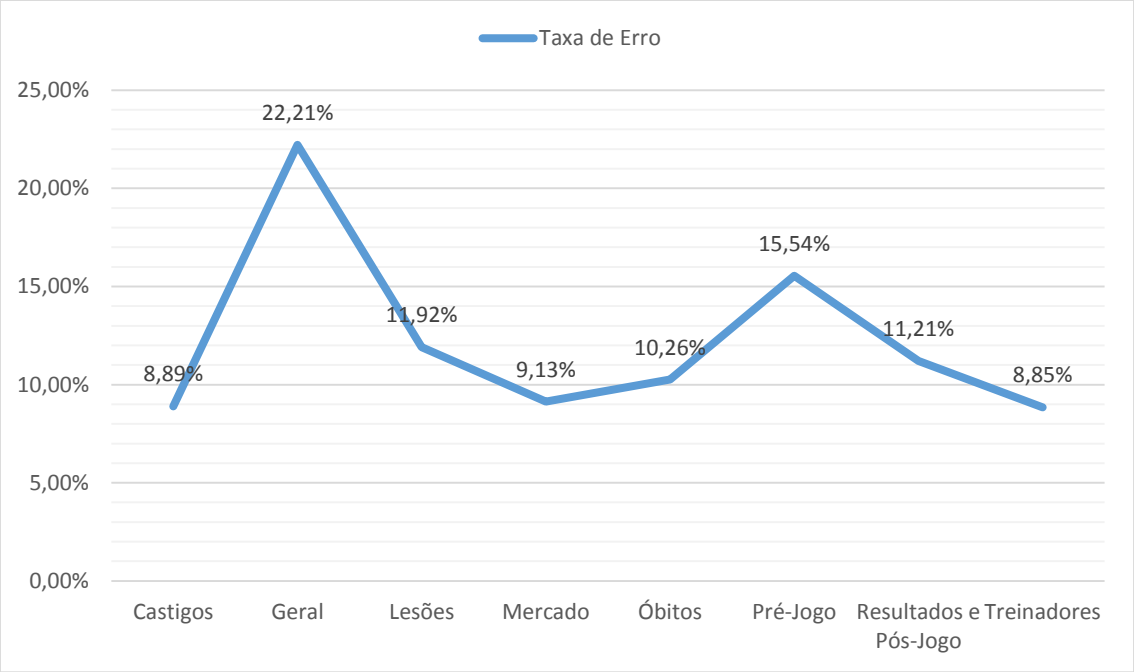
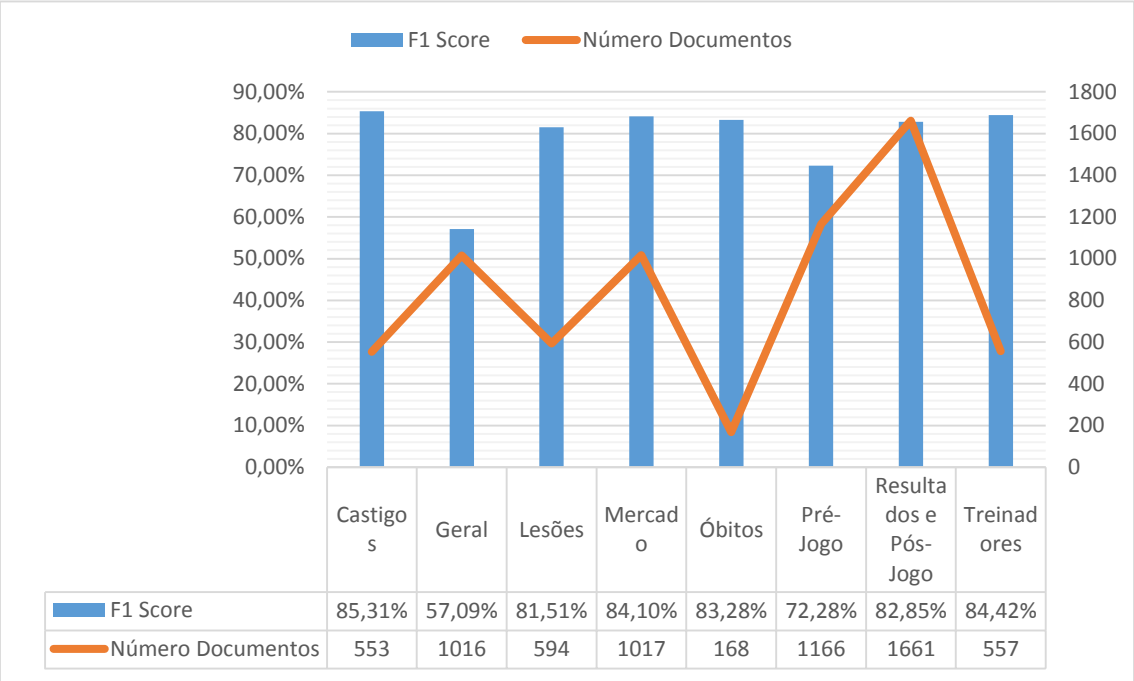


Figura 24 - Taxa de erro por categoria

A taxa de erro apresenta valores aceitáveis para a grande parte das categorias. A maioria das classes apresentam uma taxa de erro a rondar os 10%, valores que se podem considerar aceitáveis. A exceção são as classes “Geral” e “Pré-Jogo” que apresentam taxas de erro de aproximadamente 22% e 16%, respetivamente. Contudo, estes valores seriam de esperar uma vez que são, provavelmente, as duas categorias mais heterogéneas.

Um bom indicador a ter em conta na avaliação da qualidade do modelo escolhido é a relação entre o número de documentos por categoria e a sua performance em cada uma das categorias. Permite perceber se existe influência do número de documentos na classificação

final. A Figura 25 apresenta a relação entre a qualidade da classificação em cada categoria e o número de documentos em cada uma das categorias.



	Castigos	Geral	Lesões	Mercado	Óbitos	Pré-Jogo	Resultados e Pós-Jogo	Treinadores
F1 Score	85,31%	57,09%	81,51%	84,10%	83,28%	72,28%	82,85%	84,42%
Número Documentos	553	1016	594	1017	168	1166	1661	557

Figura 25 – Relação entre *F1 Score* e número de documentos por categoria

Do gráfico apresentado não é possível tirar grandes conclusões da relação entre a qualidade de classificação e o número de documentos. Pode-se verificar que existem categorias com um *F1 Score* superior a 80% tendo 1017 documentos (Mercado) ou apenas 168 documentos (Óbitos).

Pode-se verificar que a categoria "Geral" tem um *F1 Score* muito medíocre, contudo, dada a natureza da classe seria algo de esperar.

7 Clustering

Este capítulo irá apresentar com maior detalhe a metodologia utilizada para no desenvolvimento da parte respeitante à agregação de notícias. Serão apresentados os resultados obtidos e feita a conclusão e análise aos mesmos.

Os dados apresentados neste capítulo terão como por base o corpus apresentado no capítulo anterior. A diferença é que os documentos já se encontram divididos por categorias. Assim evita-se a existência de documentos de categorias diferentes num mesmo *cluster*.

7.1 Algoritmos hierárquicos

Após o levantamento do estado da arte (capítulo 4.1) conclui-se que os algoritmos hierárquicos poderiam ser uma das soluções para o problema proposto. A assunção por de trás dessa conclusão deve-se ao processo dos dados serem divididos iterativamente, formando um dendrograma, e que eventualmente será atingido um nível de corte em que a árvore gerada conterá em cada *cluster* os elementos relacionados com um determinado evento.

7.1.1 Plano experimental

Na implementação deste processo para a geração dos dendrograma foi utilizado o método de *clustering* hierárquico de Ward. O facto de privilegiar a variância mínima a cada iteração pode ter alguns benefícios. Uma vez que o objetivo final é ter um conjunto de *clusters* cuja variância intra-*cluster* seja a menor possível, optou-se por testar e avaliar este método de *clustering*.

O processo de experimentação começou com a criação para todas as categorias dos respetivos dendrogramas. Estas terminam a sua execução quando cada documento é um *cluster* de si próprio, o que não traz benefício nenhum ao processo de agregar documentos por eventos. Assim as árvores necessitam de ser “podadas” de forma a poder ser feita uma avaliação com o mínimo de qualidade.

Assim, foram criados vários dendrogramas, cada um com um nível diferente de corte. Em cada nível é possível obter vários *clusters* e os seus respetivos documentos constituintes. Porém, foram aplicadas algumas condicionantes de forma a reduzir o número de níveis. Existem alguns

níveis de corte que dada a quantidade de *clusters* que contêm, não necessitam de ser avaliados. As condicionantes são:

- No mínimo devem existir cinco *clusters*. Uma vez o corpus disponível contém um número consideravelmente elevado de documentos, a avaliação de um número reduzido de *clusters* não traz qualquer benefício. Este valor foi selecionado de forma a estar em concordância com o algoritmo apresentado no próximo capítulo.

- O número de *clusters* tem que ser no máximo de $1/3$ do número total de documentos por categoria. Esta condição foi implementada de forma a não existir um número exageradamente elevado de *clusters*. Uma vez que três das cinco fontes abordam com bastante frequências os mesmos assuntos. É expectável que para um evento existam pelo menos três documentos. Assim, todos os *clusters* terão que ter no mínimo um total de três documentos.

Um dos objetivos foi tentar perceber se era possível aplicar um nível de corte comum a todas as classes. Em caso negativo, perceber a possibilidade de o fazer para cada uma das categorias e verificar a sua aplicabilidade para data-streams. Como tal, para cada categoria foi selecionado o nível de corte que apresentou os *clusters* com melhor qualidade.

A medição da qualidade dos *clusters* foi feita com recurso ao índice de Dunn. É uma medida que atribui um *score* mediante a variância intra-*cluster* e inter-*clusters*. Quanto maior, teoricamente melhor será o *cluster*.

Recorde-se que este processo foi executado duas vezes. Uma vez com o texto integral (com o respetivo pré-processamento) e outra apenas com nomes de entidades.

7.1.2 Apresentação e análise dos resultados

Esta secção tem como finalidade a apresentação dos aspetos relevantes e de interesse acerca dos dados e dos resultados obtidos durante a fase de experimentação. Será feita a respetiva discussão dos mesmos.

Tabela 5 - Níveis por categoria

Categoria	Níveis (Pré-Processamento)	Níveis (Só Entidades)
Castigos	6	21
Geral	9	28
Pré-Jogo	10	29
Lesões	5	21
Mercado	8	21
Óbitos	3	7
Resultados	15	26
Treinadores	4	14

A Tabela 5 apresenta o número de níveis de cada dendrograma para cada um dos cenários em estudo. Pode-se verificar que existe uma diferença significativa no número de níveis para cada um dos casos. Uma justificação plausível é a presença de termos mais genéricos dentro de uma categoria. Por exemplo, na categoria Lesões as palavras: “operado”, “lesionado”, “lesão”, “recuperado”, entre outras.

Como tal, para os dendrogramas do modelo apenas com pré-processamento normal, a corte será feito com um intervalo de 0,5. Exemplificando, para a categoria óbitos serão criados 6 níveis de corte: 0,5, 1, 1,5, 2, 2,5 e 3. O nível 0 está automaticamente excluído pois aí cada documento é um *cluster* de si próprio. Para o outro caso, será utilizado o intervalo de um (1), começando no nível 1.

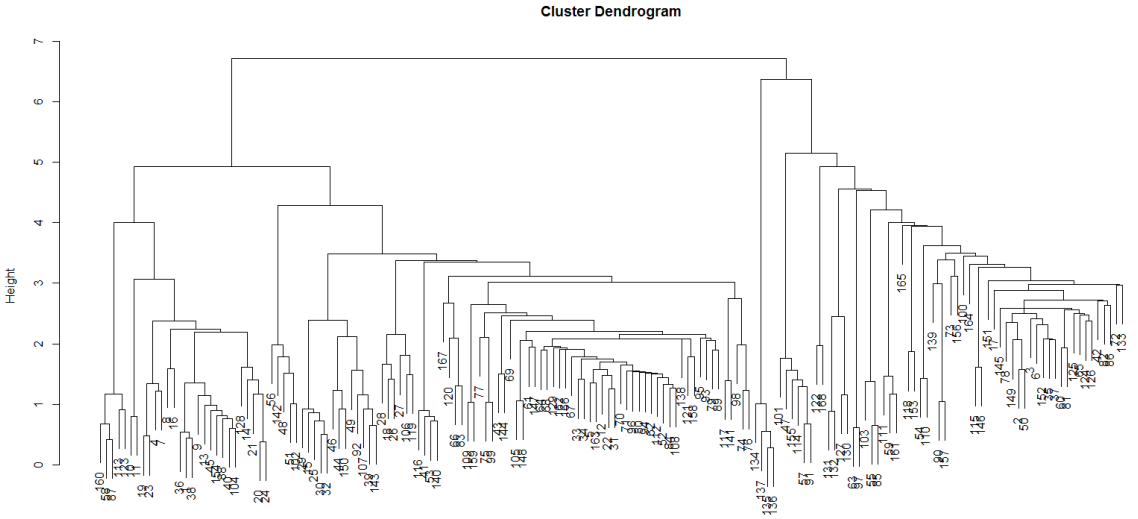


Figura 26 - Dendrograma da categoria óbitos

Para cada um dos casos foram calculados os índices de Dunn e apenas o nível de corte com melhor *score* será considerado como possibilidade para a avaliação manual.

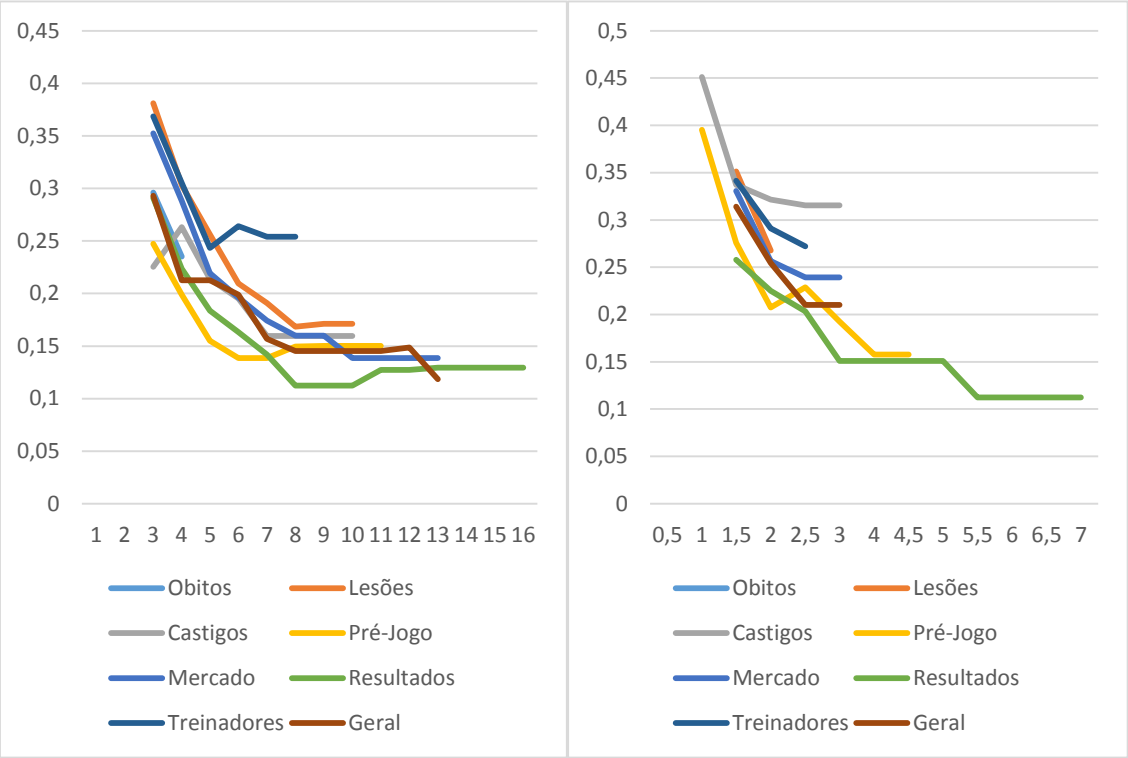


Figura 27 - Índice de Dunn por nível de corte. a) Só entidades b) Texto integral

A Figura 27 apresenta o *score* do índice de Dunn por cada nível de corte e para cada uma das categorias. Salvo raras exceções, no *score* de cada categoria não se verificam diferenças significativas entre os dois modelos.

Contudo, é facilmente observável que o valor vai decrescendo à medida que o nível de corte é mais elevado, facto que é facilmente justificável. Quanto maior for o nível de corte, menor o número de *clusters* será, como tal, a variância dentro dos *clusters* será maior, prejudicando o resultado final. Isto acontece porque o número de documentos em cada *cluster* aumenta, logo a variância intra-*cluster* também aumenta. Nos casos em que um documento é um *cluster* de si próprio a variância intra-*cluster* não se pode definir (o cálculo da variância requer um número mínimo de dois exemplos).

A Figura 28 apresenta o número de *clusters* para cada nível de corte para cada um dos modelos. É possível verificar que o nível de corte e o número de *clusters* são inversamente proporcionais.

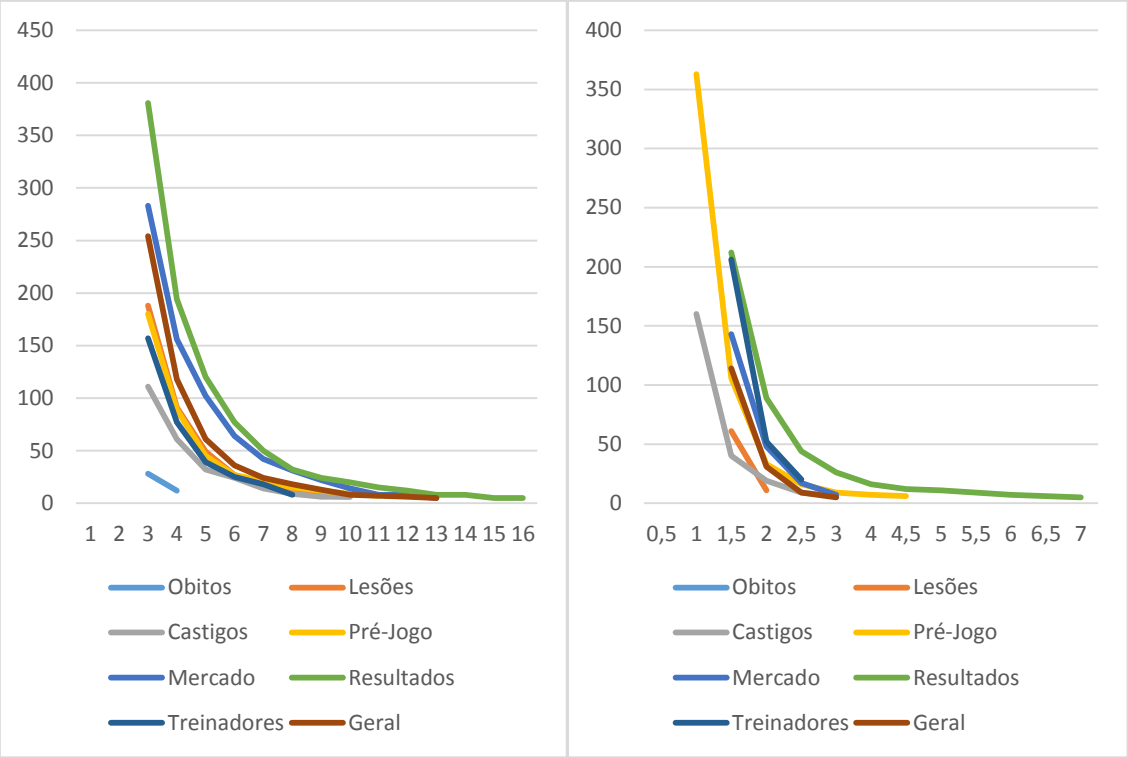


Figura 28 - Número de *clusters* por nível de corte. a) Só entidades b) Texto integral

Após a interpretação dos dados, apesar destes apresentarem alguns padrões já esperados, existem algumas dúvidas se os algoritmos hierárquicos serão realmente a melhor abordagem face ao problema proposto. Este tipo de algoritmos requerem que exista um corpus inicial com um considerável número de documentos para construir o dendrograma. A sua aplicabilidade neste projeto gera bastantes dúvidas.

Exemplificando, sempre que surgir um novo documento, este deve ser encaixado num *cluster* existente ou então num novo *cluster*. Os algoritmos hierárquicos não permitem fazer isso, exigiriam uma reorganização total dos *clusters* a partir do zero. Aos documentos já existentes seriam acrescentado o documento novo. Este processo seria não só muito demorado como existiria a possibilidade dos resultados serem diferentes cada vez que este processo tivesse que ser feito.

Os algoritmos hierárquicos apresentam-se uma boa solução para uma organização inicial dos dados, contudo, para o passo seguinte não se apresentam como a solução ideal. Este processo foi então descartado e não foi realizada a sua avaliação manual. No capítulo 7.2 é apresentada uma outra possível solução, o *k-means* adaptável baseado em *threshold*.

7.2 *K-means* adaptável

O *k-means* adaptável é um algoritmo baseado no tradicional *k-means* mas que não necessita obrigatoriamente da especificação de um número inicial de *clusters*. Este vai criando novos *clusters* à medida que necessita. Apresenta-se, à partida, como uma possível boa solução para o problema proposto.

7.2.1 Plano experimental

O algoritmo *k-means* adaptável recorre a uma valor limite (*threshold)* para construir os *clusters*. Esse valor consiste na distância máxima – Euclidiana, Cosseno, entre outros – admissível entre o documento ou *cluster* em análise e os restantes *clusters*. Este valor é, à partida, desconhecido pelo que há a necessidade de descobrir qual o valor ideal.

Assim, procedeu-se à aplicação do algoritmo múltiplas vezes, cada uma deles com um valor *threshold* diferente. O algoritmo pertence ao pacote "*akmeans: Adaptive Kmeans algorithm based on threshold*" e está disponível no website do projeto CRAN-R[11]. O algoritmo dá a possibilidade de utilizar quatro diferentes formas de cálculo da distância (Kwac 2014) na criação dos *clusters*. Devido a limitações técnicas apenas foi considerada como forma de cálculo da distância a similaridade do cosseno. As limitações técnicas referidas prendem-se com o fato de as máquinas disponíveis para processamento serem antigas, ou então máquinas recentes mas cujos componentes eram de baixo consumo, cujos componentes apresentam alguma perda no poder de processamento. Uma vez que cobrir todos os cenários tornar-se-ia um processo computacionalmente muito pesado, optou-se apenas por abordar a medida referida no parágrafo anterior.

Os valores utilizados para os testes ao algoritmo situam-se entre o intervalo de 0,025 e 0,50, existindo entre cada valor um intervalo de 0,025. Foi definido o valor máximo de 0,50 a ser testado pois é já um valor considerável para a distância entre documentos e *clusters*. À medida que o valor se aproxima de um, o algoritmo tende a criar um maior número de *clusters*.

O algoritmo implementado por Kwac (Kwac 2014) é uma implementação do *k-means* adaptável para o R. Ao contrário do apresentado no capítulo 4.1.2.2 este requer e a indicação de um número mínimo e máximo de *clusters*, contudo o funcionamento é igual, apenas cria mais *clusters* se for realmente necessário. Para cada categoria, o número máximo de *clusters*

[11] http://cran.r-project.org/web/packages/akmeans/

correspondeu ao número total de documentos da categoria. O algoritmo apresenta o seguinte funcionamento:

1. É executado o *k-means* com o valor de k a ser o inicialmente definido pelo utilizador.

2. Para cada *cluster*, é verificada a condição de *threshold*.

3. Se todos os *clusters* satisfazerem essa condição o algoritmo termina aí.

4. É feita a verificação se o valor de k é maior que o número máximo de *clusters* definido. Se sim, termina o processo nesse momento. Em caso contrário avança.

5. Para qualquer *cluster* que viole a condição de *threshold*, é executado *k-means* nos membros desse *cluster* com o valor de $k = 2$. O número de *clusters* irá aumentar de acordo o número de *clusters* que violem a condição de *threshold*. Se existirem dois *clusters* nessa situação, desses dois formarão um total de quatro *clusters*. Volta ao passo 3.

É importante explicar quando é que acontece uma violação da condição de *threshold*. Essa violação ocorre quando o valor da similaridade é inferior ao definido como limite da condição. Exemplificando com o valor de 0,99. Recorde-se que quando o valor da similaridade do cosseno é um, tem-se dois vetores com a mesma orientação, neste caso prático, estaríamos a falar de dois documentos idênticos (ambos representados pelos mesmos termos). Portanto, neste caso, para dois ou mais documentos (ou *clusters*) serem agregados o seu valor de similaridade teria de ser maior que 0,99, dois documentos praticamente idênticos. A imagem seguinte representa a aplicação deste exemplo no R.

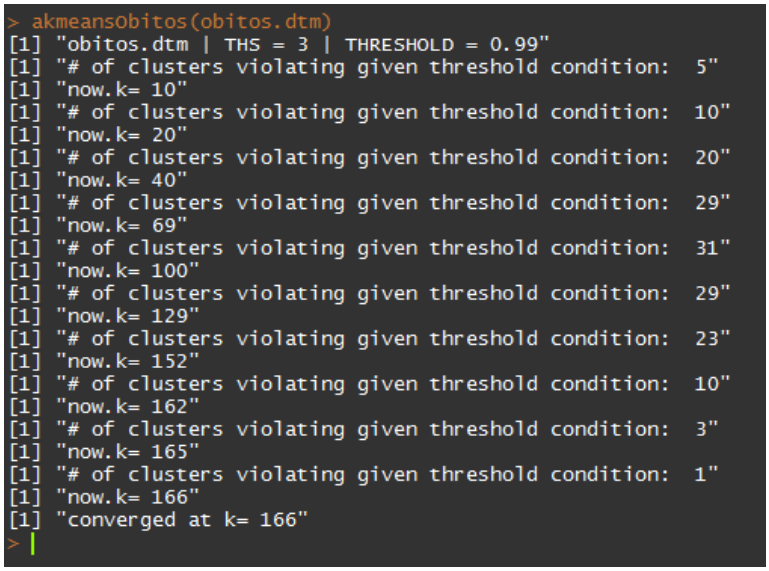


Figura 29 – Execução do *k-means* adaptável

Pode-se verificar que dos 168 documentos pertencentes à categoria "Óbitos", foram criados 166 *clusters*. Um valor próximo do que seria o expectável uma vez que não existem praticamente documentos iguais. Existem dois *clusters* com dois documentos cada pois existem dois pares de documentos repetidos.

Ainda que o objetivo seja agrupar os documentos em eventos, a forma como esses eventos são descritos varia de jornalista para jornalista, pelo que é natural que existam algumas diferenças na escrita de cada um. Como tal, o valor de *threshold* não deve ser exageradamente elevado.

Como será visível mais adiante, existem também condicionantes temporais e ao nível do processamento à medida que o valor de *threshold* vai aumentando.

À semelhança dos algoritmos hierárquicos, este algoritmo será aplicado para os documentos com o texto integral com o devido pré-processamento e apenas com entidades.

7.2.2 Apresentação e análise dos resultados

Nesta secção são apresentados alguns dados e resultados obtidos durante a fase de experimentação e a respetiva discussão dos mesmos.

A criação dos *clusters* é um processo que está diretamente relacionada com o número total de documentos. A Figura 30 (os dados do gráfico apresentam-se expressos em segundos) confirma essa relação, a qual seria algo expectável. Uma vez que existe um maior número de

pontos de entrada, terão que ser necessariamente feitos mais cálculos da distância entre documentos e *clusters*.

Nota: Os dados apresentados a partir deste momento são referentes ao modelo apenas com entidades, salvo indicação contrária. Isto porque apresentaram comportamentos semelhantes na grande parte das análises apresentadas neste capítulo.

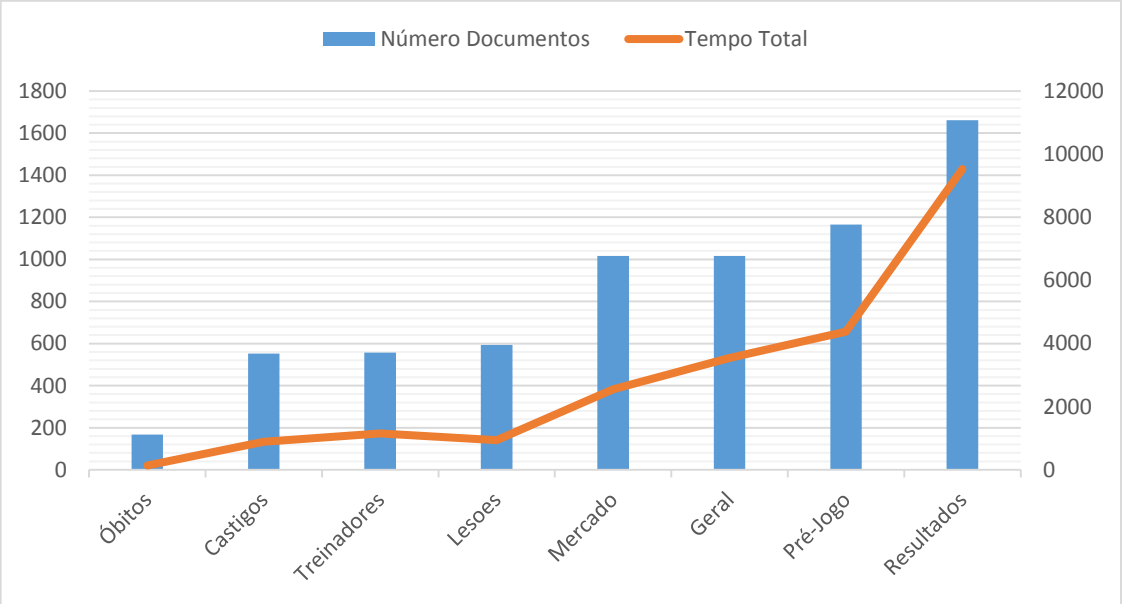


Figura 30 - Relação entre o número de documentos e o tempo na criação dos *clusters*

Apesar do gráfico confirmar a tendência referida anteriormente, há que ter em atenção que essa relação não é assim tão linear. Pode-se verificar que existe uma diferença no tempo total entre as categorias “Mercado” e “Geral” bastante considerável apesar de estes terem aproximadamente o mesmo número de documentos. Uma situação idêntica verifica-se entre as categorias “Castigos”, “Treinadores” e “Lesões”.

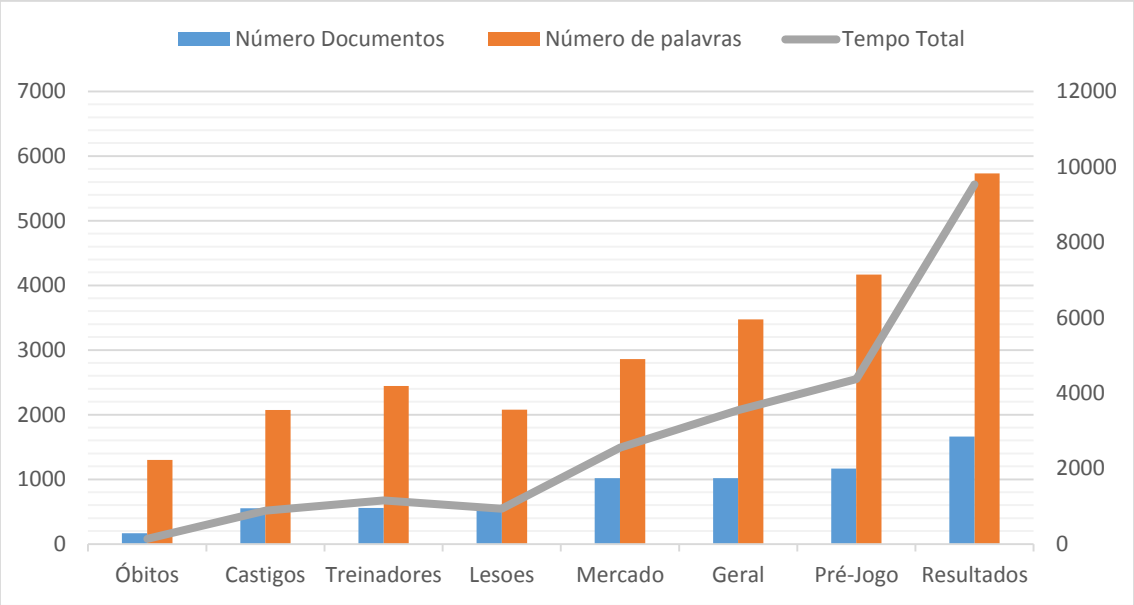


Figura 31 – Relação entre número de documentos, número de palavras e o tempo total

O gráfico anterior apresenta uma causa plausível para o facto referido no parágrafo anterior. Após uma análise cuidada, pode-se verificar que o número total de termos na matriz termo-documento tem um peso significativo no processo de *clustering*. O caso mais visível é entre as categorias "Castigos", "Treinadores" e "Lesões" e "Geral" e "Mercado". Apesar de os dois grupos de categorias terem um número de documentos muito aproximado, consegue-se verificar que a flutuação do número de palavras (termos) influencia o tempo necessário para a execução do algoritmo.

Como referido na introdução deste capítulo, existiram algumas limitações que condicionaram o intervalo do valor limite da condição de violação. Como é visível na Figura 32, à medida que esse valor vai aumentando o tempo de execução do algoritmo aumenta de forma significativa. Isto deve-se ao facto de à medida que esse valor aumenta ser exigido que os *clusters* tenham documentos cada vez mais semelhantes. Isto leva necessariamente a um aumento do número total de *clusters*.

Como tal, exige um maior número de cálculos de forma a calcular a semelhança entre os documentos e os *clusters*. É portanto expectável que o tempo de execução do algoritmo cresça.

Nota: Os valores da Figura 32 encontram-se expressos em segundos.

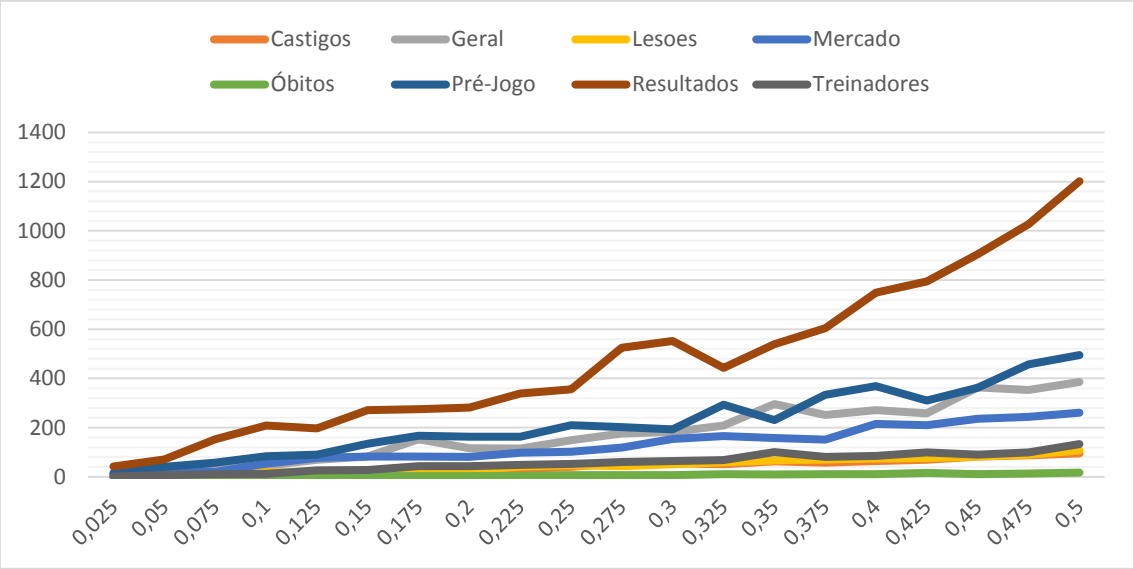


Figura 32 - Tempo execução do algoritmo por *threshold*

Para amostras consideráveis, este algoritmo pode-se tornar bastante demorado, principalmente quando o valor de *threshold* tende a aumentar.

Assim, é necessário encontrar um equilíbrio entre o valor de *threshold* e o tempo de execução do algoritmo. Um dos objetivos do projeto a desenvolver é que a obtenção dos resultados seja um processo rápido e eficaz. O sistema a implementar deve ser suficientemente rápido a agregar notícias mas, ao mesmo tempo, deve apresentar um nível aceitável de precisão.

Este algoritmo tem de facto o grande inconveniente de necessitar de um alargado tempo de execução, principalmente quando valor da condição de limite tende a aumentar. Comparativamente com os algoritmos hierárquicos, existe uma grande diferença no tempo de execução a favor dos algoritmos hierárquicos. Esta questão deve ser seriamente encarada aquando da escolha do algoritmo ideal para o problema proposto.

A Figura 33 apresenta o tempo acumulado necessário para a execução dos algoritmos hierárquicos.

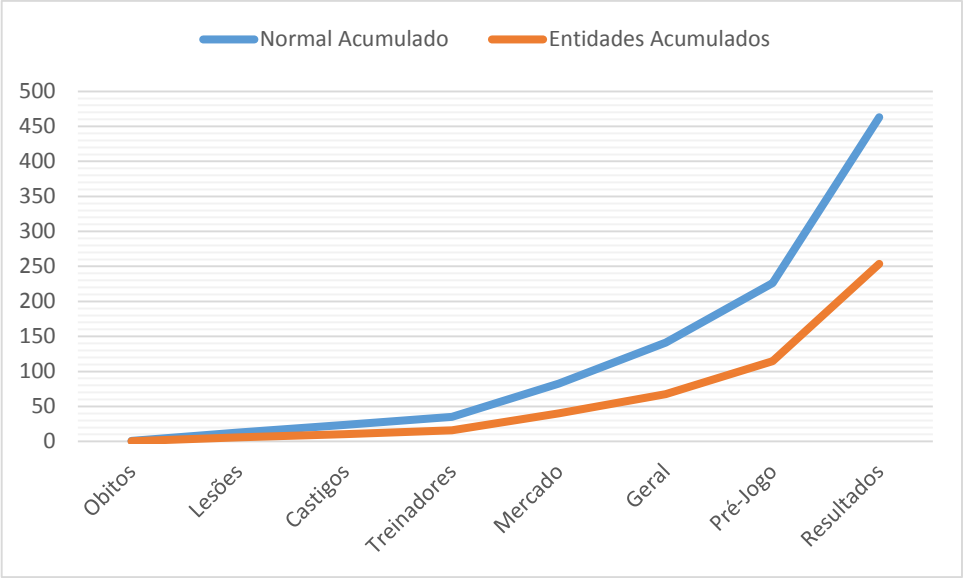


Figura 33 – Tempo execução dos algoritmos hierárquicos

É importante ter em conta que que numa fase de pleno funcionamento do produto a desenvolver, sempre que este processo for feito para novos documentos, estes serão em menor número face ao corpus apresentado. Contudo, o número de *clusters* deverá ser ainda maior à medida que o tempo passa. Mesmo que seja necessário utilizar um valor de *threshold* elevado, dado o menor número de documentos, o tempo necessário para execução deverá ser significativamente menor. Porém este será um assunto a abordar num trabalho futuro.

Poderão ser também aplicadas técnicas de forma a limitar o número de *clusters* a serem considerados sempre que chega um novo conjunto de documentos. Uma técnica a aplicar pode ser a "congelação" de *clusters* após um determinado período de tempo sem ser adicionado nenhum documento novo ao *cluster.* Poderá permitir reduzir o tempo necessário para encaixar novos documentos nos *clusters*. Novamente, este ponto não será objeto de estudo no presente relatório. O principal objetivo é perceber a aplicabilidade ou não dos algoritmos apresentados.

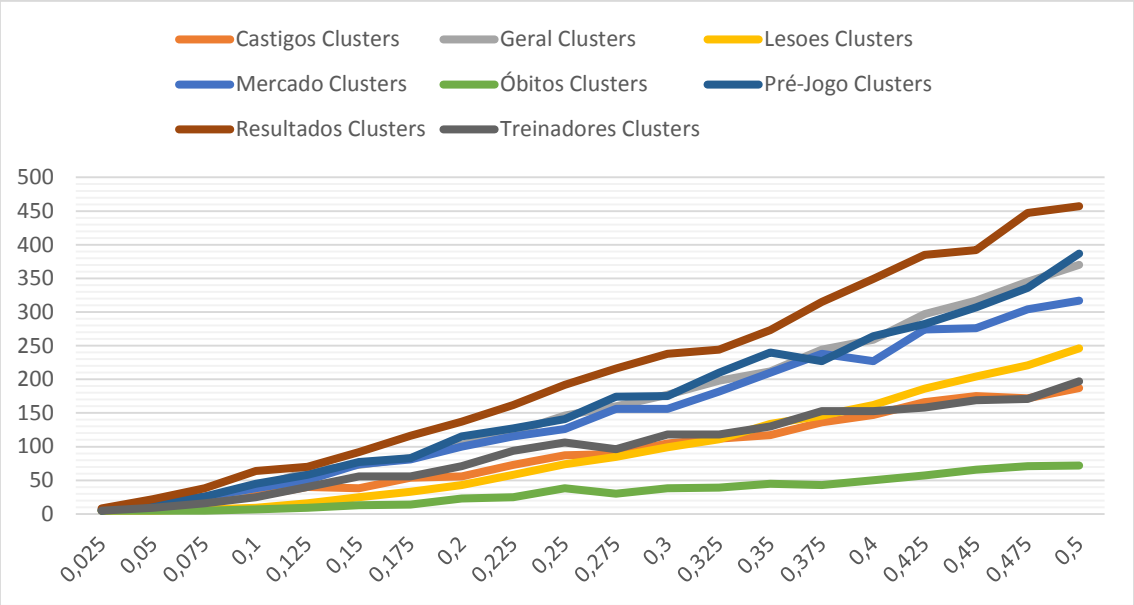


Figura 34 - Relação entre valor de *threshold* e o número de *clusters* gerados

A Figura 34 representa aquilo que já foi dito anteriormente neste capítulo. À medida que o valor limite da condição de violação aumenta, exige que os *clusters* tenham cada vez mais documentos mais semelhantes. Sendo documentos relativamente extensos e escritos por diferentes pessoas, é natural que não existam documentos idênticos, pelo que é uma situação normal este aumento do número de *clusters*.

É possível também justificar mais uma vez a opção de se limitar o valor de *threshold* ao intervalo compreendido entre 0,025 e 0,50. Recorde-se que a justificação inicial residia no facto de valores muito elevado terem exigirem uma carga de processamento relativamente pesada e, que por sua vez, está diretamente relacionado com o tempo necessário para a execução de todo o processo. Contudo, como referido no capítulo 7.1.1, número de *clusters* tem que ser no máximo de $1/3$ do número total de documentos por categoria. Esta condição parte do prossuposto que pelo menos três das cinco fontes das quais os dados serão recolhidos abordarão em grande parte das ocasiões os mesmos acontecimentos. A única exceção será entrevistas exclusivas de uma das fontes.

Antes de ser realizada a avaliação manual, foi realizada uma avaliação estatística para cada um dos modelos criados pelo algoritmo. Foi avaliada a qualidade dos *clusters* para cada uma das categorias, para cada valor de *threshold* e para cada tipo de pré-processamento. Um total de 320 modelos diferentes.

A avaliação estatística foi realizada de forma a reduzir o número de modelos a serem avaliados pelos voluntários. É impensável avaliar manualmente todos os 320 modelos que foram

criados pelo algoritmo. Os modelos foram avaliados com recurso ao índice de Dunn que é disponibilizado pelo pacote *clValid* do R[12].

Para cada categoria de cada tipo de pré-processamento foram escolhidos os dois ou três valores de *threshold* mais elevados após observação. Nos casos em que o índice de Dunn era muito aproximado e os valores de *threshold* relativamente distantes, foram selecionados aqueles com um *threshold* mais elevado. *Thresholds* demasiados baixos (menos que 0,10) foram excluídos uma vez que a tendência é a de criar *clusters* muito heterogéneos com muitos documentos.

Foi também selecionado um valor comum a todas as categorias e modelos de forma a avaliar a aplicabilidade de um valor de *threshold* comum a todas as categorias.

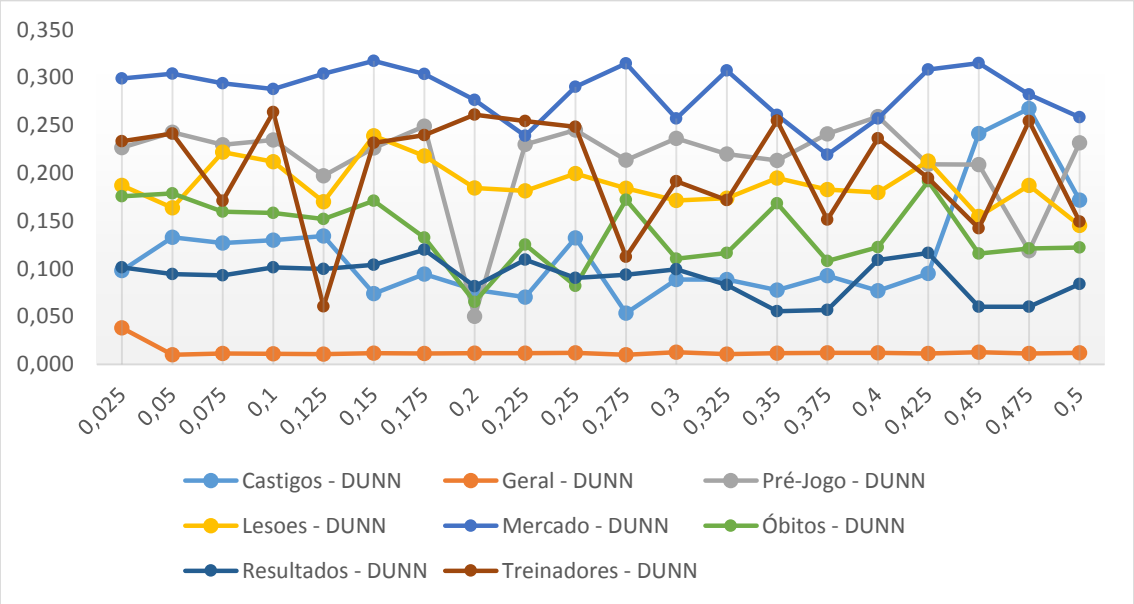


Figura 35 - Índice de Dunn por categoria

[12] http://cran.r-project.org/web/packages/clValid/index.html

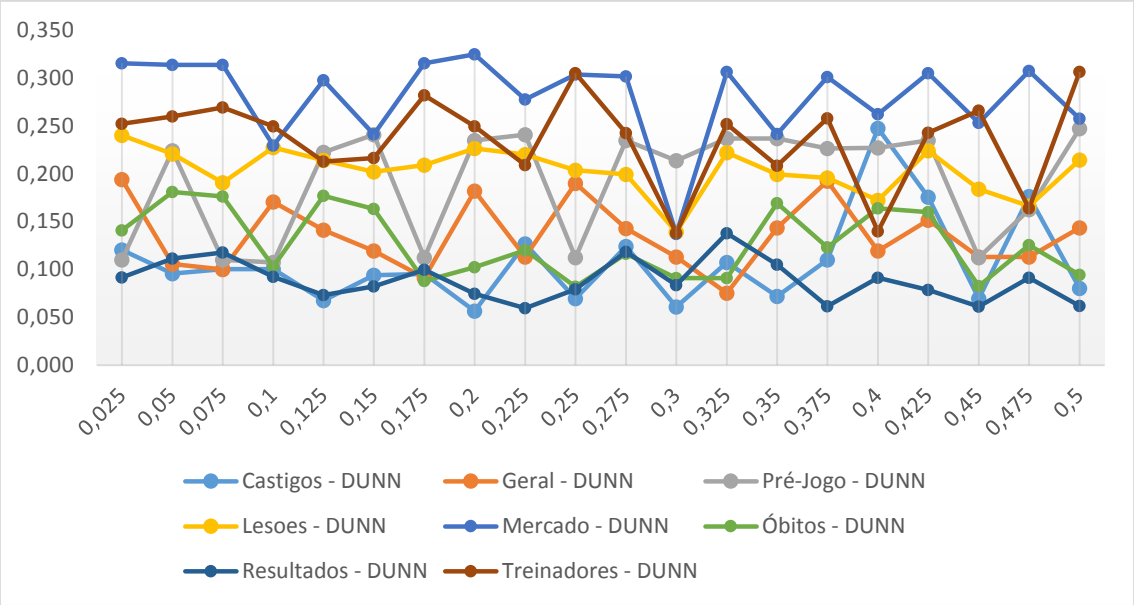


Figura 36 - Índice de Dunn por categoria (apenas entidades)

A Figura 35 e a Figura 36 apresentam as classificações obtidas por cada categoria e por cada valor de *threshold*. Os resultados revelam-se bastante surpreendentes na forma em que não se verifica nenhuma tendência à medida que o valor de *threshold* aumenta.

Numa rápida análise aos dados apresentados e, tendo em conta todas a categorias, globalmente os dois tipos de pré-processamento apresentam resultados semelhantes. Em algumas categorias apresenta melhores resultados, em outras, não. Nos anexos 1 e 2 pode-se consultar as tabelas com as avaliações estatísticas para cada categoria, *threshold* e tipo de pré-processamento.

Uma vez que a avaliação estatística indica valores aproximados para cada um dos casos e, como os tempo de execução no modelo apenas com as entidades é consideravelmente menor, justifica-se realizar a avaliação humana apenas com este caso. Assim é possível, mais uma vez, reduzir o esforço do avaliador. Em média, houve um ganho de 48% no tempo necessário para a execução dos algoritmos. A Figura 37 apresenta a comparação entre os dois modelos para cada categoria.

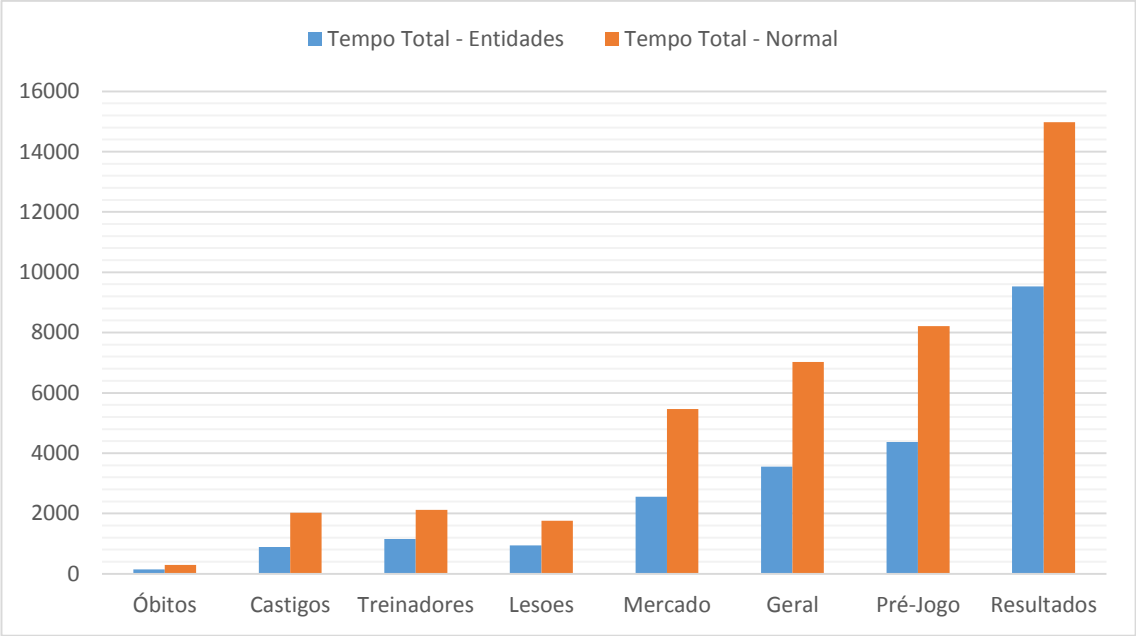


Figura 37 - Tempo total para cada tipo de pré-processamento

Na figura seguinte é apresentado um gráfico de radar que resume a distribuição de alguns valores de *threshold* com melhor índice de Dunn. Nos casos onde o índice é bastante aproximado, optou-se por selecionar aquele que apresentem um valor limite superior. A sua observação permite tirar algumas conclusões.

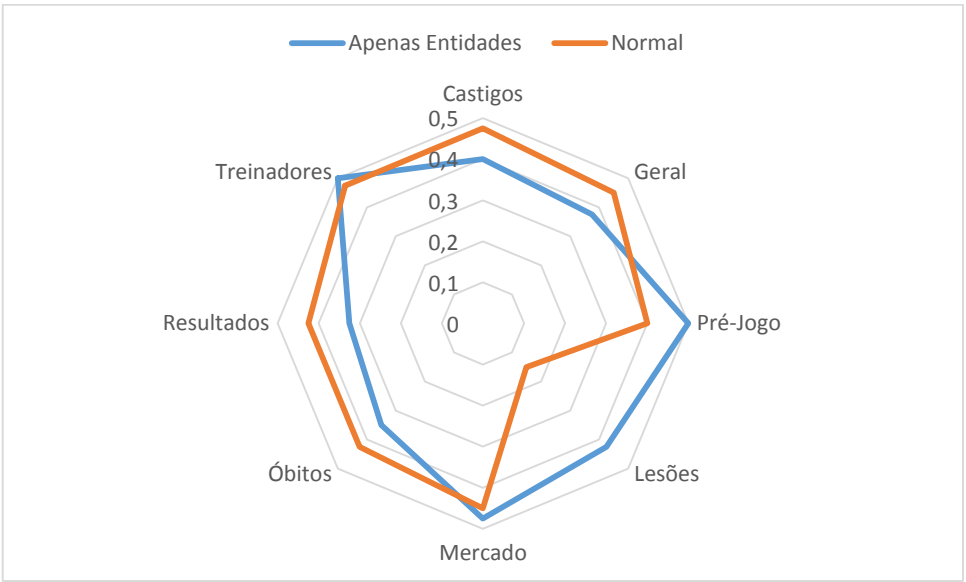


Figura 38 - Gráfico de radar dos maiores valores de *threshold* por categoria

Com base na Figura 38, é possível perceber que no intervalo entre 0,40 e 0,50 consegue-se obter, teoricamente, bons resultados. Como tal, será testado o valor de 0,45, para todas as categorias de forma a verificar a possibilidade da utilização de um valor de *threshold* comum a

todos os casos existentes A escolha do valor de 0,45 é devido ao fato de ser a metade do intervalo referido anteriormente. Apenas em 4 situações de 16 possíveis se verifica um valor de *threshold* inferior a 0,40.

No caso de existirem valores consecutivos (0,425 e 0,45 ou 0,45 e 0,475) prevalecerá o valor da categoria. Assim, tendo em conta observação das tabelas, as condições referidas anteriormente e o valor de *threshold* comum, os valores de *threshold* a serem testados na avaliação manual são:

Tabela 6 – Valores *threshold* a utilizar na avaliação manual

Categoria	Apenas Entidades
Castigos	0,4; 0,45
Geral	0,25; 0,375; 045
Pré-Jogo	0,225; 0,35; 0,45; 0,5
Lesões	0,2; 0,425
Mercado	0,20; 0,475
Óbitos	0,125; 0,35; 0,45
Resultados	0,325; 0,45
Treinadores	0,25; 0,45

7.3 Avaliação

Para além da avaliação estatística, foi realizada uma avaliação manual feita por vários voluntários. A principal razão para a necessidade da realização desta avaliação é para ser possível obter uma ideia da qualidade do processo de *clustering* sob o ponto de vista do utilizador.

Apesar do problema proposto ser a agregação de notícias por eventos, é natural que de utilizador para utilizador o conceito dessa relação entre notícias varie. Por exemplo, para um utilizador, um *cluster* com notícias sobre a nomeação de árbitros para várias partidas pode ser perfeitamente aceitável. Para outros já será mais normal apenas para um determinado jogo.

O principal objetivo é a avaliação da qualidade de cada *cluster*. Isto é, se os documentos dentro de um *cluster* se encontram minimamente relacionados e se referem a um mesmo evento ou acontecimento. Estes foram avaliados individualmente tendo em conta uma pontuação na escala de um (1) a cinco (5):

- 1 (Muito mau) – *cluster* com documentos sem qualquer tipo de relação entre eles. Não há um mínimo de dois documentos que estejam relacionados, seja acerca de um acontecimento ou entidades em comum.

- 2 (Mau) – existem neste *cluster* alguns documentos que se relacionam entre si, quer seja um acontecimento ou entidades em comum. Contudo, estes são muito poucos.

- 3 (Aceitável) – nestes *clusters* os documentos podem não se relacionar diretamente com um acontecimento, mas as entidades envolvidas são maioritariamente as mesmas. *Clusters* só com um documento.

- 4 (Bom) – c*lusters* cujos documentos estão diretamente relacionados com um determinado acontecimento. Pode haver um ou dois documentos "perdidos".

- 5 (Muito Bom) – c*lusters* perfeitos. São aqueles em que todos os documentos se referem ao mesmo evento ou cadeia de acontecimentos. Não existe ruído.

Concluída a avaliação individual dos *clusters*, foi pedido algum feedback aos avaliadores de forma a se ter uma avaliação global do algoritmo. A cada avaliador foi pedido que enumerasse problemas encontrados e sugestões, caso existissem.

Esta avaliação tem um caracter qualitativo e não quantificativo. Os avaliadores, como humanos que são, tem perceções e entendimentos diferentes para as mesmas situações, sendo natural que para uma mesma situação existam opiniões diferentes.

De forma a reduzir o esforço necessário para o avaliador, foi utilizado o mesmo corpus que anteriormente, porém, com um menor número de documentos por categoria. Para cada uma foram selecionados aleatoriamente 33% dos documentos de cada uma das fontes e categoria (amostra estratificada). Uma vez que a categoria "Geral" é uma categoria muito heterogénea não foi escolhida para avaliação.

Para a realização da avaliação foram escolhidos 7 voluntários. Para cada categoria e valor *threshold* foi feita uma avaliação. Os modelos a avaliar foram distribuídos aleatoriamente entre os voluntários. Essa distribuição pode ser consultada no anexo 3.

Tabela 7 - Resultados da avaliação manual

	0,125	**0,2**	**0,225**	**0,25**	**0,325**	**0,35**	**0,4**	**0,425**	**0,45**	**0,475**	**0,5**
Castigos							3,23		3,09		
Pré-Jogo			3,07			3,43			2,63		3,02
Lesões		2,46						2,97			
Mercado		2,39								3,01	
Óbitos	1,00					2,75			2,43		
Resultados					2,84				2,80		
Treinadores				2,89					3,03		

A Tabela 7 apresenta os resultados da avaliação de cada modelo. Para o resultado final de cada modelo foi calculada a média da pontuação atribuída a cada um dos *clusters*. Uma análise cuidada permite perceber que a subjetividade tem alguma influência nos resultados. Por exemplo, na categoria pré-jogo é visível uma quebra abrupta nos resultados entre os valores de 0,225 e 0,325 para 0,425 e 0,50.

De uma forma geral, os avaliadores consideraram a distribuição dos documentos pelos *clusters* aceitável, principalmente para valores de *threshold* mais elevados, onde existe uma menor variância entre as avaliações.

O maior problema encontrado foi o facto de no conjunto de dados fornecido a data de publicação da notícia não ter sido tomada em conta. Pelo que acontece às vezes eventos diferentes estarem no mesmo *cluster*.

Foram também enumeradas várias situações em um mesmo acontecimento estava distribuído por vários *clusters*. Ainda assim, nesses *clusters* existiam na grande maioria apenas documentos referentes a esse evento.

Outro problema referido é por vezes os *clusters* agruparem dois eventos diferentes, mesmo as entidades não tendo grande relação entre si. Por vezes, esta situação verificava-se em notícias de um determinado acontecimento em que era feita a referência a evento anterior. Por exemplo, em uma notícia prévia a um jogo de uma determinada equipa, existe a referência ao resultado da última partida que esse clube disputou. Isto levou muitas vezes à agregação de notícias referentes a esses dois acontecimentos no mesmo *cluster*.

Uma última situação era o facto de alguns *clusters* abordarem um evento, mas continham mais algumas notícias não relacionadas com esses acontecimentos, mas que eram respeitantes às mesmas entidades do evento em questão.

Estes problemas enumerados permitem numa posterior fase de implementação do sistema ter em conta esta limitações e então aplicar mecanismos de forma a reduzir esses problemas. Contudo, dada a natureza do problema é impensável atingir um nível de precisão tal que todas as falhas referidas sejam totalmente suprimidas. Após a análise às falhas mencionadas foi possível chegar a um conjunto de soluções que poderão, à partida, permitir melhor este processo.

O problema da data de publicação é uma limitação que acontece devido aos *clusters* terem sido construídos a partir de um *dataset* inicial com um intervalo de tempo alargado. Como referido no capítulo 7.2.2 este processo quando implementado "*on-the-fly*" lidará com menores conjuntos de dados a cada execução, como tal, o algoritmo poderá ser mais eficaz neste problema. A técnica de "congelação" de *clusters* é uma solução que teoricamente permitirá ajudar a ultrapassar esta limitação.

Uma outra solução que poderá permitir uma maior eficácia deste processo é a comparação entre os títulos e descrição com o texto completo. A maioria das notícias descarregadas pelos RSS *feeds* têm esses três campos, sendo a descrição uma versão resumida da notícia completa. Assim, atribuir um peso extra às entidades que se encontram no título ou descrição, teoricamente, permitirá obter uma maior performance do algoritmo. Um pouco à semelhança das técnicas apresentadas no capítulo 4.3.1 para a sumarização de documentos.

Utilizando um processo semelhante de atribuição de peso extra referido no parágrafo anterior, uma solução poderá ser a construção de dicionários com palavras-chave para cada categoria. Exemplificando, para a categoria “Lesões”, algumas palavras-chave poderão passar por partes do corpo humano onde é frequente existirem lesões, como joelho, tornozelo, ombro, entre outros. Para a categoria “Castigos” algumas palavras-chave poderão ser “expulso”, “suspenso”, etc.

A criação de um módulo exclusivamente dedicado a entidades de futebol poderá permitir um aumento da eficácia do processo, recorde-se que o método de reconhecimento de entidades implementado para a avaliação ao problema proposto é muito rudimentar. A construção de uma base de dados com recurso a terceiros para posterior comparação poderá ser uma solução.

Em jeito de conclusão, considera-se que este algoritmo apresenta enormes potencialidades para a aplicação no projeto. Atualmente já apresenta resultados aceitáveis, mas com algumas das conclusões que foi possível tirar, há ainda larga margem para melhorar o processo e apresentar uma solução ainda com maior qualidade. Como tal, o *k-means* adaptável será o algoritmo escolhido para o módulo de agrupamento de notícias por eventos.

Os valores de *threshold* entre o intervalo de 0,40 e 0,45 apresentam algum equilíbrio nas várias avaliações realizadas nesse intervalo. Como tal, apresentam-se como possíveis valores a serem utilizados. Porém, face às sugestões de melhoria referidas, caso se verifiquem acertadas, estes valores poderão não ser os mais adequados, pelo poderá haver necessidade de fazer novas avaliações.

8 *Tagging* – Nuvem de palavras

Neste capítulo será feita uma abordagem muito breve à técnica da nuvem de palavras. O objetivo para este capítulo é perceber se as palavras ou entidades que aparecem com maior frequência permitem identificar os tópicos e entidades do evento abordado no *cluster*.

8.1 Plano experimental

O procedimento experimental para esta tarefa é bastante simples. Para começar foi selecionado um dos ficheiros que foi alvo de avaliação no capítulo anterior (7.3). O documento escolhido foi o da categoria "Castigos" e com *threshold* de 0,40. Este foi o ficheiro escolhido devido a ter um número de documento não muito grande e porque teve uma das melhores avaliações (capítulo 7.3). Para esse documento foram eliminados os *clusters* cuja avaliação era inferior a quatro (4). Foram avaliados um total de 25 *clusters*.

Figura 39 - Nuvem de palavras do *cluster* 23

Para cada *cluster* foi criada uma nuvem de palavras com as cinco palavras mais frequentes. Foi depois pedido a três voluntários para avaliar a adequabilidade das palavras extraídas face às notícias que compõe o *cluster* tendo em conta três valores:

- 1 (Mau) – Existem poucas ou nenhumas palavras relacionadas com as entidades do evento abordado no *cluster*.
- 2 (Aceitável) – A grande parte das palavras adequam-se às entidades referidas no *cluster*.
- 3 (Bom) – Todas as palavras se adequam ao *cluster*.

Foi também pedido que enumerassem problemas encontrados ou sugestões de forma a permitir implementar no futuro um sistema mais eficaz.

8.2 Apresentação e avaliação dos resultados

A avaliação às palavras-chave dos *clusters* foi feita por três voluntários, cada um avaliando um total de 25 *clusters*. No final, foi calculada a média de cada um dos avaliadores.

Tabela 8 - Avaliação da nuvem de palavras

Cluster	Avaliador 1	Avaliador 2	Avaliador 3
1	3	2	3
2	2	3	2
3	3	3	3
4	3	3	3
5	2	2	1
6	2	2	2
7	2	2	2
8	2	2	2
9	3	2	2
10	2	3	3
11	1	1	1
12	2	2	2
13	3	3	3
14	2	1	2
15	3	3	3
16	1	2	1
17	3	2	3
18	3	3	3
19	2	3	3
20	2	2	2
21	3	2	3
22	3	3	3
23	3	3	3
24	3	3	3
25	2	2	2
Média	**2,4**	**2,36**	**2,4**

Os resultados estatísticos indicam que as palavras mais frequentes podem identificar os principais intervenientes de um *cluster*. Porém, no feedback dado pelos avaliadores foi referido um problema comum a todos. Nas entidades com mais de uma palavra, cada palavra é tratada apenas como uma entidade e não como um nome composto.

Uma solução para este problema poderia ser a utilização de n-gramas (ver capítulo 2.7.2). A sua utilização permitiria a utilização de termos compostos. Contudo, um sistema de reconhecimento de entidades com qualidade combinado com algumas técnicas referidas no

capítulo 4.3.1 poderão permitir a identificação dos principais intervenientes das notícias de um *cluster*.

Este processo das nuvens de palavras revelou-se também pouco indicado face ao problema proposto. As nuvens de palavras geradas no R (pacote *wordcloud*[13]) são apenas representações gráficas, não permitindo a extração das suas palavras. Assim, esta não será uma abordagem a seguir no trabalho a desenvolver futuramente.

[13] http://cran.r-project.org/web/packages/wordcloud/

9 Conclusões

Neste capítulo serão apresentadas as conclusões finais do trabalho realizado, serão enumerados os objetivos realizados, identificadas as limitações e o trabalho futuro.

9.1 Objetivos cumpridos

Os objetivos pretendidos face aos problemas propostos foram na sua grande maioria cumpridos. É importante referir que o período exclusivamente dedicado ao estudo do estado da arte permitiu adquirir novos conhecimentos e ainda consolidar alguns outros, principalmente no que diz respeito à classificação de texto e *clustering* de documentos. Permitiu que fase de experimentação, análise e avaliação dos resultados fosse abordada mais confortavelmente.

Os objetivos face ao problema da classificação de notícias foram largamente cumpridos. A experimentação realizada permitiu concluir que o *support machine vector* se perfila como um algoritmo capaz de resolver o problema apresentado de uma forma eficaz. Este é o algoritmo que será utilizado no trabalho futuro.

Relativamente ao processo de *clustering*, os objetivos pretendidos foram cumpridos também quase na íntegra. O estudo comparativo entre os algoritmos hierárquicos e o *k-means* adaptável permitiu estabelecer as vantagens e desvantagens de cada um. A experimentação permitiu concluir que o *k-means* adaptável é a solução ideal para a resolução dos problemas do agrupamento de notícias por eventos. Este será o algoritmo que será utilizado no projeto a desenvolver.

Contudo, foram detetados alguns problemas na fase experimental deste processo. Ainda que sido possível estabelecer propostas para algumas possíveis soluções, os resultados obtidos apesar de aceitáveis podem ainda ser bastante melhorados. Ainda assim, os resultados obtidos são bastante prometedores e, caso as soluções propostas tenham sucesso, espera-se que os resultados sejam muito bons. Daí considerar que os objetivos não foram totalmente atingidos. Não foi obtida uma solução final, contudo, é preciso ter em conta que para este problema é muito difícil atingir níveis muitos altos de precisão, até porque essa avaliação caberá muitas vezes à subjetividade de utilizador para utilizador.

A identificação das principais entidades de um conjunto de notícias é o último dos problemas propostos. A ideia inicial era utilizar nuvens de palavras para a extração das mesmas, contudo, concluiu-se que é uma solução desnecessária e que a sua implementação não traz qualquer tipo de vantagem. A implementação de um NER deverá permitir para além de melhorar a performance do processo de *clustering*, identificar as principais entidades num conjunto de notícias. O uso combinado de um NER mais robusto com algumas das técnicas de atribuição de pesos às entidades consoante a sua importância no texto (utilizando algumas das técnicas utilizadas na extração de sumários de documentos) poderão permitir obter melhores resultados.

9.2 Limitações e dificuldades ultrapassadas

As limitações não foram muitas. A maior delas foi o fato dos computadores que tinha disponíveis para o processamento da informação eram bastante limitados a nível de processamento e memória. O baixo nível de processamento levava a que as tarefas de classificação e *clustering* demorassem muito tempo. Houve situações que foi necessário executar algoritmos mais que duas ou três vezes após terem sido detetados erros após a execução anterior. Por exemplo, para a classificação, cada execução do SVM e KNN chegou a demorar mais de 24 horas.

A baixa memória disponível impossibilitava muitas das vezes de efetuar qualquer outra tarefa enquanto eram executados alguns algoritmos. O R exige bastante ao nível de memória RAM, pelo que não ajuda esta situação.

Uma das dificuldades enfrentadas foi o facto de o R ser algo completamente novo para mim. Não é uma ferramenta fácil para quem se aventura nela pela primeira vez, tendo, na minha opinião, uma curva de aprendizagem um pouco longa. Exigiu bastante trabalho para que conseguisse dominar minimamente a ferramenta.

Provavelmente a maior dificuldade ultrapassada foi a necessidade de categorizar manualmente todas as notícias armazenadas. Apesar de ter sido criado um *dataset* com valor, a realização desta foi bastante desgastante mentalmente. Era uma tarefa bastante maçadora e repetitiva e que exigiu muita paciência.

9.3 Trabalho Futuro

O trabalho apresentado neste relatório consistiu na aplicação das várias técnicas em corpus estático. Para a classificação automática não existe problema em utilizar a abordagem apresentada, contudo, para o *clustering* há necessidade de se fazerem algumas alterações. O trabalho futuro passa agora por adaptar os algoritmos de *clustering* (*k-means* adaptável) a *data-streams* (informação em tempo-real).

Agora que concluído este trabalho de investigação, a fase que se segue é a de planeamento e implementação. Os resultados obtidos e as respetivas conclusões vão certamente facilitar a implementação do módulo de classificação e *clustering* automático de notícias. O objetivo é iniciar a implementação em finais do último trimestre do ano de 2014.

10 Referências

Agarwal, S., Singhal, A. & Bedi, P., 2012. Classification of RSS Feed News Items Using Ontology. *12th International Conference on Intelligent Systems Design and Applications*.

Alves, A., 2010. *Modelo de representação de texto mais adequado à classificação*. Instituto Superior de Engenharia do Porto.

Bayes, T., 1763. An essay towards solving a problem in the doctrine of chances. *Phil. Trans. of the Royal Soc. of London*, 53, pp.370–418.

Bergroth, L., Hakonen, H. & Raita, T., 2000. A Survey of Longest Common Subsequence Algorithms. *7th International Symposium on String Processing and Information Retrieval*, pp.39–48.

Bezerra, E. & Goldschmidt, R., 2010. A Tarefa de Classificaçao em Text Mining. *Revista de Sistemas de Informacao da FSMA*, 5, pp.42–62.

Bhatia, S.K., 2004. Adaptive K-Means Clustering. In V. Barr & Z. Markov, eds. *FLAIRS Conference*. AAAI Press, pp. 695–699. Available at: http://dblp.uni-trier.de/db/conf/flairs/flairs2004.html#Bhatia04.

Bouras, C. & Tsogkas, V., 2010. Assigning Web News to Clusters. *5th International Conference on Internet and Web Applications and Services*.

Breiman, L. et al., 1984. *Classification and Regression Trees*, Chapman and Hall/CRC.

Briscoe, T., 2011. *Introduction to Linguistics for Natural Language Processing*, University of Cambridge.

Brock, G. et al., 2008. clValid: An R Package for Cluster Validation. *Journal of Statistical Software*, 25(4), pp.1–22. Available at: http://www.jstatsoft.org/v25/i04.

Burges, C. et al., 2005. Learning to Rank using Gradient Descent. *Proceedings of the 22nd international conference on Machine learning*, 89-96.

Chang, H.-C. & Hsu, C.-C., 2005. Using Topic Keyword for Clusters for Automatic Document Custering. *3rd International Conference on Information Technology and Applications*.

Chen, C.-M., 2010. Intelligent location-based mobile news service system with automatic news summarization. *Expert Systems with Applications*, 37, pp.6651–6662.

Cingiz, M.O. & Diri, B., 2012. Content Mining of Microblogs. *IEEE/ACM International Conference on Advances in Social Networks Analisys and Mining*.

CRAN, 2014. Biblioteca TM. Available at: http://cran.r-project.org/web/packages/tm/index.html.

Dalianis, H., 2000. SweSum – a text summarizer for Swedish.

Das, D. & Martins, A.F.T., 2007. A survey on Automatic Text Summarization.

Davies, D.L. & Bouldin, D.W., 1979. A Cluster Separation Measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-1(2), pp.224–227.

Dempster, A.P., Laird, N.M. & Rubin, D.B., 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39, pp.1–38.

Dilrukshi, I., De Zoysa, K. & Caldera, A., 2013. Twitter News Classification Using SVM. *The 8th International Conference on Computer Science & Education*.

Donoho, D.L., 2000. Aide-Memoire. High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality.

Dunn, J.C., 1974. Well separated clusters and optimal fuzzy-partitions. *Journal of Cybernetics*, 4, pp.95–104.

Escudeiro, N., 2012. *Semi-automatic classification: using active learning for efficient class coverage*. Faculdade de Engenharia da Universidade do Porto.

Evans, D.K. & Klavans, J.L., 2005. Columbia Newsblaster: Multilingual News Summarization on the Web. *HLT-NAACL*.

Fattah, M.A. & Ren, F., 2008. Automatic Text Summarization. *World Academy of Science, Engineering and Technology*, 2.

Frank, E., 2000. *Pruning Decision Trees and Lists*. The University of Waikato.

Gabriel, L., 2009. *Automatic Email Organization*. Instituto Superior de Engenharia do Porto.

Gomes, H., Neto, M. & Henriques, R., 2013. Text Mining: Analise de Sentimentos na classificaçao de noticias. *8th Iberian Conference on Information Systems and Technologies*.

Guha, S., Rastogi, R. & Shim, K., 1999. ROCK: A Robust Clustering Algorithm for Categorical Attributes. *15th International Conference on Data Engineering*, pp.512–521.

Guha, S. & Shim, R.R.K., 1998. CURE: An Efficient Clustering Algorithm for Large Databases. *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*.

Gupta, M.R. & Chen, Y., 2011. Theory and Use of the EM Algorithm. *Foundations and Trends in Signal Processing*, 4, pp.223–296.

Han, J. & Kamber, M., 2006. *Data Mining: Concepts and Techniques*, Morgan Kaufmann.

Harabagiu, S.M. & Lacatusu, F., 2002. Generating Single and Multi-Document Summaries with GISTEXTER. *Proceedings of the Workshop on Automatic Summarization*.

Hassler, M. & Fliedl, G., 2006. Text Preparation through Extended Tokenization. *Data Mining VII: Data, Text and Web Mining and their Business Applications (DATA)*, pp.13–21. Available at: http://hekkas.com/research/contribution/downloads/pub04_ExtendedTokenization.pdf.

Hollander, G. de & Marx, M., 2011. Summarization of meetings using word clouds. In *Computer Science and Software Engineering (CSSE), 2011 CSI International Symposium on*. pp. 54–61.

Hotho, A., Nurnberger, A. & Paab, G., 2005. A Brief Survey of Text Mining.

IDC, 2014. *The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things*, EMC Corporation. Available at: http://www.emc.com/collateral/analyst-reports/idc-digital-universe-2014.pdf.

Joachims, T., 1998. Text Categorization with Suport Vector Machines: Learning with Many Relevant Features. *Proceedings of the 10th European Conference on Machine Learning*, pp.137–142.

Karypis, G., Han, E.-H. & Kumar, V., 1999. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. *IEEE Computer*, pp.68–75.

Kass, G.V., 1980. An exploratory technique for investigating large quantities of categorical data. *Journal of the Royal Statistical Society. Series C - Applied Statistics*, 2, pp.119–127.

Kaur, M. & Kaur, N., 2013. Web Document Clustering Approaches Using K-Means Algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3.

Kwac, J., 2014. *Adaptive Kmeans algorithm based on threshold*, Stanford. Available at: http://cran.us.r-project.org/web/packages/akmeans/akmeans.pdf.

Lee, J., 1994. Properties of extended Boolean models in information retrieval. *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pp.182–190.

Lin, C.-Y., 2004. ROUGE: A Package for Automatic Evaluation of Summaries. *Proceedings of the Workshop on Text Summarization Branches*.

Lin, C.-Y. & Hovy, E., 2002. Automated multi-document summarization in NeATS. *Proceedings of the 2nd international conference on Human Language Technology Research*, pp.59–62.

Liu, X., 2014. *Natural Language Processing*, Center for Natural Language Processing in the School of Information Studies at Syracuse University. Available at: http://surface.syr.edu/cgi/viewcontent.cgi?article=1043&context=istpub.

Liu, X., Ruhia, G. & Liufu, S., 2012. Internet News Headlines Classification Method Based On The N-Gram Language. *International Conference on Computer Science and Information Processing*.

Lloret, E., 2009. Text Summarization: An Overview. Available at: http://www.dlsi.ua.es/ elloret/publications/TextSummarization.pdf.

Macqueen, J., 1967. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*. pp. 281–297.

Manning, C.D., Raghavan, P. & Schütze, H., 2008. *Introduction to Information Retrieval*, Cambridge University Press.

Mitkov, R. & Hovy, E., 2005. The Oxford Handbook of Computational Linguistics, chapter 32. In Oxford University Press, pp. 583–598.

Mooney, R., 2014. *CS 343: Artificial Intelligence Natural Language Processing*, University of Texas at Austin. Available at: http://www.cs.utexas.edu/ mooney/cs343/slide-handouts/nlp.pdf.

Nenkova, A. & McKeown, K., 2011. *Automatic Summarization*, Now Publishers Inc.

Neto, J.L., Freitas, A.A. & Kaestner, C.A.A., 2002. Automatic Text Summarization using a Machine Learning Approach. *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence*, pp.205–215.

Nogueira, B. et al., 2008. Winning some of the document preprocessing challenges in a text mining process. *IV Workshop em Algoritmos e Aplicações de Mineração de Dados da Universidade de Sao Paulo*, pp.10–18.

Orengo, V. & Huyck, C., 2001. A stemming algorithm for the portuguese languagem. *Proceedings of the 8th International Symposium on String Processing and Information Retrieval*, pp.186–193.

Pinto, C.M.S., 2005. *Algoritmos Incrementais para Aprendizagem Bayesiana*. Faculdade de Economia da Universidade do Porto.

Porter, M.F., 1980. An algorithm for suffix stripping.

Prakash, V.J. & Dr. Nithya, L.M., 2014. A Survey On Semi-Supervised Learning Techniques. *International Journal of Computer Trends and Technology*.

Preradovic, N., Ljubesic, N. & Boras, D., 2010. Croatian Web Text Summarizer (CroWebSum). *Proceedings of the 32nd International Conference on Information Technology Interfaces*.

Quinlan, J.R., 1993. *C4.5: programs for machine learning*, Morgan Kaufmann Publishers.

Quinlan, J.R., 1986. Induction of Decision Trees. *Machine Learning*, 1, pp.81–106.

Radev, D.R., Blair-Goldensohn, S. & Zhang, Z., 2001. Experiments in single and multidocument summarization using MEAD. *1st Document Understanding Conference*.

Radev, D.R., Fan, W. & Zhang, Z., 2001. WebInEssence: a personalized web-based multi-document summarization and recommendation system. *NAACL Workshop on Automatic Summarization*.

Rafsanjani, M.K., Varzaneh, Z.A. & Chukanlo, N.E., 2012. A survey of hierarchical clustering algorithms. *The Journal of Mathematics and Computer Science*, 5, pp.229–240.

Ramasubramanian, C. & Ramya, R., 2013. Effective Pre-Processing Activities in Text Mining using Improved Porter's Stemming Algorithm. *International Journal of Advanced Research in Computer and Communication Engineering*, 2.

Refaeilzadeh, P., Tang, L. & Liu, H., 2009. *Encyclopedia of Database Systems: Cross-Validation*, Springer.

Reis, L.F.M. dos, 2012. *Sistema de Recomendação Baseado em Conhecimento*. Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Rendón, E. et al., 2011. Internal versus External cluster validation indexes. *International Journal of Computers and Communications*, 5, pp.27–34.

Robertson, S.E. & Jones, K.S., 1976. Relevance Weighting of Search Terms. *Jounal of the American Society for Information Science*, 3, pp.129–146.

Rocchio, J.J., 1971. Relevance feedback in information retrieval. In G. Salton, ed. *The Smart retrieval system - experiments in automatic document processing*. Englewood Cliffs, NJ: Prentice-Hall, pp. 313–323.

Rodrigues, F., Descoberta de Conhecimento - Clustering. Available at: https://moodle.isep.ipp.pt/file.php/234519/DESCO-09-Clustering.pdf.

Rodrigues, F., 2013. *Descoberta de Conhecimento - Text Mining*, Departamento de Engenharia Informática (DEI/ISEP).

Rokach, L. & Maimon, O., 2005. Top-Down Induction of Decision Trees Classifiers – A Survey. *IEEE Transactions on Systems, Man, and Cybernetics*, 1, pp.476–487.

Rousseeuw, P., 1987. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *J. Comput. Appl. Math.*, 20(1), pp.53–65. Available at: http://dx.doi.org/10.1016/0377-0427(87)90125-7.

Rumelhart, D., Hinton, G. & Williams, R., 1986. Learning representations by back-propagating errors. *Nature*, 323, pp.533–536.

Sajjanhar, A. & Zhao, Y., 2012. Web Service to Deliver Filteres RSS Items to a Mobile Application. *7th ChinaGrid Annual Conference*.

Salton, G., Wong, A. & Yang, C.S., 1975. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 8.

Sanwaliya, A., Shanker, K. & Misra, S., 2010. Categorization of News Articles: A Model based on Discriminative Term Extraction Method. *2nd International Conference on Advances in Databases, Knowledge, and Data Applications*.

Schneider, J. & Moore, A.W., 2000. *A Locally Weighted Learning Tutorial Using Vizier 1.0*, Carnegie Mellon University, the Robotics Institute. Available at: http://books.google.pt/books?id=E975GgAACAAJ.

Sebastiani, F., 2002. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*.

Sebastiani, F., 2005. Text categorization. In *Text Mining and its Applications to Intelligence, CRM and Knowledge Management*. WIT Press, pp. 109–129.

Settles, B., 2009. *Active Learning Literature Survey*, University of Wisconsin.

Silva, C., Osorio, F. & Vieira, R., 2005. Evaluating the useof linguisticinformation in the preprocessing phase of Text Mining. *Revista Iberoamericana de Inteligencia Artificial*, 9, pp.59–66.

Sogaard, A., 2010. Simple semi-supervised training of part-of-speech taggers. *Proceedings of the ACL 2010 Conference Short Papers*.

Spoustova, D. et al., 2009. Semi-supervised Training for the Averaged Perceptron POS Tagger. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*.

Steinbach, M., Karypis, G. & Kumar, V., 2000. A comparison of document clustering techniques.

Sureka, R. & Kong, H., 2006. Automated Trainable Summarizer for Financial Documents. *10th International Enterprise Distributed Object Computing Conference Workshop*.

Svore, K.M., Vanderwende, L. & Burges, C.J.C., 2007. Enhancing Single-document Summarization by Combining RankNet and Third-party Sources. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Tkachenko, M. & Simanovsky, A., 2012. Named Entity Recognition: Exploring Features. *Proceedings of KONVENS*.

Toutaniva, K. et al., 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 1, pp.173–180.

Tsai, Y.-F. & Chen, K.-J., 2003. Context-rule Model for Pos Tagging. *Proceedings of the 17th Pacific Asia Conference on Language, Information and Computation*.

Ward, J.H., 1963. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58(301), pp.236–244. Available at: http://www.jstor.org/stable/2282967.

Weiss, S. et al., 2010. *Text Mining: Predictive Methods for Analyzing Unstructured Information*, Springer.

Witten, I., 2005. Practical handbook of internet computing. In M. P. Singh, ed. Chapman & Hall/CRC, pp. 14–22. Available at: http://www.cs.waikato.ac.nz/ ihw/papers/04-IHW-Textmining.pdf.

Wu, C.-W. & Liu, C.-L., 2003. Ontology-based Text Summarization for Business News Articles. *Proceedings of the ISCA Eighteenth International Conference on Computers and Their Applications*, pp.389–392.

Xu, Y. et al., 2008. *An Extended Document Frequency Metric for Feature Selection in Text Categorization*, Springer.

Yi, Y. et al., 2012. Machine Learning Algorithms with Co-occurrence based Term Association for Text Mining. *4th International Conference on Computational Intelligence and Communication Networks*.

Yu, L. & Ren, F., 2009. A Study on Cross-Language Text Summarization Using Supervised Methods. *International Conference on Natural Language Processing and Knowledge Engineering*.

Yuan, M., Ouyang, Y. & Xiong, Z., 2013. A Text Categorization Method using Extended Vector Space Model by Frequent Term Sets. *Journal of Information Science and Engineering*, 29, pp.99–144.

Zhang, T., Ramakrishnan, R. & Livny, M., 1996. BIRCH: an efficient data clustering method for very large databases. *Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, pp.103–114.

11 Anexos

ANEXO 1 – Índices de Dunn do ak-*means*

Threshold	Castigos - DUNN	Geral - DUNN	Pré-Jogo - DUNN	Lesões - DUNN	Mercado - DUNN	Óbitos - DUNN	Resultados - DUNN	Treinadores - DUNN
0,025	0,098	0,038	0,227	0,187	0,299	0,176	0,101	0,233
0,05	0,133	0,010	0,243	0,164	0,304	0,179	0,095	0,241
0,075	0,127	0,011	0,230	0,222	0,294	0,160	0,093	0,171
0,1	0,130	0,011	0,235	0,212	0,288	0,158	0,101	0,264
0,125	0,134	0,011	0,197	0,170	0,304	0,152	0,100	0,060
0,15	0,074	0,012	0,226	0,239	0,317	0,171	0,104	0,232
0,175	0,094	0,011	0,249	0,218	0,304	0,133	0,120	0,240
0,2	0,078	0,012	0,050	0,184	0,277	0,065	0,082	0,261
0,225	0,070	0,012	0,230	0,182	0,239	0,125	0,109	0,254
0,25	0,132	0,012	0,245	0,200	0,290	0,082	0,090	0,248
0,275	0,053	0,010	0,214	0,184	0,315	0,172	0,094	0,113
0,3	0,089	0,013	0,236	0,171	0,257	0,111	0,100	0,192
0,325	0,089	0,011	0,220	0,174	0,307	0,116	0,083	0,172
0,35	0,078	0,012	0,213	0,195	0,261	0,168	0,055	0,255
0,375	0,093	0,012	0,241	0,183	0,219	0,108	0,057	0,151
0,4	0,077	0,012	0,259	0,180	0,257	0,123	0,109	0,236
0,425	0,095	0,012	0,209	0,213	0,308	0,191	0,117	0,195
0,45	0,241	0,013	0,209	0,155	0,315	0,116	0,060	0,142
0,475	0,267	0,012	0,119	0,187	0,282	0,121	0,060	0,254
0,5	0,172	0,012	0,232	0,145	0,259	0,122	0,084	0,149

ANEXO 2 – Índices de Dunn do ak-*means* (Apenas entidades)

Threshold	Castigos - DUNN	Geral - DUNN	Pré-Jogo - DUNN	Lesões - DUNN	Mercado - DUNN	Óbitos - DUNN	Resultados - DUNN	Treinadores - DUNN
0,025	0,121	0,194	0,110	0,240	0,315	0,141	0,091	0,252
0,05	0,096	0,106	0,224	0,221	0,314	0,181	0,111	0,260
0,075	0,100	0,100	0,110	0,191	0,314	0,176	0,118	0,269
0,1	0,100	0,171	0,107	0,228	0,229	0,103	0,092	0,250
0,125	0,068	0,141	0,223	0,214	0,298	0,177	0,073	0,213
0,15	0,094	0,119	0,241	0,202	0,241	0,163	0,082	0,217
0,175	0,095	0,091	0,112	0,209	0,315	0,088	0,100	0,282
0,2	0,056	0,182	0,235	0,226	0,325	0,102	0,074	0,250
0,225	0,127	0,113	0,241	0,220	0,278	0,120	0,060	0,209
0,25	0,070	0,190	0,112	0,204	0,304	0,082	0,079	0,305
0,275	0,124	0,143	0,235	0,199	0,302	0,116	0,118	0,243
0,3	0,061	0,113	0,214	0,139	0,137	0,091	0,084	0,138
0,325	0,107	0,075	0,237	0,222	0,306	0,091	0,138	0,252
0,35	0,072	0,144	0,237	0,199	0,242	0,169	0,105	0,208
0,375	0,110	0,192	0,227	0,196	0,301	0,123	0,061	0,258
0,4	0,247	0,119	0,227	0,172	0,262	0,164	0,091	0,140
0,425	0,176	0,152	0,235	0,224	0,305	0,160	0,078	0,243
0,45	0,070	0,113	0,112	0,184	0,253	0,083	0,061	0,266
0,475	0,176	0,113	0,163	0,166	0,308	0,125	0,091	0,164
0,5	0,080	0,144	0,247	0,214	0,258	0,094	0,062	0,306

ANEXO 3 – Distribuição dos modelos de avaliação manual por avaliador

	0,125	0,2	0,225	0,25	0,325	0,35
Castigos						
Geral						
Pré-Jogo			Diogo Garcia			Diogo Garcia
Lesões		Pedro Henriques				
Mercado		Joaquim Silva				
Óbitos	Daniel Sousa					Daniel Sousa
Resultados					Nélson Sousa	
Treinadores				Videira		

	0,4	0,425	0,45	0,475	0,5
Castigos	Emanuel Sousa		Nélson Sousa		
Geral					
Pré-Jogo			Bruno Costa		André Almeida
Lesões		Tomas Sotomaior			
Mercado					
Óbitos			Tiago Pina		
Resultados			Tomás Sotomaior		
Treinadores			Samuel Rodrigues		