NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTAMENTO DE INFORMÁTICA

SERHIY HULEVYCH

Licenciado em Ciência e Engenharia Informática

IDENTIFICAÇÃO DE CLASSES EM TEXTO

CLASSIFICAÇÃO NÃO SUPERVISIONADA

MESTRADO EM ENGENHARIA INFORMÁTICA

Universidade NOVA de Lisboa
Novembro, 2021

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTAMENTO DE INFORMÁTICA

IDENTIFICAÇÃO DE CLASSES EM TEXTO

CLASSIFICAÇÃO NÃO SUPERVISIONADA

SERHIY HULEVYCH

Licenciado em Ciência e Engenharia Informática

Orientador: Joaquim Francisco Ferreira da Silva
Faculdadede Ciências e Tecnologia da Universidade Novade Lisboa

MESTRADO EM ENGENHARIA INFORMÁTICA

Universidade NOVA de Lisboa
Novembro, 2021

Agradecimentos

A realização desta dissertação de mestrado não seria possível sem a presença e o apoio de várias pessoas importantes.

Quero começar por agradecer ao Professor Joaquim Silva, meu orientador, que me acompanhou ao longo da tese e que sempre esteve disponível para esclarecer dúvidas e discutir ideias, principalmente durante uma fase complicada de pandemia mundial.

Um agradecimento ao professor João Lourenço que forneceu o template para a escrita desta dissertação.

Quero também agradecer aos meus colegas Afonso Almeida, Bernardo Oliveira, Miguel Alves, Rui Fernandades e Tiago Santos que me acompanharam e apoiaram desde o ínicio.

Por fim, quero agradecer especialmente à minha mãe, ao meu pai e ao meu irmão, a quem dedico este trabalho, por terem estado sempre presentes, pelo apoio incondicional e por todos os sacrificios que fizeram para eu que pudesse concluir esta etapa da minha vida.

Resumo

Na classificação de documentos, são vários os estudos já realizados, sobretudo através da classificação supervisionada. Existem em menor número também alguns que usam classificação não supervisionada.

Na classificação supervisionada, tendo cada documento a etiqueta (*label*) correspondente à classe/tópico a que pertence, está facilitado o processo de classificação, o que permite em geral melhores resultados, em termos de Precisão e Recall, quando comparados com os obtidos pela opção 'não supervisionada'. No entanto, existe uma limitação forte: a classificação de novos elementos está limitada às classes indicadas na fase de treino através da etiqueta, sendo que o sistema não consegue aprender novas classes a não ser por essa indicação explícita.

Considerando a alternativa da classificação não supervisionada, onde não existe a indicação explícita da classe, o desafio consiste sobretudo em detetar/minerar que grupos/classes de tópicos principais estão implícitos nos dados, isto é, nos documentos caracterizados pelos seus atributos (*features*). Desta forma poder-se-ão aprender de forma dinâmica novas classes, desde que estejam implícitas nos dados, isto é, desde que as *features* sejam suficientemente caracterizadoras.

Um dos objetivos desta dissertação foi a elaboração de um sistema capaz de receber um conjunto de documentos e agrupá-los por tópicos, tendo em conta o seu conteúdo. Um segundo objectivo consistiu em identificar os tópicos/subtópicos principais de cada grupo e também classificar novos documentos de acordo com o que foi aprendido na fase de treino. O trabalho envolveu a selecção e redução de *features*, a construção dos grupos (*clustering*) e a classificação propriamente dita.

Palavras-chave: Classificação de Documentos, Classificação não Supervisionada, Classificação Supervisionada, Clustering

Abstract

In document classification, there are several studies that have been done, mostly using the supervised classification. There are also some approaches using the unsupervised classification.

In supervised classification, with each document having the label corresponding to the class / topic to which it belongs, the classification process is facilitated, which generally allows better results, in terms of Precision and Recall, when compared with those chosen by the option “unsupervised". However, there is a strong limitation: the classification of new elements is limited to the classes indicated in the training phase through the label, and the system is unable to learn new classes except for this explicit indication.

Based on the alternative of unsupervised classification, where there is no explicit indication of the class, the challenge consists mainly in detecting/mining which groups/classes of main topics are implicit in the data, in other words, in the documents characterized by their attributes. In this way, new classes can be dynamically learned, as long as they are implicit in the data, in other words, as long as the features are sufficiently characterizing.

One of the goals of this dissertation was the development of a system capable of receiving a set of documents and group them by topics, based on their content. Another goal was to identify topics/subtopics of each group and also classify new documents according to what was learned in the training phase. The work involved the selection and reduction of features, the construction of groups (clustering) and a classification itself.

Keywords: Documents Classification, Supervised Classification, Unsupervised Classification, Clustering

Índice

Índice de Figuras

Índice de Tabelas

1

Introdução

A classificação de documentos é uma área que tem vindo a ser estudada cada vez mais ao longo dos anos, podendo separar-se em dois tipos de classificações: supervisionada (aprendizagem supervisionada) e não supervisionada (aprendizagem não supervisionada).

A aprendizagem supervisionada é um método onde, a partir de um conjunto de treino, se pretende criar uma função que possa ser usada para mapear novos exemplos. Este método é dividido em várias etapas. O primeiro passo consiste em escolher quais os dados que vão ser usados como conjunto de treino. No caso da classificação de documentos há que escolher quais as palavras, termos ou outros elementos associados aos textos, com poder discriminante e caracterizador dos documentos. Este é um passo muito importante, uma vez que a solução desenvolvida para classificar depende grandemente desta escolha. De seguida, é necessário reunir um conjunto de objetos, neste caso um conjunto de documentos e suas classes, que seja representativo do mundo real. Os dados são então transformados num vetor de *features*, que passará a representá-los. Se o número de *features* for demasiado grande, poderá ser necessário aplicar um método de redução de *features* (dimensões), tentando sempre minimizar a perda de informação em relação aos dados originais.

O próximo passo é escolher o algoritmo classificador que será usado (ex: SVM, entre outros) e corrê-lo usando o conjunto de treino que foi escolhido. Alguns algoritmos requerem que o utilizador indique certos parâmetros, pelo que a melhor maneira de os ajustar é otimizando a performance do algoritmo com o recurso a conjuntos de treino e validação. Por fim, após a escolha dos parâmetros e a realização da fase de treino, é feita uma avaliação final da Precisão e *Recall* usando um conjunto de testes que não fez parte da fase de treino. A partir daí podem ser feitas classificações de novos elementos.

A aprendizagem supervisionada é maioritariamente o processo escolhido por ter melhores resultados em termos de Precisão e *Recall*. No entanto, existe uma desvantagem evidente que é estar limitada às classes conhecidas no conjunto de treino.

Ao contrário da aprendizagem supervisionada, a não supervisionada deverá conseguir aprender padrões sem que haja qualquer tipo de informação à priori em relação à classe de cada amostra, o que no caso dos documentos se traduz no desconhecimento do tópico a que pertence cada documento. Mesmo com resultados normalmente inferiores em relação ao outro tipo de aprendizagem (supervisionada), tem a vantagem de não depender da língua ou uma qualquer lista prévia de tópicos típicos associados a documentos, se não forem usadas informações morfossintáticas. Neste caso, a identificação dos tópicos é feita por detecção de padrões estatísticos associados aos termos que fazem parte dos documentos. Para além disso, caso sejam usadas ferramentas estritamente estatísticas, sem recorrer portanto a elementos morfossintáticos associados à especificidade de cada língua, a abordagem não supervisionada torna-se mais dinâmica por poder identificar novos tópicos e em diferentes línguas.

Esta dissertação focou-se na classificação não supervisionada de textos não estruturados.

1.1 Problema/Motivação

A classificação não supervisionada dos documentos deveria idealmente poder identificar o(os) tópicos(os) dum qualquer documento a classificar. Não é uma tarefa fácil, tendo em conta que não se dispõe de etiquetas relativas ao tópico/assunto principal de que trata cada amostra de treino (documento). Em consequência disso, perante um conjunto de documentos a usar como treino, não se sabem à partida a quantos grupos/*clusters* corresponde esse conjunto. Desta forma, a abordagem não supervisionada terá que construir um *clustering* (configuração de *clusters*) que seja verossímil tendo em conta o conteúdo do conjunto de treino, de maneira a que posteriormente sejam classificados novos documentos com boa Precisão, tendo em conta o que foi aprendido na fase de treino. Um dos maiores problemas associados está na escolha das *features* que permitam a deteção de padrões próprios de cada classe, classe que não é explícita nos documentos. Assim, as palavras isoladas ou em grupo existentes nos documentos surgem como potenciais atributos caracterizadores. Porém, estes são em grande número e nem todos têm a mesma capacidade para discriminar, o que constitui um desafio nesta dissertação.

As dificuldades associadas a este tipo de classificação estão na base dos resultados obtidos serem normalmente mais modestos do que na classificação supervisionada. Talvez essa seja a razão pela qual existem menos trabalhos desenvolvidos do que na versão supervisionada.

Tendo em conta que na classificação não supervisionada nada se sabe acerca quer da classe dos documentos quer do conteúdo pricipal de cada um, nesta dissertação tornou-se uma motivação extra, conseguir desenvolver uma abordagem que revelasse qual o conteúdo dos *clusters*.

1.2 Objetivos

- O principal objetivo desta dissertação foi construir um sistema que fosse capaz de agrupar documentos sem que o utilizador tivesse de fornecer o número de grupos pretendido e, fosse capaz de indicar o conteúdo dos mesmos, ou seja, quais são os tópicos principais dos *clusters*. Tendo em conta que não sabemos à priori quais os tópicos dos documentos, pretendiamos que o sistema fosse o mais preciso possível, de modo a que os tópicos resultantes de cada *cluster*, coincidissem realmente com os verdadeiros assuntos dos documentos que formam cada grupo. Para além do agrupamento que se pretendia, que viesse a ser de boa qualidade, queríamos que a fase de classificação produzisse resultados com a melhor Precisão e *Recall* possíveis.

- Como já foi referido, um dos grandes desafios nesta tese, está na escolha dos atributos/*features* que deverão caracterizar os documentos, de forma que estas possam discriminar e identificar diferentes grupos. Esta tarefa, embora aparentemente não fosse um dos objetivos finais da dissertação, sendo um passo de difícil resolução, tornou-se também ele um objetivo a atingir.

- De modo a alcançar o que se pretendia, tivemos de explorar e encontrar as melhores opções para as várias etapas, tais como: i) a extração e redução de *features*; ii) a escolha do algoritmo a usar para o *clustering* dos documentos; iii) a abordagem de classificação a usar, de entre as implementações disponíveis, tendo em conta a natureza e distribuição dos dados caracterizadores dos documentos.

Mais à frente, no capítulo 3, iremos falar mais detalhadamente da solução que desenvolvemos e quais as etapas a seguir.

1.3 Estrutura do documento

Esta dissertação divide-se em 5 capítulos distintos.

Neste primeiro capítulo, é feita uma introdução ao tema da classificação de documentos, é explicado o problema com que nos deparamos e falamos ainda dos objetivos a atingir.

No segundo capítulo, descrevemos o trabalho relacionado que está dividido em 2 partes: as ferramentas relacionadas que incluem definições e funcionamento de técnicas e algoritmos que são mencionados ou que poderiam ser úteis no contexto da tese e; o estado da arte, onde analisamos os vários estudos realizados no âmbito da classificação não supervisionada de documentos e também supervisionada.

No terceiro capítulo, é apresentada a nova solução de classificação não supervisionada de documentos nas várias etapas que estão envolvidas.

O capítulo 4 contém os resultados das experiências que foram realizadas para diferentes conjuntos de documentos, onde são reportadas as avaliações necessárias.

Por fim, no capítulo 5, apresentamos as conclusões que resultaram no contexto da solução desenvolvida, bem como as melhorias possíveis a realizar no futuro.

2

Trabalho Relacionado

2.1 Ferramentas relacionadas

No contexto da classificação supervisionada ou não supervisionada, existem ferramentas úteis para este processo. Seguem-se os grupos principais onde se enquadram essas ferramentas.

2.1.1 Pré-processamento dos textos

Lematização

É o processo de deflexionar uma palavra para determinar o seu lema, ou por outras palavras, é a determinação da forma canónica de cada vocábulo. Por exemplo as palavras 'gato', 'gata', 'gatos', 'gatas' são todas formas do mesmo lema: 'gato'. Este processo é útil nos contextos onde é mais importante a identificação semântica da raiz das palavras e o seu número de ocorrências, do que as suas formas flexionadas. No caso dos termos semanticamente relevantes, esta transformação pode ajudar no processo de discriminação de tópicos com vista à construção de *clusters*.

Stemming

É uma transformação do texto, parecida com a lematização mas com um objetivo um pouco diferente. Neste processo transforma-se cada palavra flexionada pela sua raíz. Por exemplo na transformação das palavras *cats*, *catlike* e *catty*, obter-se-ia sempre a raiz *cat*, que neste caso, coincide com uma palavra. No entanto em casos como 'aluno', 'alunos', 'aluna', 'alunas', resultaria em *alun*, que não coincide com uma palavra. O algoritmo *Porter* é um *stemmer* vocacionado para a língua Inglesa [15].

2.1.2 Avaliação dos termos em texto

Tf-Idf (term frequency – inverse document frequency)

O $Tf–Idf$ é uma medida estatística que tem o intuito de indicar a importância de uma palavra num documento em relação a uma coleção de documentos ou num *corpus* linguístico (conjunto de documentos). O valor $Tf–Idf$ de uma palavra aumenta proporcionalmente à medida que o número de ocorrências da mesma no documento aumenta, no entanto, esse valor é equilibrado pela frequência da palavra no *corpus*. Esta métrica penaliza as palavras que surgem poucas vezes no documento e/ou em muitos dos documentos da coleção. A medida é bastante usada para selecionar as palavras semanticamente mais relevantes, o que pode ser importante para identificar tópicos de documentos.

Assim, o valor de $Tf–Idf$ do termo t no documento d pertencente ao conjunto de documentos D, é calculado da seguinte maneira:

$$Tf–Idf(t,d) = tf(t,d) \times idf(t,D) \tag{2.1}$$

$$tf(t,d) = f(t,d) \tag{2.2}$$

sendo $f(t,d)$ a frequência do termo t no documento d, e

$$idf(t,D) = \log\left(\frac{||D||}{||d \in D : tf(t,d) > 0||}\right) \tag{2.3}$$

Tf-Df (term frequency – document frequency)

Esta métrica é definida por:

$$Tf–Df(t) = (n_1 \times n_2 + c(n_1 \times n_3 + n_2 \times n_3)) \tag{2.4}$$

Sendo n_1 o número de documentos onde o termo t não ocorre; n_2 o número de documentos onde ocorre uma vez só; n_3 onde ocorre 2 ou mais vezes. c é um parâmetro que dá peso relativo aos casos de 1 ou mais ocorrências.

Esta métrica valoriza os termos de maior variância de ocorrência ao longo dos documentos.

TS (Term Strength)

A métrica *Term Strength*, $s(t)$, é definida como:

$$s(t) = \frac{||(d_x, d_y) \in D : t \in d_x \ \ \& \ \ t \in d_y||}{||(d_x, d_y) \in D : t \in d_x||} \tag{2.5}$$

Sendo t o termo, d_x um documento e d_y outro documento. São tomados em conta todos os possiveis pares (d_x, d_y) de documentos do *corpus*. Esta métrica tem como propósito valorizar os termos que surgem simultaneamente num maior número de pares de documentos.

TC (Term Contribution)

Esta métrica pretende valorizar os termos que surgem simultaneamente em muitos pares de documentos e com grande frequência em cada um. É definida como:

$$TC(t) = \sum_{i,j \in D} f(t, d_i) \times f(t, d_j) \tag{2.6}$$

Onde t representa o termo, e d o documento pertencente ao conjunto de documentos D, ou seja, *f(t,d)* quer dizer a frequência do termo t no documento d.

2.1.3 Redução de complexidade

PCA (Principal Component Analysis)

Esta é uma técnica estatística usada para redução de *features*. Normalmente, as *features* originais estão correlacionadas entre si, correspondendo a eixos não ortogonais num espaço vectorial. Com a aplicação do PCA, são geradas novas *features* com base nas originais, mas agora ortogonais entre si, permitindo uma redução por vezes muito significativa no número de *features* sem que se perca informação apreciável contida originalmente. É uma ferramenta útil no contexto do *clustering* de documentos, dada a quantidade normalmente grande de *features* originais associadas ao texto.

FastICA

O algoritmo FastICA é um método eficaz de realizar a estimativa ICA, a qual consiste numa redução linear da dimensão, através da transformação dos dados em colunas (features) que representam componentes independentes. Esta é uma técnica importante na classificação de documentos uma vez que a distribuição dos dados neste caso se pode afastar da distribuição Gaussiana. No entanto, surge como problema, a necessidade de indicar o número de componentes que se pretende utilizar.

T-Distributed Stochastic Neighbor Embedding

Técnica que consiste na redução não linear da informação descrita por um número elevado de dimensões. É útil para incorporar dados descritos por muitas dimensões para que possamos ter uma visualização dos mesmos num espaço de *baixa* dimensão, por norma duas ou três dimensões.

Mais especificamente, cada objeto descrito inicialmente por muitas dimensões, é modelado por um ponto num espaço de poucas dimensões (duas ou três) de tal modo que objetos semelhantes sejam modelados por pontos próximos um do outro e, objetos diferentes sejam modelados por pontos distantes.

Este algoritmo consiste em 2 etapas. Primeiro, constrói-se a distribuição de probabilidade sobre pares de objetos de *altas* dimensões de tal modo que os objetos semelhantes tenham maior probabilidade e objetos diferentes menor probabilidade. Na segunda fase, é definida uma distribuição de probabilidade semelhante para os pontos da menor dimensão. O objetivo é minimizar a divergência de *Kullback–Leibler* entre as duas distribuições.

No contexto da classificação de documentos onde tendencialmente existem muitas dimensões, esta técnica pode ser uma ferramenta útil.

IB (Information Bottleneck)

Esta técnica consiste no estabelecimento de um compromisso que tem por objetivo reduzir a complexidade dum modelo sem perder a qualidade dos seus resultados, sendo usada em vários contextos, por exemplo em *deep learning* aplicado ao *clustering* de documentos.

SVD (Singular Value Decomposition)

Tal como a técnica PCA, também esta (SVD) permite a redução de *features*. Consiste na fatorização da matriz inicial em vectores singulares e valores singulares. Esta técnica tem diversas aplicações importantes em processamento de sinais e estatística, podendo vir a ser usada no contexto que nos interessa.

2.1.4 Tópicos e documentos – ferramentas

LDA (Latent Dirichlet Allocation)

A abordagem LDA consiste num modelo probabilístico generativo que se aplica a *corpus* de texto. A ideia básica sustenta que os documentos são representados por misturas aleatórias sobre tópicos latentes, onde cada tópico é caraterizado por uma distribuição tendo em conta o universo das palavras do *corpus*. O número de tópicos tem normalmente que ser indicado. Para detalhes sobre esta técnica, consultar [1].

Word2Vec

É uma técnica que, a partir de um grande conjunto de palavras dum *corpus*, produz um espaço vetorial (multidimensional). Cada palavra única do *corpus* é representada por um vetor de números, onde palavras relacionadas irão ter vetores semelhantes e por sua vez, próximos um do outro, no espaço. Para mais detalhes sobre esta técnica, consultar [13].

Doc2Vec

O Doc2Vec é considerada uma generalizaçao do Word2Vec. Tal como o Word2Vec, é uma ferramenta do processamentos de linguagem natural. No entanto, em vez de representarmos as palavras únicas do *corpus* como um vetor de números, os vetores representam cada um dos documentos.

2.1.5 Plataformas semânticas

WordNet

É uma base de dados lexical de relações semântica entre palavras, em mais de 200 línguas. WordNet vincula as palavras a relações semânticas, incluindo sinónimos, hipónimos e merónimos. Pode ser visto como uma combinação e extensão de um dicionário e tesauro. O seu principal uso tem sido na area da inteligência artificial.

No contexto desta dissertação pode ser útil, mas terá certamente um alcance limitado pois não se adapta ao conteúdo de documentos com tópicos que sejam desconhecidos para esta plataforma (WordNet).

LCSH (Library of Congress Subject Headings)

Consiste numa lista de palavras e frases mantida pela Biblioteca do Congresso dos EUA que é usada para indicar os tópicos dos recursos da biblioteca. A LCSH garante a consistência das coleções da biblioteca. Por exemplo, quando existem palavras ou expressões diferentes mas que se referem ao mesmo, umas dessas palavras/expressões é escolhida como tópico e as restantes referenciam-na.

Apesar de antiga, a LCSH está constantemente a ser atualizada, tornando-se portanto numa ferramenta útil para vários estudos. No entanto, no contexto da presente dissertação, esta ferramenta tem uma aplicação limitada já que, para além de estar restrita às designações em Inglês, não é, por assim dizer, uma lista dinâmica, pois não se adapta ao conteúdo de novos documentos.

2.1.6 Métodos de avaliação de performance

Precisão

Esta é uma métrica que avalia a qualidade do que é produzido. Por outras palavras mede a taxa de verdadeiros positivos (TP) em relação à soma do número de verdadeiros positivos e de falsos poitivos (FP), ou seja:

$$P = \frac{TP}{(TP + FP)} \tag{2.7}$$

Recall

Também entendida como cobertura, mede a taxa de verdadeiros positivos em relação à soma do número de verdadeiros positivos e do número de falsos negativos, ou seja:

$$R = \frac{TP}{(TP + FN)} \tag{2.8}$$

F-score

Também conhecido por F1-score é uma medida para avaliar simultaneamente a precisão e *recall* de um modelo num conjunto de dados. Sendo calculado pela média harmónica entre a Precisão e *Recall*, tende a refletir qual dos valores das duas métricas é mais baixo:

$$F1 = \frac{2}{Recall^{-1} + Precision^{-1}} = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{2.9}$$

Onde TP são os *true positive*, FP são os *false positive* e FN os *false negative*.

Accuracy

Esta métrica é dada pela seguinte taxa, em que TN representa o número de verdadeiros negativos.

$$Acc = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{2.10}$$

2.1.7 Métodos de clustering

Sendo o foco desta dissertação no domínio da classificação não supervisionada, torna-se necessário obter *clusters* (grupos de documentos) a partir dos dados de entrada, isto é, os documentos caracterizados pelas suas *features*. Existem várias abordagens de *clustering*.

K-Means

Esta técnica tem como objetivo particionar n observações em K grupos, onde cada observação pertence ao grupo com o centróide mais próximo. Este método minimiza a variância dentro dos grupos, recorrendo normalmente à distância Euclidiana para a decisão sobre a qual dos grupos pertence cada obervação.

A imagem abaixo é o exemplo do resultado deste método de classificação.

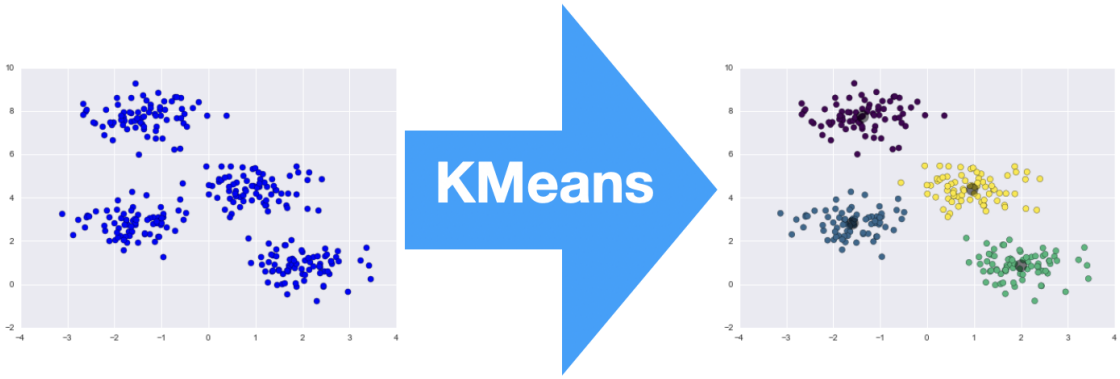

Figura 2.1: K-Means *Clustering*

Esta abordagem tem 2 desvantagens principais no contexto desta dissertação:

a) é necessário indicar o número de *clusters*, enquanto que na maior parte das situações o número de tópicos principais dos documentos (*clusters*) é desconhecido, sendo essa uma informação que queremos que seja automaticamente estimada pela abordagem.

b) utilizando distâncias como por exemplo a Euclideana, para o K-Means os *clusters* são hiper-esféricos, tendencionalmente de igual volume, o que não corresponde aos casos em que os *clusters* têm diferentes orientações, forma e volume, sendo uns compactos e outros dispersos.

No entanto, nalguns casos é uma solução com bons resultados e preferível dada a sua simplicidade.

Bisecting K-Means

Esta abordagem consiste numa modificação do algoritmo K-Means. Consegue realizar *clustering* particionado/hierárquico e reconhecer *clusters* de diferentes tamanhos e formas. Em vez de dividir os dados em K grupos, em cada iteração, o algoritmo divide um *cluster* em dois *sub-clusters* usando o K-Means simples até o número de *clusters* K ser alcançado.

O esquema abaixo mostra um exemplo das divisões realizadas pelo Bisecting K-Means.

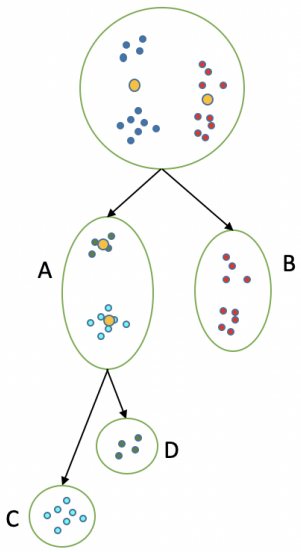

Figura 2.2: Bisecting K-Means *Clustering*

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise)

HDSCAN é um algoritmo criado por Campello, Moulavi e Sander [3] que estende o DBSCAN, convertendo-o num algoritmo de *clustering* hierárquico.

O DBSCAN agrupa os pontos que estão juntos (pontos com vários vizinhos perto), marcando também os pontos que estão em zonas com baixa densidade (cujos vizinhos estão longe). O critério principal na construção destes *clusters* está no facto de esses mesmos grupos serem criados por elementos com um número mínimo de vizinhos e uma distância máxima entre eles. Os pontos que não satisfazem esta condição são, em geral, considerados ruído.

Na imagem abaixo podemos ver a diferença do resultado obtido pelo K-Means e o HDSCAN, respetivamente.

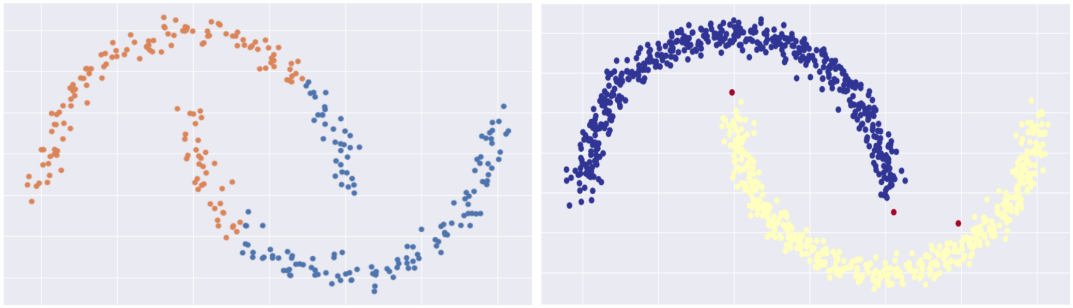

Figura 2.3: K-Means *vs* HDBSCAN *Clustering*

A imagem mostra que o algoritmo HDBSCAN é vocacionado para detetar *clusters* de forma irregular, muito diferentes dos hiper-esféricos ou hiper-elipsoidais. A sua utilização pode mostrar bons resultados para *clustering* de documentos dependendo do conjunto de *features* usado.

EM (Expectation Maximization)

Algoritmo que consiste num método iterativo para estimar parâmetros em modelos estatísticos, quando o modelo depende de variáveis latentes, ou seja, não observadas.

A iteração do EM alterna entre o passo de expectativa (E), e o passo de maximazação (M). A etapa de expectativa (E) cria uma função para a expectativa da verossimilhança logarítmica usando a estimativa atual para os parâmetros. A etapa de maximização (M), calcula os parâmetros que maximizam a verossimilhança logarítmica encontrada na etapa E. Este processo é repetido até se realizarem o número de iterações necessárias.

Na sequência de imagens seguinte temos os resultados de várias iterações resultantes do algoritmo.

A última imagem (canto inferior direito) corresponde ao *clustering* mais verossímil.

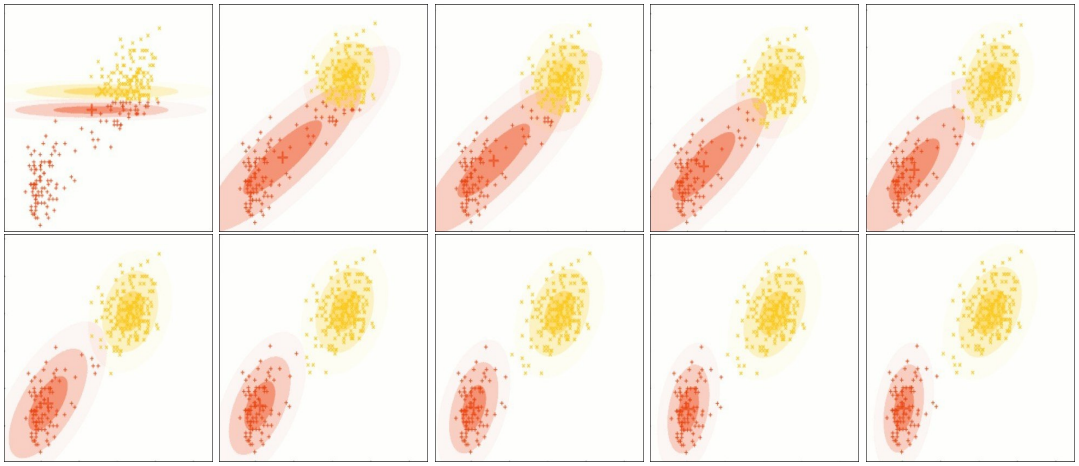

Figura 2.4: Sequência de iterações do EM

MeanShift

Tal como o K-Means, é uma técnica de *clustering* baseada num centroide. No entanto, ao contrário do K-Means onde é necessário indicar o número de *clusters*, o MeanShift deteta-os automaticamente através das densidades encontradas no conjunto de dados.

Começa-se por escolher um ponto aleatório e colocamos a 'janela' centrada no ponto. O tamanho da janela *bandwidth* é escolhido previamente. De seguida, calcula-se o ponto médio de entre todos os pontos que estejam dentro da 'janela' e move-se-a para o ponto médio. Este processo é repetido até convergir. A sequência abaixo é um exemplo deste processo.

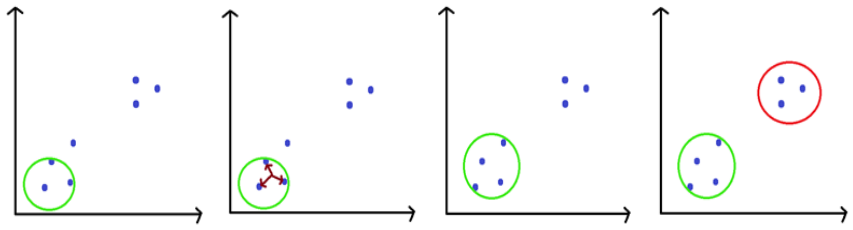

Figura 2.5: Sequência de iterações do MeanShift

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)

BIRCH é um algoritmo usado principalmente para realizar *clustering* de grandes conjuntos de dados. Inicialmente, é gerada uma síntese estatística para resumir o conjunto de dados maior, preservando o máximo de informação acerca do mesmo. Ou seja, em vez de se fazer o *clustering* do conjunto original de dados, é feito o *clustering* da síntese estatística que foi gerada. Esta é uma abordagem com potencialidade para agrupamento de documentos, tendo sido testada na abordagem desenvolvida nesta dissertação.

Spectral Clustering

O Spectral Clustering é um algoritmo que, através dos valores próprios de uma matriz de semelhança, aplica uma redução das dimensões para depois realizar o *clustering*.

A partir de um grafo feito tendo em conta os K vizinhos para cada ponto, o algoritmo constrói uma matriz Laplaciana. De seguida, são calculados os valores próprios dessa matriz e, tendo em conta esses valores, os pontos (dados) são representados em menores dimensões. Por fim, é realizado o *clustering*.

Desde que tenhamos uma *Affinity Matrix*, (matriz de semelhança), é possível usá-la no Spectral Clustering (saltando todos os passos até à contrução da matriz Laplaciana). Dado que nesta dissertação foi construída uma matriz de semelhança entre documentos, o Spectral Clustering surgiu como uma técnica necessariamente a considerar para agrupamento, tendo sido testada mostrando bons resultados.

A Figura 2.6 mostra um exemplo da comparação do *clustering* feito pelo K-Means e pelo Spectral Clustering, respetivamente.

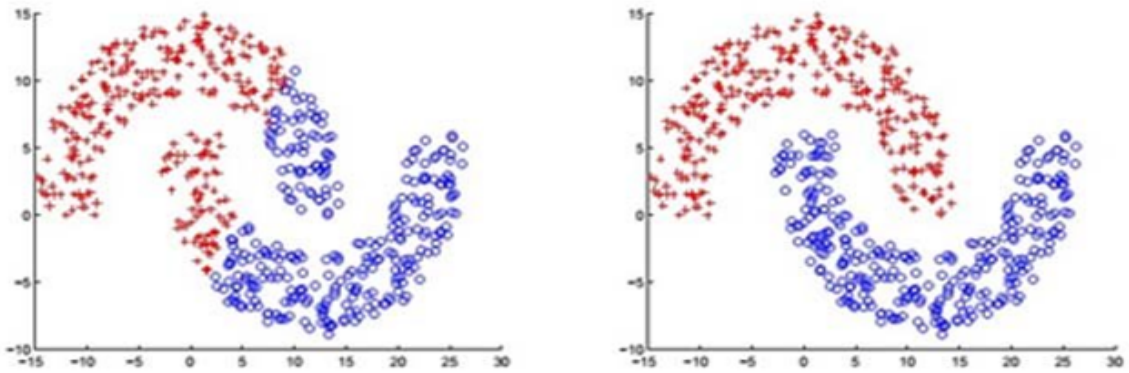

Figura 2.6: K-Means *vs* Spectral *Clustering*

O Spectral clustering não assume formas para os *clusters* e permite também fazer o *clustering* de dados que não sejam apenas representados por grafos.

2.1.8 Algoritmos classificadores

K-Nearest Neighbours

O K-NN é um classificador em que os objetos são classificados tendo em conta a classe dos K vizinhos mais próximos. É escolhido o parâmetro K (número de vizinhos) e a classe mais comum de entre esses vizinhos é atribuída ao objeto a classificar.

O K-NN é um classificador *lazy learning* e que, apesar de ter que comparar o objeto a classificar com os elementos da amostra disponível, as implementações existentes são bastante eficientes (ex: as disponíveis no python). É um classificador com bons resultados, naturalmente dependentes da qualidade das *features*.

Embora esta dissertação se foque na classificação não supervisionada, onde portanto não existem classes conhecidas à partida, o K-NN poderá ser útil numa fase posterior à determinação das classes e aí, quando estas forem conhecidas, poder-se-ão atribuir classes a amostras e obter a classificação de novos elementos.

Naive Bayes

É um classificador que faz parte da familia dos classificadores probabilísticos simples. Baseia-se em aplicar o teorema Bayes com fortes suposições de independência entre as *features*.

Apesar de ser um dos modelos mais simples, quando ocoplado com a estimativa de densidade *kernel*, consegue alcançar altos niveis de Precisão. Este tipo de classificadores é altamente escalável, precisando apenas de um número de parâmetros linear em relação ao número de *features* do problema. No contexto desta tese, os resultados a obter por este classificador deveriam ser comparados com os produzidos por outros classificadores tendo em conta que não é garantido que, para cada classe, as *features* sejam independentes.

SVM (Support Vector Machine)

Na sua versão original, por assim dizer, é um classificador binário não probabilístico. A partir de um conjunto de exemplos de treino, marcados com a classe a que pertencem, de um modo geral o SVM tenta encontrar uma linha de separação (hiperplano) entre os dados de duas classes. Essa linha procura maximizar a distância entre os pontos mais próximos em relação a cada uma das classes, tal como podemos ver abaixo:

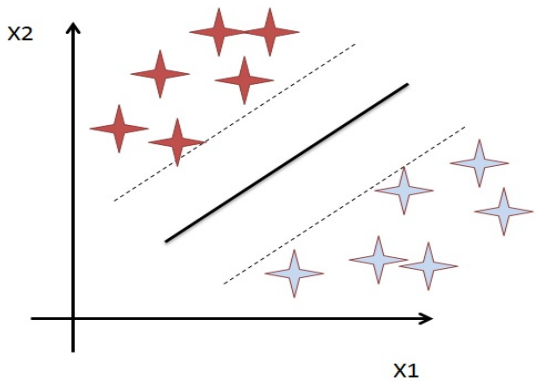

Figura 2.7: Hiperplano resultante do SVM

Com base no SVM também se pode classificar em contexto de várias classes. Tal como foi referido para o caso do K-NN, embora esta dissertação se foque na classificação não supervisionada, onde portanto não existem classes conhecidas à partida, também o SVM poderá ser útil numa fase posterior à determinação das classes. Acresce que, este classificador dispõe de vários *kernel* que potencíam a qualidade final da classificação.

2.2 Estado da arte

No domínio da classificação de documentos, existem várias abordagens, podendo separar-se em classificação supervisionada e classificação não supervisionada.

Em relação à classificação supervisionada, o seu uso permite obter, na maior parte dos casos, melhores resultados em termos de Precisão e *Recall*. Porém, este tipo de abordagem está limitado às classes conhecidas, sendo estas indicadas nas amostras de treino.

Por outro lado, na classificação não supervisionada, o principal objetivo é identificar classes não conhecidas previamente. Apesar de os resultados tenderem a ser inferiores em comparação ao outro tipo de abordagem, conseguimos classificar documentos, independentemente da língua e do tópico.

2.2.1 Classificação não supervisionada

Na abordagem [16], o autor Rohith Ramesh tem como base o processamento de linguagem natural. Inicialmente, são usadas expressões regulares para extrair o conteúdo do conjunto de dados. De seguida são removidas as pontuações, palavras irrelevantes, espaços adicionais, dígitos e lematiza-se o texto. Após este processo, é feita uma vectorização através do Tf-Idf e aplica-se o K-Means para encontrar os grupos.

O problema desta solução é termos de indicar o número de grupos a usar no K-Means, o que no nosso caso não seria uma abordagem correta tendo em conta que não temos nenhuma informação acerca dos mesmos. O processamento de linguagem natural também é uma limitação, pois está limitado à língua usada.

Em [6], uma abordagem que funciona em contexto de *Questioning-Answering*, os autores pretendem agrupar perguntas de estudantes, de modo a evitar respostas cujo conteúdo seja repetido. O modelo LDA é usado com o objetivo de formar grupos. Esta abordagem tem no entanto a desvantagem de ser necessário indicar o número de grupos.

Um outro procedimento existente, é o de Antoine Doucet e Miro Lehtonen [4]. O processo baseia-se em 4 fases e contém 2 tipos de *features* diferentes. *Text features* que são as palavras dos documentos após a remoção das *stop words* e, as *tag features* que usa o nome das *tags* do XML. Começa-se por representar os documentos em vetores de N dimensões, sendo N o número de *features*. A partir desses vetores e com o uso do K-Means, é feito um *clustering* usando apenas as *tag features* e outro usando apenas as *text features*.

Por último, os grupos formados em cada um dos métodos são combinados. Os *clusters* resultantes das *tag features* com similaridade interna acima de um certo valor são mantidos e, os restantes *clusters* são resultado das *text features*.

Os resultados obtidos encontram-se representados na tabela abaixo.

Features	Micro F1	Macro F1	Overall rank (out of 13)
Text	.348789	.290407	3rd
Tags	.132453	.081541	7th
Text+Tags	.253369	.203533	5th
Tags→Text, 0.8	.270350	.222365	4th

Figura 2.8: Resultados obtidos por Antoine Doucet e Miro Lehtonen

Na Figura 2.8, *Micro F1* corresponde à média ponderada (tendo em conta o tamanho (número de documentos) dos *clusters*), ou seja, maiores *clusters* irão ter um maior impacto no valor final. *Macro F1* é a média não ponderada de todas as pontuações *F1*.

A notação '*Tags→Text*, 0.8' significa que foi usada uma abordagem com *tag features* e depois alterada para *text features*, onde apenas foram mantidos *clusters* resultantes das *tag features* que obtiveram similaridade interna superior a 0.8. O *overall rank* é a classificação do método usado tendo em conta o *Micro F1* e *Macro F1* após se realizar o *clustering* 13 vezes; para mais detalhes, consultar [4].

Este tipo de procedimento não pode ser usado para o nosso objetivo porque, os documentos que pretendemos analisar não são documentos XML e mesmo que fossem, o uso de *tag features* não favorece a classificação dos mesmos, tal como podemos ver pelos resultados obtidos. E mais uma vez, o facto de termos de indicar ao K-Means o número de grupos é um obstáculo que apesar de poder se resolvido, não achamos que seja a melhor abordagem.

Segundo o trabalho de Youngjoong Ko e Jungyun Seo [8], uma possivel solução para a classificação de documentos, envolveria criar previamente conjuntos de treino para as diversas categorias, extrair as *features* e usar o Naive Bayes para classificar os documentos.

Há que referir que, independentemente dos resultados, esta abordagem está limitada às categorias dos conjuntos de treino que foram criados manualmente e, por sua vez, à língua das mesmas.

Em [20], Michael Snow explorou uma opção diferente. Após a remoção das palavras que são consideradas ruído, pontuações e dígitos, os documentos são vetorizados usando o Doc2Vec. Posto isto, com o auxílio do t-Distributed Stochastic Neighbor Embedding, é feita uma redução das dimensões de modo a prosseguirem, com o auxílio do HDSCAN, à classificação dos documentos. No final, o tópico dos documentos é obtido através do SVD (*singular value decomposition*). A limitação desta abordagem está no facto do Doc2Vec se focar nas palavras individuais para estabecer ligações semânticas, ignorando os termos multipalavra, termos que, podem ter uma semântica que não deriva da composição das palavras individuais, por exemplo: 'raining cats and dogs' não significa que chovem cães e gatos, mas sim que chove muito.

Uma abordagem que engloba o Tf-Idf é a que foi exlorada no Instituto Indiano de Tecnologia de Madras [17]. Aos documentos retirados de um dado *corpus*, é aplicado o Tf-Idf e de seguida, com o K-Means, classificam-nos. A escolha de *K* foi feita por tentativas. De entre os grupos obtidos pelo K-Means, são extraídas as palavras representativas de cada um deles, palavras essas que são escolhidas tendo em conta a proximidade ao respetivo centróide do grupo. Cada um dos documentos é associado ao *cluster* cujas palavras representativas têm maior frequência nesse documento. As palavras representativas ou parte delas irão servir para categorizar os documentos.

Deparamo-nos com dois problemas neste método. Um deles é, como já referimos anteriormente, a necessidade de indicar o número de grupos para o K-Means. O outro problema está na extração das palavras representativas de cada grupo. A extração é feita escolhendo as cinco palavras mais próximas do centróide. Uma limitação desta técnica reside no facto dos tópicos ficarem representados apenas por palavras individuais, ou seja, um tópico como por exemplo, 'aquecimento global', não será identificado por esta abordagem.

Uma solução diferente e com menos passos é proposta em [21] e consiste apenas em extrair as *features* dos documentos, fazer uma classificação inicial e depois ir otimizando essa mesma classificação com a ajuda da LCSH. A desvantagem está na LCSH, pois o seu uso está restrito ao Ingês.

O artigo de Diego Reforgiato Recupero [18] mostra-nos um procedimento com resultados interessantes. Inicialmente é feito um pré-processamento dos documentos. Cada documento é particionado em frases, de modo a extrair apenas as que têm substantivos e verbos e, é criada uma lista de *stop words* que irá diminuir a imprecisão do método de *clustering*. As palavras são substituídas pelos seus radicais com a ajuda do WordNet. Para as restantes palavras que não apareciam no WordNet, foi aplicado o algoritmo Porter para se fazer *stemming*. O último passo da fase de pré-processamento é remover as palavras que não ajudam a identificar os grupos, ou seja, as palavras que raramente aparecem nos documentos. De seguida, com o uso do WordNet, são criados dois tipos de *features*: as que foram obtidas através das categorias lexicais (WLC) e as que foram obtidas com a ontologia da WordNet (WO). Estas *features* são então usadas nos algoritmos Bisecting K-Means e Multipole.

A imagem abaixo mostra os resultados obtidos, sendo que os valores representam a medida *F1*.

Data set	$WLC_{Bisecting}$	$WO_{Bisecting}$	$WLC_{Multipole}$	$WO_{Multipole}$
$Reuters_{13}$	0.61	0.62	0.57	0.58
$Reuters_{25}$	0.60	0.59	0.55	0.55
$Classic_3$	0.59	0.60	0.56	0.60
$ManGen_{30}$	0.62	0.64	0.58	0.57

Figura 2.9: Resultados obtidos por Diego Recupero

Onde a primeira coluna indica o conjunto de dados que foi usado e os cabeçalhos das restantes colunas representam o tipo de *features* e algoritmo usados.

Apesar de resultados razoáveis, esta não é uma solução que possa ser usada em qualquer língua, tendo em conta que o algoritmo Porter está limitado ao Inglês, restrição essa que pretendemos evitar.

Num estudo realizado [11] para testar vários métodos de extração de *features* baseados nas métricas Tf-Idf, Tf-Df, TC e TS (já definidas acima) foi concluído que a métrica Tf-Idf produz genericamente melhores resultados. Na verdade, relativamente às métricas TC e TS, os termos considerados semanticamente vazios, tais como 'the', 'in' (em Inglês) ou 'o', 'com' (em Português), aparecem com forte tendência em todos os documentos da mesma língua, o que implica alto valor quer de TC quer de TS. No entanto, à partida, estes não são os termos aos quais se esperaria atribuir valores altos de TC e TS.

2.2.2 Classificação supervisionada

Existe uma maior variedade de estudos realizados na área da classificação supervisionada. Vários são os classificadores usados para este efeito, por exemplo SVM [7] entre outros; Naive Bayes e K-Nearest Neighbours [12] entre outros. São usadas também outras ferramentas neste contexto, por exemplo LDA [9], EM (Expectation Maximization) [14] [10], entre outras.

Embora o foco desta dissertação seja a classificação não supervisonada de documentos, após a formação dos *clusters*, será necessário recorrer a um algoritmo classificador. Este algoritmo poderá ser um dos conhecidos, tipicamente usados na classificação supervisionada.

3

Agrupamento e Classificação não Supervisionada de Documentos – Uma Contribuição

O agrupamento e a classificação não supervisionada constituem um grande desafio dado que desconhecemos à partida quais são as classes existentes, ou seja, não podemos treinar o sistema tal como é feito na classificação supervisionada. Este capítulo mostra como se podem encontrar as features discriminantes capazes de separar documentos por grupos/clusters segundo os seus tópicos. Mostra-se ainda como se podem classificar novos documentos tendo em conta os clusters obtidos na fase de aprendizagem, e como se podem extrair os tópicos principais dos grupos obtidos.

3.1 A selecção de features e a caraterização dos documentos

Nesta secção apresentam-se as fases que levam à extração e selecção de *features* caracterizadoras dos documentos.

3.1.1 A extração de expressões relevantes

Na aprendizagem não supervisionada, a selecção de *features* é uma tarefa especialmente importante, já que é através destas que se fará a discriminação dos objetos em grupos diferentes. Quando os objetos são documentos de texto não estrututado (texto cru), as *features* centram-se na 'relevância semântica' das sequências de palavras, relevância que contribui para discriminar textos relativamente ao tópico de que tratam.

Assim, há sequências de palavras tais como 'agricultura biológica', 'física nuclear', entre muitas outras, que reconhecemos serem semanticamente fortes, sendo indiciadoras do tópico a que provavelmente pertencem os documentos que as contêm. É pois necessário, antes de mais, extrair automaticamente estes termos relevantes multi-palavra a partir do conjunto completo de documentos.

De forma a extrair expressões relevantes multi-palavra, também designadas por Multiword Expressions (MWE), usámos o algoritmo LocalMaxs, [19], que se baseia nas coesões dos termos, para determinar se são ou não expressões relevantes.

Por outras palavras, seja W um n-grama, isto é, um termo constituído por um conjunto de n palavras contíguas. De acordo com o LocalMaxs:

O n-grama W é uma expressão relevante se para:

$\forall x \in \Omega_{n-1}(W), \forall y \in \Omega_{n+1}(W)$,

se $length(W) = 2 \implies g(W) > y$

se $length(W) > 2 \implies g(W) > (x+y)/2$

onde $g(.)$ representa em abstrato a coesão entre palavras contíguas, medida por uma métrica,

$\Omega_{n-1}(W)$ é o conjunto dos valores $g(.)$ de todos os termos de tamanho $n-1$ contidos em W, existentes no *corpus* e

$\Omega_{n+1}(W)$ é o conjunto dos valores $g(.)$ de todos os termos de tamanho $n+1$ que contêm W, no *corpus*.

Assim, um termo W de tamanho 2, é uma expressão relevante se a coesão do mesmo for superior a todas as coesões dos termos de tamanho 3, que contenham o primeiro termo. Por outro lado, um termo W de tamanho n superior a 2, só é expressão relevante, se a sua coesão for superior à média entre as coesões de qualquer combinação com um termo de tamanho $n-1$ contido em W, e outro de tamanho $n+1$ contendo W.

Nesta dissertação, os valores da coesão $g(.)$ dos termos são calculados, usando a métrica $SCP(.)$ por razões de eficiência e simplicidade. Esta métrica tem a seguinte estrutura básica, percetível quando aplicada a n-gramas de tamanho 2, ou seja, a uma sequência de palavras x e y:

$$SCP(x,y) = p(x|y) \times p(y|x) = \frac{p(x,y)^2}{p(x) \times p(y)} \quad . \tag{3.1}$$

Sendo $p(x,y)$, $p(x)$ e $p(y)$ os valores da frequência relativa, também vulgarmente denominada *probabilidade*, do bigrama x,y e dos unigramas x e y. Assim, $p(.) = f(.)/N$, sendo $f(.)$ a frequência absoluta do n-grama no *corpus* de tamanho N, isto é, composto por n palavras. Se um termo W tem tamanho superior a 2, composto pelas palavras $w_1 ... w_n$:

$$SCP(w_1 ... w_n) = \frac{p(W)^2}{F} \tag{3.2}$$

em que

$$F = \frac{1}{n-1} \sum_{i=1}^{n-1} (p(w_1 ... w_i) \times p(w_{i+1} ... w_n)) \tag{3.3}$$

Apenas para efeitos de extração de MWE dum conjunto de documentos, estes são tomados como se se tratasse apenas dum grande documento composto por esse conjunto. Após a execução do LocalMaxs sobre o conjunto de documentos, é vulgar obter-se várias dezenas de milhares de MWE. Sendo o objetivo destes termos a sua utilização como *features* catacterizadoras dos documentos, este número é obviamente excessivo. Veremos adiante como reduzi-lo.

Como já foi referido, o conjunto de MWE obtido pode compreender expressões tais como 'agricultura biológica', 'física nuclear', 'organização das Nações Unidas', etc.. É de realçar que o LocalaMaxs não tem uma precisão de 100%, podendo assim extrair falsas MWE tais como 'uma camisola amarela', 'tem a mania de', etc..

3.1.2 A eliminação das palavras de fraca semântica (*stop words*)

Considerando que uma parte das palavras isoladas (1-gramas / unigramas) existentes nos textos são também semanticamente fortes e por isso potencialmente discriminantes, tais como 'agricultura', 'pesca', 'economia', é necessário separá-las do grupo das chamadas *stop words* (palavras vazias de conteúdo), tais como 'e', 'a', 'de', 'um', 'uma', etc.. O procedimento mais comum às abordagens que precisam de identificar *stop words* consiste na simples eliminação das primeiras n palavras mais frequentes no *corpus* (conjunto de documentos), sendo n escolhido de forma mais ou menos arbitrária, por visualização dessa mesma lista. Nesta dissertação procurámos identificar um limiar, que sugerisse a existência duma separação natural entre o grupo das *stop words* e as restantes palavras de semântica mais ou menos relevante.

Encontrámos como possíveis soluções dois métodos, um baseado no número de palavras distintas junto a cada um dos unigramas, isto é ocorrendo à sua esquerda ou à sua direita, e outro baseado nas probabilidades dos unigramas nos documentos. Como ambos os métodos apresentavam resultados muito semelhantes, optámos pela primeira opção: considerar o número de palavras distintas vizinhas que cada unigrama contém.

Tendo em conta que as palavras de fraca semântica ocorrem com alta frequência nos textos, isso irá, por sua vez, traduzir-se num grande número de palavras distintas à esquerda e à direita. Ou seja, após o cálculo do número de palavras distintas para cada unigrama, é necessário um critério para escolher a partir de que número, o unigrama é considerado como tendo uma fraca semântica, sendo por isso uma *stop word*. Para tal, ordenámos os unigramas de acordo com o número de palavras distintas junto a cada um deles e usámos o método do *Elbow*. Este método consiste em traçar um gráfico de acordo com os valores ordenados segundo o número de vizinhos distintos de cada unigrama. Assim, nas abcissas estão representados os unigramas segundo o seu *rank* obtido pela referida ordenação. No eixo das ordenadas estão os valores dos vizinhos distintos. O objetivo do método é encontrar o *Elbow* da curva. Este «cotovelo» reflete o ponto (unigrama) onde se dá o maior salto/mudança no número de vizinhos, considerando pontos consecutivos. A imagem seguinte ilustra a seleção do *Elbow*.

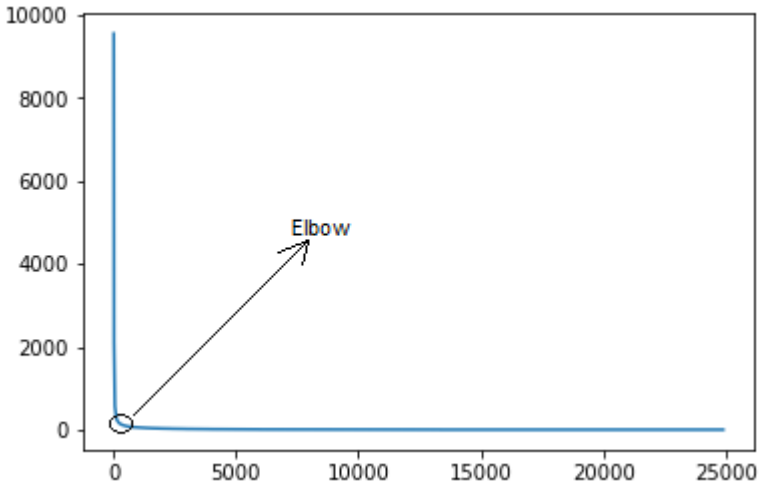

Figura 3.1: Seleção do Elbow

Este método foi originalmente proposto em [5], com objetivos diferentes dos tratados nesta dissertação. Nesse artigo, o ponto correspondente ao *Elbow* foi encontrado por observação da curva, de forma não automática. Porém, dado que nesta tese se pretendeu elaborar um mecanismo de treino e classificação automático, fez sentido tentar encontrar este «cotovelo» também automaticamente.

Assim, para encontrar o *Elbow* baseámo-nos no conceito de derivada e na sua variação ao longo dos vários pontos do gráfico. A derivada num ponto reflete a tangente, isto é, a inclinação, da função nesse ponto e, interessa-nos detetar quando é que se dá o maior salto (diferença) em valor absoluto entre duas derivadas de pontos próximos. Dado que não conhecemos a função associada à curva, não podemos calcular a derivada com precisão. No entanto, é possível estimá-la se aplicarmos a seguinte aproximação:

$$f'(x) \approx t(x) = \frac{\triangle y}{\triangle x} = \frac{f(x+\triangle x) - f(x)}{\triangle x} \tag{3.4}$$

onde $\triangle x$, em vez de ser um valor infinitesimal, é, por um lado, um valor suficientemente pequeno (diferença entre valores próximos em x) para refletir, de forma aproximada, o valor da inclinação real, mas por outro, suficientemente grande para ignorar as irregularidades locais do gráfico.

Deste modo, o *Elbow* será o ponto em x que maximiza o valor absoluto da diferença entre o valor de duas *derivadas* contíguas:

$$elbow = \underset{x}{\operatorname{argmax}}(|t(x+\triangle x) - t(x)|) \ . \tag{3.5}$$

Encontrado o «cotovelo», consideramos como palavras de fraca semântica, todos os unigramas à sua esquerda, ou seja, serão apenas unigramas com um grande número de palavras distintas. No entanto, como é possível que o nosso método não tenha detetado todos os termos de fraca semântica, decidimos ainda incluir termos com tamanho inferior ou igual a três, uma vez que independentemente da língua, são termos que podem ser excluídos, tais como 'the', 'has', 'in', 'os', etc.

Após a seleção e remoção das palavras de fraca semântica (*stop words*), foram também removidas as expressões relevantes que começassem ou acabassem numa *stop word*, visto que seriam expressões com relevância mas formadas por sintagmas incompletos, como por exemplo, 'guerra mundial na' ou 'e guerra mundial'.

3.1.3 Matriz de semelhança entre documentos

Tendo em conta que o número de termos relevantes extraídos dos documentos pode ser da ordem das dezenas de milhares, há que reduzir este número de features. Nesta dissertação, esta redução começa por construir uma matriz de semelhança entre documentos, fazendo com que cada documento seja avaliado quanto à sua semelhança com os seus pares, isto é, com todos os outros documentos.

Por seu turno, a semelhança entre documentos é calculada com base no coeficiente de Pearson e em métricas desenvolvidas nesta dissertação tendo em conta vários fatores que atribuem diferentes valores relativos a cada *feature*, como veremos em seguida. Esta semelhança entre um par de documentos (i, j) é definida por:

$$s(d_i, d_j) = \frac{cov(d_i, d_j)}{\sqrt{cov(d_i, d_i)} \times \sqrt{cov(d_j, d_j)}} \tag{3.6}$$

$$cov(d_i, d_j) = \frac{1}{\|T\|} \sum_{t \in T} \left(\left(g(t, d_i) - \overline{g(., d_i)} \right) \times \left(g(t, d_j) - \overline{g(., d_j)} \right) \right) \tag{3.7}$$

onde a média $\overline{g(., d)}$ é dada por

$$\overline{g(., d)} = \frac{1}{\|T\|} \sum_{t \in T} g(t, d) \tag{3.8}$$

Sendo que T representa o conjunto de termos constituído pelas expressões relevantes e pelos unigramas. O valor de $g(.,.)$ significa o quanto um termo se destaca em cada um dos documentos, e é composto por várias componentes, sendo $p(t, d_i)$, a frequência relativa do termo t em d_i (também designada probabilidade), a componente base que mais distingue cada documento.

$s(d_i, d_j)$ segue a estrutura do coeficiente de correlação de Pearson, onde a covariância $cov(d_i, d_j)$ é dividida pelo produto dos desvios padrão $\sqrt{cov(d_i, d_i)}$ e $\sqrt{cov(d_j, d_j)}$ apenas para obter valores de semelhança $s(d_i, d_j)$ limitados entre -1 e +1. Assim, $cov(d_i, d_j)$ reflete as contribuições dos diferentes termos t para a semelhança entre os documentos d_i e d_j. Mais pormenorizadamente, se a ocorrência de cada termo t se destacar relativamente à média das ocorrências dos outros termos no documento, quer em d_i quer em d_j, então quer $(g(t, d_i) - \overline{g(., d_i)})$ quer $(g(t, d_j) - \overline{g(., d_j)})$ serão valores positivos, originando pelo produto uma contribuição positiva. Se t não ocorrer em ambos d_i e d_j, também se perbece pela mesma fórmula (3.7), que a contribuição é positiva. Por outro lado, se t ocorrer apenas num dos documentos d_i ou d_j, a contribuição será negativa.

Note-se que, no âmbito do desenvolvimento duma abordagem que se pretende que venha a ser não superviosada, é necessário apesar disso, fazer monitorização dos resultados que se vão obtendo durante o desenvolvimento da abordagem. Para tal, é necessário conhecer a classe de cada documento, embora essa classe, como é sabido, não seja explicitamente etiquetada como tal. Sem esse conhecimento durante o desenvolvimento, não seria possível avaliar quão capaz é a abordagem para futuramente discriminar correctamente os documentos segundo as classes a que verdadeiramente pertencem.

Assim, como foi referido, $g(t,d_i)$ em (3.7) tem como principal factor o valor de $p(t,d_i)$. Porém, a matriz de semelhança obtida por (3.6), usando apenas esta componente em $g(t,d_i)$, mostrou uma qualidade insuficiente, já que, para os documentos que sabíamos serem da mesma classe, nem sempre as respetivas semelhanças eram altas entre si. Posto isto, foi necessário recorrer a várias métricas que foram sendo experimentadas e combinadas de modo a melhorar a qualidade da matriz, como a seguir se explica.

3.1.3.1 O coeficiente de variação

Inicialmente, optámos por incluir como componente de $g(.)$ o coeficiente de variação da probabilidade de cada termo no documento:

$$cv(t) = \frac{\sqrt{\frac{1}{\|Docs\|}\sum_{d\in Docs}(p(t,d) - \overline{p(t,.)})^2}}{p(t,.)} \tag{3.9}$$

Onde $Docs$ representa o conjunto de documentos, e $\overline{p(t,.)}$ é dado por

$$\overline{p(t,.)} = \frac{1}{\|Docs\|}\sum_{d\in Docs} p(t,d) \ . \tag{3.10}$$

Com $cv(t)$ pretende-se medir quão dispersa/variante é a probalilidade de cada termo t ao longo dos documentos. Em princípio, esperar-se-ia que esta variação fosse informativa. No entanto, a inclusão deste factor, traduzido por $g(t,d_i) = p(t,d_i) \times cv(t)$, não produziu melhorias na qualidade da matriz. A Figura 3.2 mostra a matriz obtida para um conjunto de documentos.

	14	15	16	17	18	19	20	21	22
0	0.281782	0.114288	0.073434	0.0389062	0.0465489	0.0561236	0.0171824	0.0189517	0.0058246
1	0.243248	0.100036	0.0988087	0.0319217	0.0573111	0.0519081	0.00627579	0.0198797	0.0017944
2	0.4432	0.315748	0.0776609	0.0411839	0.0573491	0.0565567	0.0164494	0.0368454	0.0061979
3	0.139916	0.0677892	0.0720535	0.0230919	0.0355661	0.0529575	0.00869672	0.0233583	0.0032113
4	0.0434341	0.0187795	0.00637062	0.0137949	0.0174239	0.032255	0.00354814	0.00220772	-0.0003375
5	0.120628	0.0905999	0.0532805	0.0150611	0.0203915	0.034082	0.0124356	0.0135311	0.0013559
6	0.494289	0.235044	0.202169	0.0543522	0.0886587	0.106152	0.0175052	0.0223599	0.0062006
7	0.172506	0.0645817	0.10378	0.0232206	0.039869	0.0431046	0.0138087	0.0210051	0.0095529
8	0.125619	0.0456007	0.117062	0.0225394	0.0256511	0.0203376	0.0234005	0.0258638	0.017373
9	0.100572	0.0412704	0.0658273	0.0589875	0.0192269	0.0188868	0.0113627	0.0125805	0.0066832
10	0.144829	0.0677364	0.901396	0.0332816	0.0591209	0.0396689	0.00522839	0.0100176	0.021188
11	0.380069	0.269736	0.0738453	0.0286588	0.0431867	0.0486629	0.00731405	0.0238545	0.0006896
12	0.103891	0.0467515	0.060966	0.016703	0.0225293	0.0292594	0.0176909	0.00409704	0.0032463

Figura 3.2: Matriz de semelhança com a inclusão do coeficiente de variação

Para melhor controlo da qualidade da matriz, os documentos que sabemos serem da mesma classe foram colocados como vizinhos para facilitar a visualização dos valores de semelhança entre documentos da mesma classe, que se pretende serem altos. No caso da Figura 3.2 os primeiros 20 documentos fazem parte da mesma classe e podemos constatar que apenas para alguns pares de documentos, a semelhança é superior a 0.20, como por exemplo no par (10,16), o que mostra que a semelhança entre os documentos ainda não é plenamente detetada.

3.1.3.2 O $Tf-Idf$

Como foi referido no capítulo 2, a métrica $Tf-Idf$ indica o quão importante um termo é num documento em relação ao conjunto de documentos. Assim, também este foi experimentado como factor a incluir em $g(t,d)$, e de facto, apresentou valores altos para alguns termos que sabemos serem importantes. No entanto, um termo que seja importante e que apareça em todos os documentos de uma determinada classe, por ser característico dela, tenderá a ter um valor baixo de $Tf-Idf$, o que não corresponde à importância que o termo deveria ter para efeitos de discriminação da classe. Isto deve-se ao fator idf desta métrica, que considera os termos não raros como tendencialmente insignificantes, como se pode perceber pelo fator da direita na Equação (2.1).

Por outro lado, se um termo surgir apenas num documento, este termo será considerado de grande importância segundo o $Tf-Idf$. Porém, um termo que surge apenas num documento não deve ser considerado um elemento representativo da classe, sobretudo se a classe tiver vários documentos. Por estas razões, compreende-se que a inclusão desta mátrica em $g(t,d_i)$, ou seja $g(t,d_i) = p(t,d_i) \times Tf-Idf(t,d_i)$, não tenha produzido melhorias na qualidade da matriz de semelhança.

3.1.3.3 O *Skewness*

A primeira métrica, para além da frequência relativa do termo t no documento d, isto é, $p(t,d)$, usada como componente de $g(t,d)$ em (3.7), e que apresentou melhorias na qualidade da matriz, foi a medida estatística conhecida como *Skewness*, definida por:

$$sk(t) = \frac{\frac{1}{\|Docs\|}\sum_{i=1}^{\|Docs\|}(p(t,d_i) - \overline{p(t,.)})^3}{\left(\sqrt{\frac{1}{\|Docs\|}\sum_{i=1}^{\|Docs\|}(p(t,d_i) - \overline{p(t,.)})^2}\right)^3} \quad . \tag{3.11}$$

O *Skewness*, permite-nos ver se um termo é equilibrado em torno da média das probabilidades desse mesmo termo ao longo dos documentos. Se houver equilibrio perfeito, o valor de *Skewness* é 0, como se pode compreender pelo numerador da Equação (3.11). Se apenas uma parte dos documentos apresentar uma probabilidade do termo t com valores significativamente maiores do que a média dessa probabilidade, então o valor de $sk(t)$ é positivo, refletindo um *desiquilíbrio para a direita*, que corresponde a um termo t que tende a ser característico duma classe e portanto discriminante dela.

Por outro lado, se apenas uma parte dos documentos apresentar valores de probabilidade de t significativamente menores do que a média, então o valor de $sk(t)$ será negativo. Porém, esta situação correponde apenas a uma hipótese teórica, resultante da fórmula em (3.11), já que, na prática corresponderia a um termo ausente em apenas alguns documentos (ou com uma probabilidade muito baixa em relação à média), o que é invulgar e não seria característico de qualquer classe. Por esta razão e apenas por precaução, eventuais valores negativos de $sk(t)$ foram considerados nulos.

O denominador em (3.11) representa o cubo do desvio padrão dessa probabilidade, como o objetivo de normalizar o valor do referido equilíbrio.

Apesar das melhorias conseguidas na qualidade da matriz em (3.6) pela inclusão de $sk(t)$ em $g(t,d)$, a qualidade desta ainda não era suficientemente boa para podermos prosseguir para próxima fase. Assim, outras componentes tiveram ainda que ser criadas.

3.1.3.4 A popularidade do termo

Em geral, não se espera que um termo t que ocorra em todos os documentos, digamos um termo muito *popular*, seja discriminante e característico duma classe. Por seu turno, um termo que apareça apenas num dos documentos de um grupo, é pouco provável que identifique uma classe, a não ser que essa classe apenas tenha um documento no grupo. Por outro lado, se um termo surgir em cerca de metade dos documentos dum grupo, não será de estranhar que o termo seja próprio duma classe ou dum subconjunto das classes representados no grupo de documentos. Assim a popularidade dum termo ao longo dos documentos, pode ser, à partida, um indicador da sua capacidade para discriminar classes.

Embora não se saiba quantas classes à partida existem num grupo de documentos, existirão pelo menos duas a discriminar. Assim, pareceu-nos que seria de valorizar os termos que surgissem com uma popularidade próxima e não superior a metade do número de documentos. Embora este critério pareça favorecer os casos em que existam apenas duas classes, na prática, como verificámos posteriormente, o critério mostrou bons resultados, já que os termos que surgem com uma popularidade próxima da metade, tendem de facto a ser discriminantes. Assim, foi definida a métrica $log-pop(t)$ por:

$$log-pop(t) = \begin{cases} \ln(n+1) & \text{se } n \leqslant \frac{\|Docs\|}{2} \\ \varepsilon & \text{se } n > \frac{\|Docs\|}{2} \end{cases} \tag{3.12}$$

Onde n representa a popularidade do termo t, ou seja, o número de documentos que contêm t. ε representa um valor relativamente baixo que foi atribuído a $log–pop(t)$ para os casos em que o termo surgia em mais do que metade dos documentos, não sugerindo por isso, à partida, capacidade discriminante. As experiências feitas mostraram que em geral um termo t tende a ter uma capacidade discriminante que cresce de forma suave, mas não linear, com n (o número de documentos onde t ocorre) até valores de n próximos da metade do número de documentos; daí a opção pela função ln(.). Foram experimentados diferentes valores para a base do logarítmo sendo que no fim a base natural mostrou melhores resultados. O argumento $(n+1)$ impede que um termo que ocorra apenas num documento não seja valorizado.

Esta métrica permite compensar valores altos da componente *Skewness* (3.11) para termos que não são representativos de uma classe. Por exemplo, um termo que só apareça num documento, isto é, um *outlier*, terá um *Skewness* alto. No entanto, de acordo com a componente $log-pop(t)$, será fracamente valorizado.

Nas experiências feitas, a inclusão da métrica $log–pop(.)$ em $g(t,d)$ na fórmula (3.7) melhorou os resultados na matriz, tendo em conta que os valores de semelhança entre documentos que sabíamos serem da mesma classe, tinham subido. No entanto, existiam vários pares de documentos da mesma classe que exibiam valores de semelhança ainda relativamente fracos. Isto levou-nos a criar a componente $srm(t)$ que a seguir se descreve.

3.1.3.5 O *salto relativo médio* da probabilidade do termo

Esta métrica é definida por:

$$srm(t) = \frac{1}{\|Docs\|-1} \sum_{i=1}^{\|Docs\|-1} \frac{p(t,d_i) - p(t,d_{i+1})}{p(t,d_i)} \tag{3.13}$$

Em (3.13), para cada termo t, os documentos estão ordenados por ordem decrescente da frequência relativa de t. Por outras palavras, estas frequências obedecem a:
$p(t,d_i) \geq p(t,d_{i+1}) \wedge \nexists\Big(d_j \neq d_i \wedge d_j \neq d_{i+1} : p(t,d_i) > p(t;d_j) \wedge p(t,d_j) > p(t,d_{i+1})\Big)$ Esta métrica pretende detetar saltos relativos significativos na probabilidade de t entre documentos vizinhos (segundo a ordenação das probabilidades do termo). Desta forma, um termo que seja específico de uma classe, tende a ter probabilidades maiores nos seus documentos, e probabilidades nulas ou muito baixas em todos os outros, provocando saltos na probabilidade entre *classes*.

Tomemos como exemplo um conjunto de 4 documentos, 2 de cada classe, e os termos t_1 com probabilidades (0.8, 0.78, 0.76, 0.75), e t_2 com probabilidades (0.5, 0.49, 0.05, 0.049). Aplicando (3.13), $srm(t_1) = 0.021$ e $srm(t_2) = 0.313$. Assim, srm favorece t_2 em relação a t_1 enquanto termos característicos de alguma classe, o que se percebe pelo exemplo.

Em (3.13), nos casos em que o valor de $p(t,d_i) = 0$, então, por razões de ordenação, $p(t,d_{i+1}) = 0$ e, para evitar a divisão por 0, sendo a diferença no numerador nula, este cálculo foi igualado a 0.

$srm(.)$ é uma métrica que serve também para dar menos peso a *stop words* que não tenham sido eliminadas pelo método automatizado já descrito na subsecção 3.1.2.

A inclusão em $g(t,d)$ da componente $srm(.)$, refletiu-se numa melhoria na qualidade da matriz. No entanto, foram ainda consideradas duas últimas componentes relacionadas com o comprimento dos termos e com o número de n-gramas distintos.

3.1.3.6 O comprimento médio do termo

Através de uma análise detalhada, facilmente se concluiu que os termos cujo comprimento médio (em número de caracteres) era relativamente grande, geralmente eram semanticamente mais fortes do que os outros. Um exemplo simples que mostra esta diferença consiste nos termos 'agricultura' e 'rio': o termo 'agricultura' é semanticamente mais preciso/informativo e portanto potencialmente mais representativo de uma classe, do que 'rio', que é mais vago, o que pode ser refletido através do comprimento médio das palavras do termo. O termo 'agricultura' tem um comprimento de 11, que é muito superior ao do termo 'rio', com apenas 3. O mesmo se aplica a termos com várias palavras. Assim, considerando o n–grama $W = w_1,\ldots w_n$, define-se $compM(w_1,\ldots w_n)$ como:

$$compM(w_1,\ldots w_n) = \frac{1}{n}\sum_{i=1}^{i=n} length(w_i) \tag{3.14}$$

onde $length(w_i)$ nos dá o número de caracteres da palavra w_i.

3.1.3.7 O número de n-gramas distintos

Os resultados parciais obtidos durante o desenvolvimento de toda esta abordagem, mostraram que as componentes a incluir em $g(t,d)$ e descritas até à subsecção anterior, tendem a atribuir maior relevância aos termos compostos apenas por uma palavra, isto é, os 1-gramas. No entanto, sabemos que outros termos composto por mais palavras têm também poder discriminante para caracterizar classes de documentos; ex: 'agricultura biológica', 'efeitos da pandemia', etc.. Deste modo procurou-se incluir uma componente para promover os termos compostos por mais do que 1 palavra. O critério usado consistiu em medir o número de n-gramas distintos em cada documento, tendo em conta o valor de n. Sabemos por experiência que o número de 2-gramas distintos num documento tende a ser superior ao número de 1-gramas distintos; em geral, o número de termos distintos de tamanho $n+1$ é superior ao número de termos de tamanho n no mesmo documento. Assim, foi, por último, incluída a componente $dist(t,d)$ em $g(t,d)$.

Em suma, a composição de $g(t,d)$ que apresentou melhores resultados na matriz de semelhança, ficou definida por:

$$g(t,d) = p(t,d) \times sk(t) \times log{-}pop(t) \times srm(t) \times compM(t) \times dist(t,d) \tag{3.15}$$

É de referir que foram feitas outras experiências atribuindo diferentes pesos a cada uma das componentes referidas, de maneira o obter altas semelhanças entre documentos da mesma classe sem que surgissem semelhanças com valores relativamente altos entre documentos de classes diferentes. No entanto, a fórmula em (3.15) mostrou ser o melhor compromisso, como se veio a constatar pelos resultados. A Figura 3.3 mostra os valores de semelhança obtidos para o mesmo conjunto de documentos usados na matriz 3.2, após a inclusão de todas as componentes referidas anteriormente.

	14	15	16	17	18	19	20	21	22
0	0.581244	0.30037	0.171854	0.114015	0.304151	0.336768	0.03405	0.0278066	0.033083
1	0.486038	0.288286	0.198586	0.0887142	0.264095	0.379682	0.00739439	0.0238684	0.010680
2	0.687701	0.573097	0.177898	0.118509	0.399802	0.37242	0.0196369	0.0171103	0.023570
3	0.59717	0.409846	0.274602	0.154231	0.446302	0.557404	0.0257727	0.0140278	0.027222
4	0.0938197	0.0890158	0.00573304	0.0389378	0.105199	0.110536	0.0222287	0.0181952	0.021865
5	0.540307	0.431407	0.218538	0.113261	0.326834	0.391724	0.0599713	0.0230426	0.035367
6	0.770402	0.528056	0.322042	0.155836	0.472468	0.569756	0.0380423	0.0198944	0.038919
7	0.243343	0.141093	0.121379	0.0401449	0.106834	0.150901	0.0157805	0.0276964	0.032170
8	0.407197	0.278724	0.270038	0.132018	0.395592	0.335652	0.0761316	0.116127	0.099886
9	0.231627	0.164163	0.12153	0.0976153	0.174218	0.124515	0.0296002	0.0389408	0.053007
10	0.303114	0.209161	0.965719	0.098357	0.258246	0.239815	0.00546808	0.0164006	0.101669
11	0.705643	0.549874	0.182599	0.0965969	0.283352	0.341589	0.0129045	0.0136415	0.013697
12	0.170418	0.10701	0.0798316	0.0536311	0.116675	0.138047	0.0526518	0.00448096	0.025583

Figura 3.3: Matriz de semelhança com a inclusão das componentes

Podemos ver que os valores de semelhança entre documentos da mesma classe subiram relativamente à matriz da Figura 3.2, sem que existam valores altos entre pares de classes distintas — até à coluna 19 os documentos fazem parte duma classe; a partir da coluna 20 são doutra classe.

3.1.4 Principal Component Analysis (PCA) e FastICA

Supondo que se pretende fazer agrupamento, por exemplo, de uma centena de documentos, então a matriz de semelhança obtida, por ser quadrada, terá 100 colunas, ou seja, 100 atributos. Obviamente que este é um número excessivo de $features$ que é preciso reduzir. Com o objectivo de conseguir uma redução deste número, foram conisideradas à partida as abordagens PCA e FastICA.

O PCA, que é um dos métodos mais usados e eficazes, permitiu uma redução significativa das *features*, sem que houvesse uma grande perda de informação associada aos dados originais, antes da redução. No nosso caso, conseguimos manter 95% da informação original com uma redução de mais de 20% das *features*.

O método alternativo aqui considerado para reduzir as *features*, FastICA, não produziu bons resultados. Isto porque o FastICA requer que se indique o número de componentes ($features$) que queremos manter, o que na prática é algo que não sabemos. Como experiência, testamos usar como número de componentes, o número resultante do PCA, de modo a podermos compará-los. No entanto, com as $features$ obtidas pelo FastICA, a próxima fase de agrupamento/*clustering*, não era feita corretamente, o que provavelmente se deve ao facto de ter havido uma grande perda de informação associada aos dados originais. Sendo assim, prosseguimos com a nossa solução, optando pelo uso do PCA.

3.2 Agrupamentos/*Clustering*

Como já foi referido, sendo esta uma abordagem não supervisionada, é necessário que o sistema seja capaz de detetar a composição de cada *cluster*.

3.2.1 O número de clusters

Obtida a matriz de documentos caracterizados pelas *features* que resultaram da redução PCA, seguiu-se a escolha da abordagem mais adequada para construir o agrupamento/*clustering* de documentos. Esta etapa constitui um desafio, pois não se conhecem as distribuições dos dados a analisar, pelo que foi necessário experimentar vários algoritmos que pudessem ser eficazes e eficientes neste processo. No entanto, uma vez que a generalidade dos algoritmos de *clustering* exige saber qual o número de clusters do conjunto a agrupar, foi necessário obter qual o número mais provável de grupos associados à matriz.

Para tal aplicou-se o método da *silhueta*. Este método consiste em calcular a média das *silhuetas* associadas a cada ponto, sendo que o valor da *silhueta* indica o quão semelhante é esse ponto em relação ao *cluster* a que supostamente pertence, em comparação aos outros *clusters*. O valor da *silhueta* varia entre -1 e 1, e é definida por:

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \quad . \tag{3.16}$$

Onde $a(i)$ corresponde à distância média do ponto i a todos os pontos do *cluster* a que supostamente pertence e $b(i)$ à distância média do ponto i a todos os pontos do *cluster* mais próximo.

Assim, o valor que resulta do maior score atribuído a cada um dos possíveis diferentes números de *clusters*, indica qual o número de grupos a usar como parâmetro de entrada nos algoritmos alternativos para *clustering*.

3.2.2 A avaliação da qualidade do agrupamento

Durante o desenvolvimento desta abordagem, a avaliação da qualidade do agrupamento foi feita de um modo semi-automático, sendo que foi necessário indicarmos o número de *clusters* que sabíamos existirem. De modo a simplificar este processo, os documentos que representam o mesmo tópico, encontram-se seguidos uns aos outros na matriz de semelhança, como já foi dito. Assumindo por exemplo que existem 60 documentos em 3 classes, 20 de cada, temos de ver a que *cluster* corresponde cada um dos *clusters* resultantes do agrupamento e se os documentos fazem realmente parte do *cluster* indicado. Isto é feito analisando a maioria dos documentos de cada *cluster*. Por exemplo, se o agrupamento resultante indicar que os 20 primeiros documentos são do *cluster* 3, sabemos que esse *cluster* corresponde ao nosso *cluster* 1, porque os documentos estão seguidos.

3.2.2.1 O Spectral Clustering e outras abordages

Foram testados alguns algoritmos de *clustering* de forma a adotar definitivamente aquele que melhor resultado apresentasse neste domínio do agrupamento de documentos.

Gaussian Mixtures Esta abordagem assume que nos dados existem n distribuições gaussianas e que cada uma delas corresponde a um $cluster$. Dado que numa distribuição gaussiana, os valores surgem mais densamente perto da sua média e cada vez com menos densidade à medida que se afastam dessa média, não é de estranhar que esta não seja a distribuição mais comum para os termos discriminantes das classes de documentos. Na verdade, o número de ocorrências, por exemplo, do termo 'agricultura' distribui-se pelos documentos dessa classe de forma não gaussiana, já que pode apresentar *outliers* e uma distribuição desiquilibrada. Assim, para se poder testar esta abordagem (Guassian Mixtures) foi necessário proceder à *gaussianização* dos dados [2]. Para tal, aplicámos as seguintes etapas:

Para cada coluna da matriz resultante do processo de PCA, a cada entrada é somado, se necessário, um valor igual de forma a não existirem entradas negativas. Em seguida, cada entrada y é modificada para y^λ onde

$y^\lambda = \frac{y^\lambda - 1}{\lambda}$ se $\lambda \neq 0$ ou

$y^\lambda = ln(y)$ se $\lambda = 0$

λ é escolhido de tal modo que $l(\lambda)$ é maximizado, sendo

$l(\lambda) = -\frac{m}{2} ln\left(\frac{1}{m}\sum_{j=1}^{m}(y_j^\lambda - \overline{y^\lambda})^2\right) + (\lambda - 1)\sum_{j=1}^{m} ln(y_j)$ e

$\overline{y^\lambda} = \frac{1}{m}\sum_{j=1}^{m} y_j^\lambda$

em que m corresponde ao número de elementos do *cluster*. Os valores de λ usados varreram o espaço entre 0 e 5, com saltos suficientemente pequenos. Desta forma, se considerássemos $\lambda = 0$, um conjunto de dados com os valores (4.5, 5, 5.5, 9) que não tem uma distribuição gaussiana, passaria a (1.50, 1.61, 1.70, 2.20). Ou seja, os valores seriam 'suavizados' fazendo com que se aproximassem mais de uma distribuição gaussiana. Mesmo assim, esta transformação não foi suficiente, já que, por exemplo, numa das experiências em que o número real de grupos de documentos era de 3, aplicada a Guassian Mixture a vários números de *clusters*, o clustering que resultou com melhor score segundo a *silhueta*, foi 4 clusters, o que diferiu do número real (3), na experiência feita. Assim, no sentido de usar a facilidade de leitura fornecida pelas matrizes de confusão, a Tabela 3.1 mostra os resultados obtidos pelo *clustering* resultante das Gaussian Mixtures nesta experiências com 60 documentos, 20 dos quais em cada uma de 3 classes, bem como os valores de Precisão e Recall.

		Classes Previstas			
		0	**1**	2	**3**
Classes Reais	**0**	20	0	0	0
	1	0	20	0	0
	2	0	0	10	10
	3	0	0	0	0

Tabela 3.1: Matriz de Confusão para as Gaussian Mixtures

$$
\begin{aligned}
Precisao_0 &= \frac{20}{20} = 1 \qquad & Recall_0 &= \frac{20}{20} = 1 \\
Precisao_1 &= \frac{20}{20} = 1 \qquad & Recall_1 &= \frac{20}{20} = 1 \\
Precisao_2 &= \frac{10}{10} = 1 \qquad & Recall_2 &= \frac{10}{10+10} = 0.5 \\
Precisao_3 &= 0 \qquad & Recall_3 &= N/A \\
Precisao_{global} &= \frac{1+1+1+0}{4} = 0.75 \qquad & Recall_{global} &= \frac{1+1+0.5}{3} = 0.83
\end{aligned}
\tag{3.17}
$$

Assim, para além do número errado de *clusters* propostos (4 em vez de 3), a Tabela 3.1 mostra que a número de documentos atribuídos ao *cluster* 2 foi de 10 em vez de 20; daí o Recall relativo a este *cluster* ser de 0.5. Esta insuficiência levou-nos a considerar outras abordagens.

MeanShif Foi considerada também a abordagem MeanShift, já explicada no capítulo 2. O melhor *clustering* indicado pelo MeanShift para a mesma experiência atrás referida de 60 documentos e 3 *clusters* reais, foi composto por mais de 30 grupos; um valor muito diferente do número real (3). Por motivos óbvios, este resultado dispensa a apresentação de qualquer matriz de confusão.

BIRCH Outro dos algoritmos testados, cujo modo de agrupamento é diferente dos restantes, como referido no capítulo 2, foi o BIRCH. Como se pode ver pela Tabela 3.2, para a mesma experiência de 60 documentos em 3 *clusters* reais, o número de grupos propostos por este algoritmo foi correto (3). No entanto, quer a Precisão quer o Recall foram de 0.95, valores que ainda tentámos melhorar. Um dos motivos que pode ter impedido o BIRCH de ter tido melhores resultados é o facto de ser um algoritmo vocacionado principalmente para grandes conjuntos de dados, o que no nosso caso, nem sempre acontece.

		Classes Previstas		
Classes Reais		**0**	**1**	2
	0	19	1	0
	1	0	19	1
	2	0	1	19

Tabela 3.2: Matriz de Confusão para o Birch

$$
\begin{aligned}
Precisao_0 &= \frac{19}{19} = 1 \qquad & Recall_0 &= \frac{19}{19+1} = 0.95 \\
Precisao_1 &= \frac{19}{19+1+1} = 0.9 \qquad & Recall_1 &= \frac{19}{19+1} = 0.95 \\
Precisao_2 &= \frac{19}{19+1} = 0.95 \qquad & Recall_2 &= \frac{19}{19+1} = 0.95 \\
Precisao_{global} &= \frac{1+0.9+0.95}{3} = 0.95 \qquad & Recall_{global} &= \frac{0.95+0.95+0.95}{3} = 0.95
\end{aligned}
\tag{3.18}
$$

Spectral Clustering Por último, testámos a abordagem *Spectral Clustering* que, por consultas feitas, se percebeu que podia ter uma boa performance para dados não gaussianos. Tendo em conta que da nossa solução resultam atributos que tendem a distribuir-se de forma não gaussiana, como já vimos, este parecia ser o algoritmo mais adequado para o *clustering* de documentos. Os resultados obtidos vieram a confirmar isso, pelo que decidimos usar como opção definitiva o Spectral Clustering. A título de exemplo, como se pode ver pela Tabela 3.3, para a mesma experiência de 3 *clusters* reais com 60 documentos, o número de grupos propostos coincidiu (3) e, quer a Precisão quer o Recall global foi de 0.98, valores superiores aos apresentados pelas outras abordagens.

Classes Reais \ Classes Previstas	**0**	**1**	**2**
0	19	0	1
1	0	20	0
2	0	0	20

Tabela 3.3: Matriz de Confusão para o Spectral Clustering

$$
\begin{aligned}
Precisao_0 &= \frac{19}{19} = 1 \qquad & Recall_0 &= \frac{19}{19+1} = 0.95 \\
Precisao_1 &= \frac{20}{20} = 1 \qquad & Recall_1 &= \frac{20}{20} = 1 \\
Precisao_2 &= \frac{20}{20+1} = 0.95 \qquad & Recall_2 &= \frac{20}{20} = 1 \\
Precisao_{global} &= \frac{1+1+0.95}{3} = 0.98 \qquad & Recall_{global} &= \frac{0.95+1+1}{3} = 0.98
\end{aligned}
\tag{3.19}
$$

3.3 A classificação de novos documentos

Obtidos os *clusters*, o que constitui a fase de treino, é possível classificar novos documentos através de classificadores já disponíveis. Nesta dissertação foram testados os classificadores Support Vector Machines (SVM) e o K Nearest Neighbors (K-NN).

Para realizar a classificação baseámo-nos na técnica 'leave one out' que consiste em realizar a fase de treino com todos os documentos do grupo, excepto um, à vez. Ou seja, para N documentos, irão ser feitas N iterações em que, para cada uma, é construída uma nova matriz de semelhança usando as equações (3.6) e (3.15), excluindo o documento que ficou de fora. De seguida, com o auxílio do PCA — já explicado na subsecção 3.1.4 — é feita a redução das *features* da nova matriz e aplica-se o Spectral Clustering para realizar o agrupamento. Por fim, uma vez obtidos os *clusters* e os documentos com a identificação do respetivo *cluster*, é feita a classificação do documento que ficou de fora, usando para tal um dos classificadores: o K-NN ou o SVM. No fim das N iterações é calculada a Precisão e o Recall quer para cada *cluster*, quer em termos globais. Apesar de ambos os classificadores serem muito usados em vários domínios, no caso da classificação de documentos e tendo em conta as *features* usadas para tal, os resultados finais evidenciaram uma superioridade do SVM em relação ao K-NN, com precisões acima dos 90%, o que levou à escolha deste classificador. No contexto da escolha do *kernel* a usar no SVM, foram testados os *linear*, *poly*, *rbf*, *sigmoid* e *precomputed*, tendo este último apresentado os melhores resultados.

3.4 A extração de tópicos dos documentos

Num contexto não supervisionado, para além do *clustering* e da classificação, seria conveniente saber quais os tópicos de cada grupo de documentos obtidos. Assim, nesta dissertação foi desenvolvida uma abordagem para extrair o conteúdo semântico *principal* de cada *cluster*. Para tal, houve que comparar possíveis medidas valorizadoras dos termos dentro dos *clusters*.

A primeira métrica desenvolvida neste contexto, consistiu numa fórmula que representa a posição média que um termo tem nos *clusters*, tendo em conta a sua *importância*. Assim, para calcular a essa posição média, baseámo-nos no valor do *rank* do termo t dentro de cada documento d, valor imposto por $g(t,d)$ — Equação (3.15) — que foi usado na matriz de semelhança. Pretende-se valorizar os termos que num *cluster* ocupam em média os primeiros *ranks* dentro dos seus documentos. Desta forma, quanto menor for o valor retornado por esta métrica, a que demos o nome de *ranking*, maior é a importância do termo:

$$R(t,c_i) = \frac{1}{\|c_i\|}\sum_{d_i \in c_i} rank(t,d_i) + \sum_{c_j \neq c_i}\frac{1}{\|c_j\|}\sum_{d_j \in c_j}(\|T\| - rank(t,d_j)) \ . \tag{3.20}$$

Onde $rank(t, d_i)$ representa a posição do termo t no documento d_i de acordo com a ordenação dos valores do $g(.,.)$ dos termos e, $\|T\|$ corresponde à cardinalidade do conjunto de termos constituído pelas expressões relevantes e pelos unigramas, tal como é usado por exemplo na Equação (3.7). Caso o termo não exista no documento, é como se estivesse na última posição, ou seja, o $rank(t, d_i)$ será igual ao número de termos. Deste modo, a fórmula permite-nos também valorizar os termos que aparecem exclusivamente num dos *clusters*, sendo por isso, provavelmente, um termo representante desse grupo de documentos. É de realçar que a ordenação das posições é feita separadamente no contexto de cada tamanho de n-grama ($n = 1 \ldots l$), em que l representa o limite de 7 palavras por termo.

Uma métrica alternativa consistiu numa variante da medida anterior, Equação (3.20). Baseia-se na utilização da mediana em vez da média aritmética. Pretendeu-se com esta alteração, valorizar alguns unigramas que, por terem mais do que um significado, podem surgir em mais do que um *cluster*, podendo por isso perder a *importância* que é dada a um termo quando surge em exclusividade num só *cluster*. A tendência da mediana para ignorar *outliers* poderia recuperar essa *importância*.

Outra alternativa que nos pareceu promissora, foi baseada na métrica $Tf-Idf$ que adaptámos para *clusters*. Com esta variante procurou-se fazer com que os termos que surgem maioritariamente num *cluster*, tenham maior valor nesse grupo, pelo que a fórmula adaptada ficou definida por:

$$cluster{-}tfidf(t, c_i) = \frac{1}{M(t, c_i)} \times log\left(\frac{\|clusters\|}{1 + \sum_{c_j \neq c_i} H(t, c_j)}\right) \tag{3.21}$$

$$M(t, c_i) = \frac{1}{\|c_i\|} \sum_{d_i \in c_i} rank(t, d_i) \tag{3.22}$$

$$H(t, c_j) = \frac{|d \in c_j : f(t, d) > 0|}{\|c_j\|} \tag{3.23}$$

À semelhança do $Tf-Idf$ conhecido, que mostra o quão exclusivo e frequente é um termo num documento, a variante aqui apresentada, $cluster{-}tfidf(t, c_i)$, espelha o quão exclusivo e frequente é um termo num *cluster*.

Pelo resultado dos testes feitos às alternativas aqui introduzidas, apesar de as três métricas apresentarem bons resultados, quando comparadas ao pormenor, a primeira, $R(t, c)$, Equação (3.20), mostrou ter em geral uma qualidade ligeiramente superior, sendo por isso essa a nossa escolha final para a extração dos tópicos de cada *cluster*.

No protótipo que resultou da presente abordagem, o número de termos a extrair como representantes principais do conteúdo do *cluster*, por tamanho de n-grama, pode ser escolhido pelo utilizador.

4

Resultados

Neste capítulo iremos mostrar os resultados para várias etapas de desenvolvimento da nossa solução. O conjunto de documentos (*corpus*) que foi usado para apresentar os resultados é constituido por 60 documentos, 20 de cada classe, sendo as classes 'medicina', 'finanças' e 'educação'. Para efeitos de comparação, iremos também apresentar os resultados do clustering e da classificação para dois conjuntos de documentos alternativos.

4.1 Palavras de fraca semântica (*stop words*)

Com a aplicação do nosso método, descrito em (3.1.2), conseguimos remover grande parte das palavras de fraca semântica. Os termos seguintes são exemplos que foram considerados como *stop words*: { and, the, in, to, for, a, by, that, or, is, as, was, it, has, its, on, if }. Para o *corpus* referido, foram detetados automaticamente 1699 *stop words*, o que correspondeu a uma Precisão de 100% e um Recall de 99%, tendo em conta a avaliação manualmente feita.

É de realçar que sendo a avaliação manual, os valores de Precisão e Recall podem variar ligeiramente, uma vez que existem palavras discutiveis, podendo ou não ser consideradas como *stop words*, dependendo do avaliador. Assim, essas palavras irão depender do critério de quem está a analisar.

4.2 Expressões relevantes

O mecanismo de seleção de expressões relevantes mostrou ter bons resultados. A título de exemplo, conseguimos obter expressões tais como: { 'higher education', 'secondary education', 'health care', 'emergency medicine', 'colleges and universities', 'financial crisis', 'medical school', 'alternative medicine', 'basic education', 'financial institutions', 'financial crises', 'public schools', 'equality in education', 'education system', 'global financial', 'stock price', 'doctor of medicine', 'traditional chinese medicine', 'individuals with disabilities', 'emergency medical systems', 'buyers and sellers' }.

No entanto, o extrator de expressões relevantes (LocalMaxs) não era perfeito, pelo que também extraía alguns termos que não eram importantes, tais como: { 'developing countries', 'more likely', '19th century' }. Apesar disso, essa imperfeição não tem uma influência capital na qualidade da matriz de semelhança, já que, por exemplo, mesmo a expressão 'developing countries', embora não sendo considerada Expressão Relevante, ela não é completamente vazia de conteúdo. Na verdade, 'developing countries' está vagamente associada à classe 'educação'.

4.3 Matriz de semelhança entre documentos

As figuras seguintes mostram partes da matriz obtida na etapa da construção da matriz de semelhança entre documentos.

	14	15	16	17	18	19	20	21	22
0	0.581244	0.30037	0.171854	0.114015	0.304151	0.336768	0.03405	0.0278066	0.033083
1	0.486038	0.288286	0.198586	0.0887142	0.264095	0.379682	0.00739439	0.0238684	0.010680
2	0.687701	0.573097	0.177898	0.118509	0.399802	0.37242	0.0196369	0.0171103	0.023570
3	0.59717	0.409846	0.274602	0.154231	0.446302	0.557404	0.0257727	0.0140278	0.027222
4	0.0938197	0.0890158	0.00573304	0.0389378	0.105199	0.110536	0.0222287	0.0181952	0.021865
5	0.540307	0.431407	0.218538	0.113261	0.326834	0.391724	0.0599713	0.0230426	0.035367
6	0.770402	0.528056	0.322042	0.155836	0.472468	0.569756	0.0380423	0.0198944	0.038919
7	0.243343	0.141093	0.121379	0.0401449	0.106834	0.150901	0.0157805	0.0276964	0.032170
8	0.407197	0.278724	0.270038	0.132018	0.395592	0.335652	0.0761316	0.116127	0.099886
9	0.231627	0.164163	0.12153	0.0976153	0.174218	0.124515	0.0296002	0.0389408	0.053007
10	0.303114	0.209161	0.965719	0.098357	0.258246	0.239815	0.00546808	0.0164006	0.101669
11	0.705643	0.549874	0.182599	0.0965969	0.283352	0.341589	0.0129045	0.0136415	0.013697
12	0.170418	0.10701	0.0798316	0.0536311	0.116675	0.138047	0.0526518	0.00448096	0.025583

Figura 4.1: Matriz de semelhança parte 1

	47	48	49	50	51	52	53	54	55
34	0.133691	0.0738785	0.0180666	0.0134943	0.0475559	0.0309772	0.0739811	0.0239695	0.029784
35	0.0261625	0.0141287	0.0112971	0.0113224	0.0320033	0.0350322	0.00303779	0.0149845	0.003377
36	0.0103547	0.00747772	0.00657188	0.00288024	0.0113472	0.0172516	0.0160014	0.00386446	0.022395
37	0.023244	0.0309752	0.0590455	0.00808223	0.0577779	0.00534529	0.00807451	0.0164295	0.010585
38	0.0423158	0.0708866	0.0589524	0.0134237	0.0406736	0.0343917	0.0331291	0.0357399	0.076555
39	0.0235121	0.0046979	0.0190903	0.0149508	0.033516	0.0298615	0.00399227	0.0205028	0.023873
40	0.337751	0.335325	0.441113	0.0950534	0.214959	0.423903	0.310778	0.271446	0.212778
41	0.279497	0.259221	0.292191	0.144741	0.116632	0.145504	0.205153	0.146637	0.149133
42	0.326227	0.157947	0.193628	0.0597879	0.18897	0.18069	0.706663	0.193223	0.122955
43	0.401003	0.55985	0.399532	0.14404	0.241793	0.293235	0.309588	0.264114	0.272436
44	0.48919	0.649184	0.359815	0.230556	0.360156	0.333703	0.353845	0.335224	0.313415
45	0.171513	0.562484	0.136602	0.0696098	0.148394	0.243419	0.190001	0.207951	0.144707
46	0.13587	0.172948	0.203376	0.115373	0.366127	0.0694964	0.10931	0.0715995	0.102004

Figura 4.2: Matriz de semelhança parte 2

	31	32	33	34	35	36	37	38	39
6	0.0186248	0.0312574	0.0153514	0.0404016	0.00494347	0.00706438	0.0205733	0.0399584	0.0065073
7	0.00568517	0.00533405	0.0151266	0.0318954	0.0128784	0.00590318	0.00835404	0.019189	0.0088386
8	0.0625172	0.0204162	0.0351699	0.11099	0.025107	0.0148855	0.0803015	0.100857	0.033193
9	0.0220193	0.0216071	0.0217668	0.13713	0.0172626	0.0154279	0.0650899	0.0507969	0.015146
10	0.00391494	0.0100772	0.0166106	0.0388385	0.00213748	0.0031665	0.00790666	0.0955642	0.0009054
11	0.0145256	0.00631904	0.00379667	0.034372	0.00790254	0.00159094	0.00964463	0.00951412	0.011758
12	0.00826529	0.00917208	0.00210324	0.0218748	0.0016185	0.00127932	0.004606	0.0166292	0.0049924
13	0.0180499	0.0503041	0.0223977	0.0142995	0.0147633	0.00503713	0.0238408	0.0191282	0.016968
14	0.0377878	0.124442	0.140794	0.072457	0.0171711	0.00522749	0.0317473	0.0485292	0.017864
15	0.0311401	0.00719721	0.00689238	0.0251184	0.00943581	0.00109058	0.0332048	0.0224104	0.044071
16	0.00899248	0.0192101	0.0165883	0.0231156	0.00198208	0.00119087	0.0109274	0.12985	0.0008953
17	0.0159588	0.00706409	0.0187919	0.0161899	0.0138683	0.00101604	0.0214931	0.0189756	0.011809
18	0.021145	0.00737503	0.0175991	0.0394118	0.0315537	0.00812517	0.0053129	0.0304262	0.037752

Figura 4.3: Matriz de semelhança parte 3

Através da Figura 4.1 podemos ver que os valores de semelhança entre documentos da mesma classe (documentos 0 a 19) são relativamente altos. Alguns pares de documentos apresentam uma semelhança menor, como por exemplo os pares que contêm o documento 4 ou o documento 17. Isto pode dever-se ao facto de, apesar de serem à partida da mesma classe, não apresentarem termos semanticamente fortes em comum com os restantes documentos. A Figura 4.2 mostra mais uma vez a semelhança entre documentos da mesma classe, neste caso para a segunda classe, (documentos 40 a 59). Por fim, na última Figura, 4.3, conseguimos ver que, para todos os pares entre documentos de classes diferentes, os valores de semelhança são muito baixos, o que corresponde à realidade.

4.4 Agrupamento/*Clustering*

As seguintes matrizes de confusão e valores de Precisão e Recall mostram os resultados obtidos para o *clustering* realizado a diferentes conjuntos de documentos, usando o Spectral Clustering, que como atrás explicado, foi a abordagem de agrupamento escolhida.

corpus A: 60 documentos - 3 classes (Medicina, Finanças, Educação) - 20 de cada classe

		Classes Previstas		
		0	**1**	**2**
Classes Reais	**0**	19	0	1
	1	0	20	0
	2	0	0	20

Tabela 4.1: Matriz de Confusão para o *clustering* — *corpus* A

$$
\begin{aligned}
Precisao_0 &= \frac{19}{19} = 1 & Recall_0 &= \frac{19}{19+1} = 0.95 \\
Precisao_1 &= \frac{20}{20} = 1 & Recall_1 &= \frac{20}{20} = 1 \\
Precisao_2 &= \frac{20}{20+1} = 0.95 & Recall_2 &= \frac{20}{20} = 1 \\
Precisao_{global} &= \frac{1+1+0.95}{3} = 0.98 & Recall_{global} &= \frac{0.95+1+1}{3} = 0.98
\end{aligned}
\tag{4.1}
$$

corpus B: 75 documentos - 5 classes (Poluição, Música, Plantas, Literatura, Nutrição) - 15 de cada classe

		Classes Previstas				
		0	**1**	**2**	**3**	**4**
Classes Reais	**0**	15	0	0	0	0
	1	0	15	0	0	0
	2	0	0	15	0	0
	3	0	0	0	15	0
	4	0	0	0	0	15

Tabela 4.2: Matriz de Confusão para o *clustering* — *corpus* B

$$
\begin{aligned}
Precisao_0 &= \frac{15}{15} = 1 \qquad Recall_0 = \frac{15}{15} = 1 \\
Precisao_1 &= \frac{15}{15} = 1 \qquad Recall_1 = \frac{15}{15} = 1 \\
Precisao_2 &= \frac{15}{15} = 1 \qquad Recall_2 = \frac{15}{15} = 1 \\
Precisao_3 &= \frac{15}{15} = 1 \qquad Recall_3 = \frac{15}{15} = 1 \\
Precisao_4 &= \frac{15}{15} = 1 \qquad Recall_4 = \frac{15}{15} = 1 \\
Precisao_{global} &= \frac{1+1+1+1+1}{5} = 1 \qquad Recall_{global} = \frac{1+1+1+1+1}{5} = 1
\end{aligned}
\tag{4.2}
$$

corpus C: 60 documentos - 3 classes (Agricultura, Baskteball, Religião) - 20 de cada classe

		Classes Previstas		
Classes Reais		**0**	**1**	**2**
	0	20	0	0
	1	0	20	0
	2	0	0	20

Tabela 4.3: Matriz de Confusão para o *clustering* — *corpus* C

$$
\begin{aligned}
Precisao_0 &= \frac{20}{20} = 1 \qquad Recall_0 = \frac{20}{20} = 1 \\
Precisao_1 &= \frac{20}{20} = 1 \qquad Recall_1 = \frac{20}{20} = 1 \\
Precisao_2 &= \frac{20}{20} = 1 \qquad Recall_2 = \frac{20}{20} = 1 \\
Precisao_{global} &= \frac{1+1+1}{3} = 1 \qquad Recall_{global} = \frac{1+1+1}{3} = 1
\end{aligned}
\tag{4.3}
$$

Através das várias matrizes de confusão podemos ver que, em geral, o Spectral Clustering tem resultados com precisões e Recall muito próximos de 100%. Tendo em conta que os resultados do *clustering* dependem da matriz de semelhança, para eventuais casos em que os documentos da mesma classe não apresentem termos semanticamente fortes e comuns entre si, este agrupamento poderá ser afetado.

4.5 Classificação de documentos

Tal como para o *clustering*, os resultados da classificação de documentos foram também realizados para 3 conjuntos diferentes, mais concretamente, para os mesmos *corpora* A, B e C, apresentados na fase de *clustering*, Secção 4.4, o que foi possível por recurso à técnica do *leave-one-out*. Os resultados foram obtidos usando o classificador SVM e estão representados nas matrizes de confusão abaixo.

corpus A: 60 documentos - 3 classes (Medicina, Finanças, Educação) - 20 de cada classe

Classes Reais \ Classes Previstas	**0**	**1**	**2**
0	18	0	2
1	0	18	2
2	0	0	20

Tabela 4.4: Matriz de Confusão para a classificação — *corpus* A

$$
\begin{aligned}
Precisao_0 &= \frac{18}{18} = 1 \qquad Recall_0 = \frac{18}{18+2} = 0.9 \\
Precisao_1 &= \frac{18}{18} = 1 \qquad Recall_1 = \frac{18}{18+2} = 0.9 \\
Precisao_2 &= \frac{20}{20+2+2} = 0.83 \qquad Recall_2 = \frac{20}{20} = 1 \\
Precisao_{global} &= \frac{1+1+0.83}{3} = 0.94 \qquad Recall_{global} = \frac{0.9+0.9+1}{3} = 0.93
\end{aligned}
\tag{4.4}
$$

corpus B: 75 documentos - 5 classes (Poluição, Música, Plantas, Literatura, Nutrição) - 15 de cada classe

Classes Reais \ Classes Previstas	**0**	**1**	**2**	**3**	**4**
0	15	0	0	0	0
1	0	15	0	0	0
2	0	0	15	0	0
3	0	0	0	15	0
4	0	0	0	0	15

Tabela 4.5: Matriz de Confusão para a classificação — *corpus* B

$$
\begin{aligned}
Precisao_0 &= \frac{15}{15} = 1 \qquad Recall_0 = \frac{15}{15} = 1 \\
Precisao_1 &= \frac{15}{15} = 1 \qquad Recall_1 = \frac{15}{15} = 1 \\
Precisao_2 &= \frac{15}{15} = 1 \qquad Recall_2 = \frac{15}{15} = 1 \\
Precisao_3 &= \frac{15}{15} = 1 \qquad Recall_3 = \frac{15}{15} = 1 \\
Precisao_4 &= \frac{15}{15} = 1 \qquad Recall_4 = \frac{15}{15} = 1 \\
Precisao_{global} &= \frac{1+1+1+1+1}{5} = 1 \qquad Recall_{global} = \frac{1+1+1+1+1}{5} = 1
\end{aligned}
\tag{4.5}
$$

corpus C: 60 documentos - 3 classes (Agricultura, Baskteball, Religião) - 20 de cada classe

Classes Reais \ Classes Previstas	**0**	**1**	**2**
0	20	0	0
1	0	20	0
2	0	0	20

Tabela 4.6: Matriz de Confusão para a classificação — *corpus* C

$$
\begin{aligned}
Precisao_0 &= \frac{20}{20} = 1 \qquad Recall_0 = \frac{20}{20} = 1 \\
Precisao_1 &= \frac{20}{20} = 1 \qquad Recall_1 = \frac{20}{20} = 1 \\
Precisao_2 &= \frac{20}{20} = 1 \qquad Recall_2 = \frac{20}{20} = 1 \\
Precisao_{global} &= \frac{1+1+1}{3} = 1 \qquad Recall_{global} = \frac{1+1+1}{3} = 1
\end{aligned}
\tag{4.6}
$$

A Precisão e o Recall obtidos para os diferentes conjuntos de documentos foram altos. Conseguimos ver que existe uma relação clara entre os resultados obtidos pelo *clustering* e a classificação do SVM, sendo que quanto melhor for o agrupamento obtido, melhor será a classificação realizada. De facto, nos casos dos *corpora* B e C, onde a Precisão e Recall associados ao processo de *clustering* foram de 1 (100%), não houve erros de classificação. No entanto, para o *corpus* A, por o *clustering* não ter sido perfeito, também a classificação apresenta alguns erros, ainda que relativamente baixos no contexto da área da classificação não supervisionada de documentos: por exemplo, $Precisao_2 = 0.83$ e $Recall_1 = 0.9$.

4.6 Extração de tópicos dos documentos

Como foi referido, o conteúdo principal de cada *cluster* c, é obtido por ordenação da métrica $R(t,c)$, Equação (3.20), criada para este efeito.

Como se sabe, o número de n-gramas a extrair por comprimento do n-grama (n), pode ser escolhido pelo utilizador. À partida, não existe um número óptimo de n-gramas como tópicos/subtópicos, uma vez que se forem poucos, podem formar um conjunto *incompleto*, isto é, insuficientemente informativo relativamente ao conteúdo principal do *cluster*. Por outro lado, se forem muitos, o conteúdo do *cluster* não será apresentado de forma compacta. Tendo em conta as experiências feitas, um critério que se mostrou eficaz para mostrar o conteúdo de cada grupo pode ser descrito da seguinte maneira: o utilizador pode começar por pedir para cada comprimento de n-grama um número relativamente elevado de tópicos/subtópicos, por exemplo 4 ou 5. Dado que o conjunto obtido pode conter elementos sementicamente equivalentes (por exemplo 'patient' e 'patients') ou outros pouco relevantes, poder-se-á repetir o pedido indicando um número mais reduzido de n-gramas e verificar se o novo conjunto não perdeu conteúdo relevante.

As tabelas 4.6, 4.6 e 4.6, mostram o conteúdo extraído por esta abordagem, para o exemplo do *corpus* A, cujas classes são, como foi referido, Medicina, Finanças e Educação.

patients, physician, treatments, patient, diseases
'health care', 'human body', 'medical school', 'infectious diseases', 'primary care'
'practice of medicine', 'diagnosis and treatment', 'american medical association', 'physicians and surgeons', 'college of physicians'
'played an important role', 'hippocratic oath for physicians', 'accreditation council for graduate'
'college of physicians and surgeons', 'department of health and human', 'erasistratus connected the increased complexity'
'journal of the american medical association', 'department of health and human services', 'asclepeia provided carefully controlled spaces conducive'

Tabela 4.7: Tópicos/subtópicos — classe Medicina

investors, sell, buying, banks, stock
'interest rates', 'financial institutions', 'financial crisis', 'financial markets', 'real estate'
'internation monetary fund', 'rate of return', 'bank of england', 'credit default swaps', 'amount of money'
'1998 russian financial crisis', 'financial crisis of 2007-2009', 'lender of last resort'
'banks and other financial institutions', 'reserve bank of new york', 'advanced and sophisticated ever seen'
'federal reserve bank of new york', 'about the early 1600s to about', 'general agreement on tariffs and trade'

Tabela 4.8: Tópicos/subtópicos — classe Finanças

elementary, schooling, parents, teacher, curricula
'secondary education', 'primary school', 'public schools', 'high school', 'education system'
'ministry of education', 'public and private', 'college or university', 'institutions of higher', 'quality of education'
'children between the ages', 'vocational education and training', 'center for education statistics'
'programme for international student assessment', 'historically black colleges and universities', 'secondary education act of 1965'
'information on reusing text from wikipedia', 'organisation for economic co-operation and development', 'incorporates text from a free content'

Tabela 4.9: Tópicos/subtópicos — classe Educação

Através das tabelas podemos ver que, independentemente da classe, grande parte dos n-gramas extraídos são semanticamente fortes e representativos da classe correspondente. No entanto, à medida que n aumenta, isto é, o número de palavras por n-grama, existe uma maior probabilidade de encontrarmos termos fracamente relevantes ou mesmo irrelevantes, como por exemplo o termo 'information on reusing text from wikipedia' da classe Educação. Tal não aconteceria se o extrator de expressões relevantes (LocalMaxs) fosse 100% preciso.

Assim, a Precisão global associada à qualidade dos n-gramas enquanto tópicos/subtópicos, extraídos por esta abordagem, foi de cerca de 84%, como se pode comprovar pelo conteúdo das tabelas 4.6, 4.6 e 4.6. O critério usado nesta avaliação pode ser explicado resumidamente da seguinte maneira: tomando o exemplo da Tabela 4.6, nos 2-gramas temos 'interest rates', 'financial institutions', 'financial crisis', 'financial markets' e 'real estate', sendo todos relevantes, ou seja, 100% de Precisão neste grupo de n-gramas, como facilmente se reconhece; no entanto, o conjunto dos 5-gramas é constituído por 'banks and other financial institutions', 'reserve bank of new york' e 'advanced and sophisticated ever seen', sendo este último claramente irrelevante, o que corresponde a 67% de Precisão. O mesmo critério foi aplicado nas 3 tabelas tendo sido calculada a média final (84%).

5

Conclusões

No domínio da classificação em *Machine Learning*, a vertente não supervisionada constitui um desafio acrescido pelo facto de não serem conhecidas as classes de cada amostra, impedindo assim, numa primeira fase, que se possa recorrer a implementações tecnológicas disponíveis para treino onde é exigido o conhecimento da classe de cada objecto.

Em particular, no *clustering* e classificação de documentos de texto não estruturado, isto é, texto cru, a seleção de atributos discriminantes é crucial. Nesta dissertação optou-se pela utilização dos termos que nos documentos concentram maior relevância semântica: as chamadas Expressões Relevantes e os unigramas (palavras singulares) relevantes.

Para o caso das Expressões Relevantes usámos o LocalMaxs que, apesar de não ser perfeito em termos de Precisão e Recall, é independente da língua. Para além disso, o uso deste extrator torna o processo da nossa abordagem computacionalmente mais simples e eficiente, uma vez que assim não serão usadas todas as expressões existentes no conjunto de documentos.

Para a deteção dos unigramas, decidimos excluir palavras irrelevantes e para isso usámos um método automático que se baseia no número de palavras distintas que cada unigrama tem à sua esquerda e direita e, através do gráfico traçado, na localização do ponto que sugere uma fronteira *natural* entre palavras relevantes e não relevantes: o *Elbow*. Este método mostrou ter uma grande Precisão nas experiências realizadas e, mesmo que existam algumas palavras irrelevantes que não tenham sido detetadas, é de se realçar que este processo é completamente automatizado e, tal como na seleção das Expressões Relevantes, não depende da língua usada nos documentos.

Esta primeira escolha de *features* (Expressões Relevantes e unigramas) resulta num número muito grande de atributos (milhares), havendo por isso a necessidade de reduzi-lo. Para tal, foi criada uma matriz de semelhança entre documentos que é calculada através de uma correlação. Com esta matriz foi possivel reduzir milhares de *features* a

apenas algumas dezenas ou centenas, dependendo do número de documentos a analisar. Pretendeu-se que, nas células desta matriz, fosse possivel ler a *classificação* indireta dos documentos através dos seus pares, sendo que valores altos indicariam, à partida, que os documentos pertencem à mesma classe.

De forma a obter uma tal matriz de semelhança com qualidade suficiente, isto é, que refletisse a organização correta dos *clusters* reais, através do valor das suas células, foi necessário desenvolver uma métrica para esse efeito.

Ainda assim, o número de *features* era relativamente grande pelo que se realizou uma última redução. O procedimento usado foi o PCA, através do qual conseguimos uma redução de cerca de 20% das features originais sem que houvesse perda de informação. Uma redução maior do número de *features* só seria possivel se estivessemos dispostos a perder parte significativa da informação original dos documentos, o que poderia afetar negativamente o *clustering* e a classificação.

Na fase de *clustering*, para os vários algoritmos de agrupamento testados, escolhemos como número de *clusters* aquele cujo agrupamento resultante apresentava maior valor de *silhueta*, para indicar o quão 'correto'/bom era o agrupamento. O algoritmo cuja *silhueta* indicou um número de *clusters* que coincidia sempre com o número real, foi o Spectral Clustering, com uma Precisão e Recall de agrupamento acima de 90%.

Para realizar a classificação aplicámos a técnica 'leave-one-out', como forma de obter treinos e classificações fiávies, mesmo com *data sets* relativamente reduzidos. Assim, com esta técnica foi possivel fazer uma avaliação dos resultados obtidos pelo SVM, que se traduziram numa Precisão e Recall semelhantes aos obtidos no *clustering* (acima de 90%).

Para além do agrupamento e classificação dos documentos, a extração dos tópicos constitui um objetivo importante desta dissertação uma vez que nos permite saber o conteúdo principal de cada *cluster* obtido. Para tal, foi criado um método que extraí os termos de acordo com a sua importância no respetivo *cluster*. Os resultados obtidos mostraram que o método é eficaz, com uma precisão de 84%. Uma vez que, não foram usados quaisquer ferramentas morfossintáticas (gramáticas, *part-of-speech taggers*, *stemers*, etc..) esta abordagem é independente da língua em que estão escritos os documentos.

Além do mais, todas as ferramemntas usadas ou desenvolvidas nesta dissertação têm natureza estatística, podendo aplicar-se a qualquer língua.

5.1 Possíveis Melhorias

Mesmo com resultados finais que apresentaram uma boa qualidade, é normal existirem melhorias que possam ser feitas posteriormente. Começando pelas expressões relevantes, tendo em conta que a Precisão do LocalMaxs é de apenas cerca de 70%, a seleção das mesmas pode ser melhorada através da inclusão de algum critério que permita excluir falsas expressões relevantes. No caso da deteção de palavras de fraca semântica (*stop*

words), apesar da Precisão nas experiências realizadas ser de 100%, também poderá ser possivel tentar melhorar o mecanismo de modo a detetar *stop words* que não foram detetadas, isto é, aumentar o Recall. Poder-se-ia, por exemplo, ter em conta tanto as palavras distintas junto a cada unigrama como as probabilidades das mesmas.

A matriz de semelhança entre documentos, ainda que apresente valores de semelhança relativamente altos para documentos da mesma classe, os seus valores podem ainda ser aperfeiçoados. Uma experiência que se poderia realizar, seria mexer nos pesos que tem cada uma das componentes da métrica conjunta $g(t,d)$ que foi usado para construir a matriz de semelhança.

O *clustering*, a classificação e a extração de tópicos, apesar de globalmente apresentarem bons resultados, estão relacionados com as etapas anteriores, e assim, melhorias que sejam feitas para as fases mencionadas anteriormente, irão, provavelmente, traduzir-se também em melhorias nestas últimas etapas.

Em particular, na fase de classificação, é vulgar atribuir-se ao objecto a classificar, a classe que é representada pelo centróide que 'está mais perto' ou 'menos distante' do vetor que representa o objecto (documento). No entanto, este vetor pode estar muito distante de qualquer centróide (aprendido) e, mesmo assim, estando mais perto de um deles, receberá como classe a que corresponde este centróíde, o que não é desejável. Por exemplo, vamos supor que para caracterizar pessoas utilizamos apenas a altura (por simplicidade), como única *feature*, e que, as amostras de treino se centram em 3 grupos com alturas médias de 0.8 metros, 1.3 metros e 1.7 metros, representando as crianças, os pré-adolescentes e adultos. Na fase de classificação, se quisermos atribuir um grupo a alguém com 1.35 metros não restam dúvidas de que, o centróide mais perto é o que tem a média de 1.3 metros. No entanto, se quisermos atribuir um grupo a 'alguém' com 3 metros de altura, é desejável que o classificador não atribua cegamente o grupo de centróide 1.7 metros, só porque é o que está mais perto de 3 metros. Uma possível solução para este problema seria considerar como ruído, os objetos cuja distância ao *cluster* mais próximo fosse superior a um determinado valor.

Bibliografia

[1] D. M. Blei, A. Y. Ng e M. I. Jordan. *Latent Dirichlet Allocation*. 2019. URL: https://dl.acm.org/doi/10.5555/944919.944937 (ver p. 8).

[2] G. E. P. Box e D. R. Cox. "An Analysis of Transformations". Em: *Journal of the Royal Statistical Society. Series B (Methodological)* 26.2 (1964), pp. 211–252. ISSN: 00359246. URL: http://www.jstor.org/stable/2984418 (ver p. 32).

[3] R. J. G. B. Campello, D. Moulavi e J. Sander. "Density-Based Clustering Based on Hierarchical Density Estimates". Em: *Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II*. Ed. por J. Pei et al. Vol. 7819. Lecture Notes in Computer Science. Springer, 2013, pp. 160–172. DOI: 10.1007/978-3-642-37456-2_14. URL: https://doi.org/10.1007/978-3-642-37456-2%5C_14 (ver p. 12).

[4] A. Doucet e M. Lehtonen. "Unsupervised Classification of Text-Centric XML Document Collections". Em: *Comparative Evaluation of XML Information Retrieval Systems, 5th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2006, Dagstuhl Castle, Germany, December 17-20, 2006, Revised and Selected Papers*. Ed. por N. Fuhr, M. Lalmas e A. Trotman. Vol. 4518. Lecture Notes in Computer Science. Springer, 2006, pp. 497–509. DOI: 10.1007/978-3-540-73888-6_46. URL: https://doi.org/10.1007/978-3-540-73888-6%5C_46 (ver pp. 16, 17).

[5] M. Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise". Em: AAAI Press, 1996, pp. 226–231 (ver p. 23).

[6] N. Fernandes et al. "Unification of HDP and LDA Models for Optimal Topic Clustering of Subject Specific Question Banks". Em: *CoRR* abs/2011.01035 (2020). arXiv: 2011.01035. URL: https://arxiv.org/abs/2011.01035 (ver p. 16).

[7] ". Joachims". *Text categorization with Support Vector Machines: Learning with many relevant features*. 2005. URL: https://link.springer.com/chapter/10.1007/BFb0026683 (ver p. 19).

[8] Y. Ko e J. Seo. "Automatic Text Categorization by Unsupervised Learning". Em: *COLING 2000, 18th International Conference on Computational Linguistics, Proceedings of the Conference, 2 Volumes, July 31 - August 4, 2000, Universität des Saarlandes, Saarbrücken, Germany*. Morgan Kaufmann, 2000, pp. 453–459. URL: https://www.aclweb.org/anthology/C00-1066/ (ver p. 17).

[9] X. Li et al. "Supervised labeled latent Dirichlet allocation for document categorization". Em: *Appl. Intell.* 42.3 (2015), pp. 581–593. DOI: 10.1007/s10489-014-0595-0. URL: https://doi.org/10.1007/s10489-014-0595-0 (ver p. 19).

[10] B. Liu et al. *Partially Supervised Classification of Text Documents*. 2002. URL: https://www.cs.uic.edu/~liub/S-EM/unlabelled.pdf (ver p. 19).

[11] A. Mackute-Varoneckiene e T. Krilavicius. "Empirical Study on Unsupervised Feature Selection for Document Clustering". Em: *Human Language Technologies - The Baltic Perspective - Proceedings of the Sixth International Conference Baltic HLT 2014, Kaunas, Lithuania, September 26-27, 2014*. Ed. por A. Utka et al. Vol. 268. Frontiers in Artificial Intelligence and Applications. IOS Press, 2014, pp. 107–110. DOI: 10.3233/978-1-61499-442-8-107. URL: https://doi.org/10.3233/978-1-61499-442-8-107 (ver p. 19).

[12] A. K. Mandal e R. Sen. "Supervised learning Methods for Bangla Web Document Categorization". Em: *CoRR* abs/1410.2045 (2014). arXiv: 1410.2045. URL: http://arxiv.org/abs/1410.2045 (ver p. 19).

[13] T. Mikolov et al. "Efficient Estimation of Word Representations in Vector Space". Em: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. Ed. por Y. Bengio e Y. LeCun. 2013. URL: http://arxiv.org/abs/1301.3781 (ver p. 8).

[14] B. Nigam et al. "Document Classification Using Expectation Maximization with Semi Supervised Learning". Em: *CoRR* abs/1112.2028 (2011). arXiv: 1112.2028. URL: http://arxiv.org/abs/1112.2028 (ver p. 19).

[15] M. F. Porter. "An algorithm for suffix stripping". Em: *Program* 14.3 (1980), pp. 130–137. DOI: 10.1108/eb046814. URL: https://doi.org/10.1108/eb046814 (ver p. 5).

[16] R. Ramesh. *Unsupervised-Text-Clustering using Natural Language Processing(NLP)*. 2019. URL: https://medium.com/@rohithramesh1991/unsupervised-text-clustering-using-natural-language-processing-nlp-1a8bc18b048d (ver p. 16).

[17] D. Rao, D. P e D. Khemani. "Corpus Based Unsupervised Labeling of Documents". Em: *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference, Melbourne Beach, Florida, USA, May 11-13, 2006*. Ed. por G. Sutcliffe e R. Goebel. AAAI Press, 2006, pp. 321–326. URL: http://www.aaai.org/Library/FLAIRS/2006/flairs06-063.php (ver p. 18).

[18] D. R. Recupero. "A new unsupervised method for document clustering by using WordNet lexical and conceptual relations". Em: *Inf. Retr.* 10.6 (2007), pp. 563–579. DOI: 10.1007/s10791-007-9035-7. URL: https://doi.org/10.1007/s10791-007-9035-7 (ver p. 18).

[19] J. Silva et al. "Using LocalMaxs Algorithm for the Extraction of Contiguous and Non-contiguous Multiword Lexical Units". Em: vol. 1695. Set. de 1999, pp. 113–132. ISBN: 978-3-540-66548-9. DOI: 10.1007/3-540-48159-1_9 (ver p. 21).

[20] M. Snow. *Unsupervised Document Clustering with Cluster Topic Identification*. 2018. URL: https://www.researchgate.net/publication/328530481_Unsupervised_Document_Clustering_with_Cluster_Topic_Identification (ver p. 17).

[21] X. Tao et al. "Unsupervised Multi-label Text Classification Using a World Knowledge Ontology". Em: *Advances in Knowledge Discovery and Data Mining - 16th Pacific-Asia Conference, PAKDD 2012, Kuala Lumpur, Malaysia, May 29-June 1, 2012, Proceedings, Part I*. Ed. por P.-N. Tan et al. Vol. 7301. Lecture Notes in Computer Science. Springer, 2012, pp. 480–492. DOI: 10.1007/978-3-642-30217-6_40. URL: https://doi.org/10.1007/978-3-642-30217-6%5C_40 (ver p. 18).