

MSC

2.º
CICLO

FCUP
2013

U. PORTO

Aplicação de Aprendizagem Ativa em Tarefas de
Extração de Opinião

Lucio Verdial

FC



Aplicação de Aprendizagem Ativa em Tarefas de Extração de Opinião

Lucio Verdial

Dissertação de Mestrado apresentada à
Faculdade de Ciências da Universidade do Porto em
Ciência de Computadores
2013



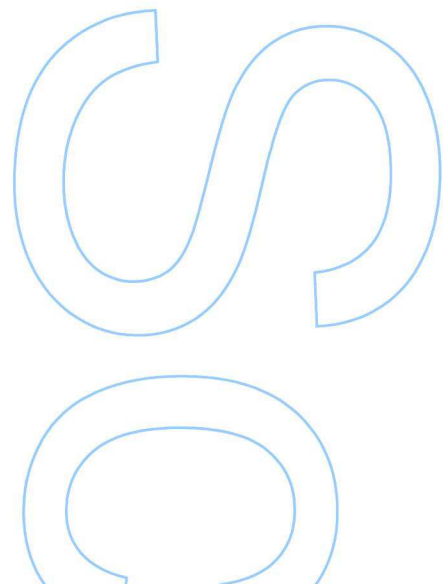
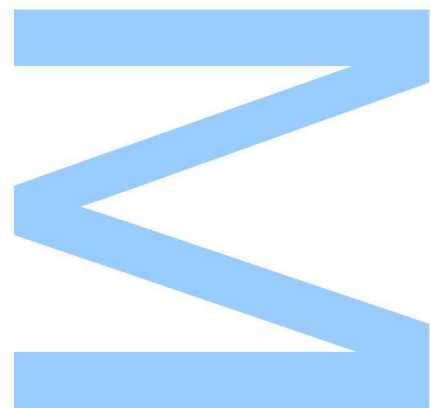
Aplicação de Aprendizagem Ativa em Tarefas de Extração de Opinião

Lucio Verdial

Mestrado em Ciência de Computadores
Departamento de Ciência de Computadores
2013

Orientador

Alípio Mário Guedes Jorge, Professor associado, Faculdade de Ciências da
Universidade do Porto





Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, ____/____/____

N

S

O

Dedico à minha família, principalmente aos meus pais:
Rogerio Verdial e Jacinta Dias
e em memória do meu avô
Manuel da Costa

Agradecimentos

Ao longo destes dois anos em que estive fora do meu país, longe da minha família e do meu povo timorense para estudar, enriquecer o meu conhecimento técnico e prático, aumentar a minha experiência, apreender coisas novas, ganhar nova experiência para que quando voltar o meu país poderia contribuir para o seu desenvolvimento.

Durante estes dois anos, tive oportunidade de contar com o apoio de diversas pessoas que, diretamente e indiretamente, contribuíram, desde que saí do meu país até escrever esta dissertação.

Estou especialmente grato ao meu orientador, Prof. Alípio Jorge, pela sua valiosa orientação, pela paciência, disponibilidade, conselho e ajuda para finalizar esta dissertação e terminar o meu mestrado.

Um especial agradecimento ao programa de Erasmus Mundus e toda a sua estrutura, pela ajuda financeira, pela paciência e disponibilidade da equipa para procurar a solução sobre quais quer problemas que enfrentei durante estes dois anos.

Um especial agradecimento à Universidade do Porto, Faculdade de Ciência, Departamento de Ciências dos Computadores porque me deixou entrar nesta universidade. Com muito orgulho fui estudante da Universidade do Porto.

Agradeço projeto de *e-Policy (Engineering the Policy-making LIfe Cycle)* [1] pela disponibilização de alguns dados e pelo enquadramento do problema.

Agradeço também ao Pedro Coelho e Brett Drury pela disponibilização de conjunto de dados.

Agradeço também aos meus amigos Adelaide Carvalho, Maria Pereira e José Carlos Marques, pelas suas ajudas na parte da língua portuguesa, principalmente pela paciência que tiveram em corrigir o meu português.

Por fim, mas não menos importante, à minha família. Especialmente aos meus pais: Rogerio Verdial e Jacinta Dias, os meus irmãos, especialmente a minha irmã mais velha Ludovina Verdial e o meu tio Domingos Verdial. Pois se não fossem eles, não vinha até Portugal.

Quero que saibam que entendo todas as dificuldades que vocês passaram para que eu cumprisse o meu objetivo, e agradeço sinceramente por toda a ajuda ao longo deste dois anos. Espero, de certo modo, recompensá-los com o meu sucesso. Para vocês todos, mais uma vez o meu sincero e profundo agradecimento.

Lucio Verdial

Resumo

A quantidade da informação no formato de texto na *Web* está continuamente a aumentar. Desenvolver um sistema que seja capaz de classificar automaticamente estes textos em função dos interesses dos utilizadores é muito desafiante.

A área de extração de conhecimento de texto (*text mining*) aborda o problema da classificação de texto. Tipicamente, a classificação de texto é utilizada para determinar os tópicos dos documentos, como por exemplo, Desporto, Educação ou Música. A extração de opinião (*opinion mining*) tem como objetivo a classificação de documentos em função dos sentimentos nelas expressam. Essa análise é dificultada pela subjetividade das frases e expressões utilizadas. Neste problema de extração de opinião a classificação de documentos é frequentemente baseada em palavras ou termos positivos ou negativos como: Bom, Mau, Melhor, Pior, Feio, Bonito, Gosta, Desgosta, etc.

Esta dissertação foca-se na extração de opinião, em que se classifica os documentos como positivos ou negativos.

As abordagens utilizadas nesta dissertação são: (i) calcular os *scores* dos documentos assumindo que temos uma base de dados de palavras positivas e negativas; (ii) extrair só as palavras positivas e negativas; (iii) procurar sinónimos na base de dados da *WordNet*. Estas abordagens designam-se como técnicas linguísticas.

Além das técnicas linguísticas, utilizam-se também vários modelos de representação de documento na matriz de indexação, nomeadamente Binário, Tf (*Term frequency*) e Tf-Idf (*Term frequency-Inverse document frequency*).

A utilização destas técnicas relativamente superficiais tem como objetivo melhorar o resultado obtido com as técnicas de extração de conhecimento de texto nas tarefas de extração de opinião.

Na aprendizagem supervisionada, precisamos de um conjunto de treino com informação da variável objetivo, com o tamanho suficiente para construir um modelo de classificação ou regressão. No nosso caso interessam-nos modelos de classificação. No caso de termos exemplos não etiquetados (sem informação da variável objetivo) a técnica de aprendizagem

ativa serve para selecionar automaticamente os exemplos a etiquetar que são mais favoráveis para a construção de um modelo de classificação. Esta seleção pode ser feita iterativamente.

Utilizam-se então três algoritmos de aprendizagem ativa para etiquetar os casos do conjunto de treino. Os algoritmos referidos são Confiança, Distância e D-confiança.

A utilização destes três algoritmos tem como objetivo reduzir os esforços de etiquetagem por parte do utilizador, mantendo um nível razoável de capacidade preditiva.

Na maioria dos problemas, a Confiança produz melhores resultados comparando com a D-confiança e a Distância. Em certos problemas, como se mostra nesta dissertação, a D-confiança tem melhores resultados. Enquanto a Distância não tem melhores resultados para nenhum dos problemas.

As experiências mostram que, na maioria dos conjuntos de dados podemos reduzir significativamente o número de exemplos necessários para obter um classificador comparável ao que se obtém a partir de todos os exemplos.

Abstract

The quantity of information in text format on the Web is continuously increasing. Develop a system that is able to automatically classify these texts as a function of users' interests is very challenging.

The area of the text mining approaches a text classification. Typically, text classification is used to identify topics in documents, such as Sports, Education or Music. The aim of Opinion Mining is to classify documents according to their sentiment. The subjectivity of the phrases used makes this analysis more difficult. In Opinion Mining document classification is instead based on negative and positive words or terms such as Best, Worst, Ugly, Beautiful, Like, Dislike, etc.

This thesis focuses on the opinion mining that classifies documents as positive or negative.

The approaches used in this thesis are: (i) calculation of the scores of the documents assuming that we have a database of positive and negative words, (ii) extracting only the positive and negative words (iii) looking for synonyms in the WordNet database. These approaches are designated as linguistic techniques.

Besides of the linguistic techniques, various models of representing document on document-term matrix are used, including Binary, Tf (Term Frequency) and Tf-Idf (Term frequency-Inverse document frequency).

Using of these techniques relatively surfaces with the objective to improve the results obtained with the text mining techniques in tasks of opinion mining.

In supervised learning, we need a training set with the information of target variable that is big enough to build a classification or a regression model. In our case we are interested in classification model. In the case we have unlabelled examples (without information on the target variable) active learning technique is used to automatically select the examples label that are more favourable for the construction of a classification model. This selection can be done iteratively.

We use three algorithms of active learning for labelling the cases of the training set. The

algorithms are referred as Confidence, Distance and D-confidence.

Using of these three algorithms with the objective to reduce the intensity of labelling by the user while maintaining a reasonable level of predictive capability.

In most problems, the Confidence produces better results when compared to the D-confidence and the Distance. In certain problems discussed in this thesis the D-confidence has better results while the Distance does not have best results for any problems.

The experiments show that in most datasets we can significantly reduce the number of samples to obtain a classifier comparable to that one obtained from the whole set of examples.

Conteúdo

Resumo	3
Abstract	5
Lista de Tabelas	11
Lista de Figuras	12
1 Introdução	13
1.1 Problemas	14
1.2 Hipóteses	15
1.3 A Nossa Abordagem	15
1.4 Estrutura da Dissertação	15
2 Extração de Opinião	16
2.1 Extração de Opinião	16
2.2 Classificação de Sentimento	19
2.2.1 Classificação baseada nas frases de sentimento	20
2.2.2 Usando métodos de classificação de texto	21
2.2.3 Classificação utilizando uma função <i>score</i>	22
2.3 Técnicas Utilizadas Nesta Dissertação	22
2.3.1 Matriz de indexação (<i>Document-term matrix</i>)	23

2.3.2	Pré-processamentos	23
2.3.2.1	Tokenização (<i>Tokenization</i>)	24
2.3.2.2	<i>Stopwords removal</i>	24
2.3.2.3	<i>Stemming</i>	24
2.3.2.4	Seleção de atributos	25
2.3.2.5	Outras tarefas de pré-processamento	25
2.3.3	Técnicas linguísticas	25
2.3.3.1	Calcular <i>Score</i>	25
2.3.3.2	Extrair palavras positivas e negativas (PosNeg)	26
2.3.3.3	Sinónimos	26
3	Aprendizagem Ativa	27
3.1	Aprendizagem Ativa (<i>Active learning</i>)	27
3.2	Confiança (<i>Confidence</i>)	28
3.3	Distância (<i>Farthest first</i>)	29
3.4	D-confiança (<i>D-confidence</i>)	30
4	Metodologia Experimental	32
4.1	Máquinas de Vetores Suporte - SVMs	32
4.1.1	SVMs lineares	33
4.1.2	SVMs não lineares	33
4.2	Medidas de Avaliação	34
4.3	Procedimento da Avaliação	35
4.3.1	Procedimento da avaliação da extração de opinião	35
4.3.1.1	Validação cruzada (<i>Cross-validation</i>)	35
4.3.2	Procedimento da avaliação da aprendizagem ativa	36
4.4	Os passos necessários para cada técnica utilizada nesta dissertação	36

5	Experiências e Resultados	39
5.1	Conjuntos de Dados	39
5.2	Resultado das Técnicas de Extração de Opinião	40
5.2.1	<i>Movie-Review-data (polarity dataset v1.0)</i>	41
5.2.2	<i>Airlines Food Reviews</i>	41
5.2.3	<i>Music Reviews</i>	42
5.2.4	<i>Teacher Reviews</i>	42
5.2.5	<i>Post-tagati</i> Economia	43
5.2.6	<i>Post-tagati</i> Tecnologia	43
5.2.7	<i>Post-tagati Environment</i>	43
5.3	Resultados das Técnicas de Aprendizagem Ativa	44
5.3.1	<i>Movie</i>	44
5.3.2	<i>Airline</i>	45
5.3.3	<i>Music</i>	46
5.3.4	<i>Teacher</i>	47
5.3.5	<i>Post-tagati</i> economia	48
5.3.6	<i>Post-tagati</i> tecnologia	49
5.3.7	<i>Post-tagati environment</i>	50
5.3.8	Resumo do resultado de aprendizagem ativa	50
6	Conclusão e Trabalho Futuro	52
6.1	Conclusão	52
6.1.1	Contribuição de trabalho	52
6.1.2	Conclusão de trabalho	52
6.2	Trabalho Futuro	53
A	Abreviaturas e Acrónimos	54

Lista de Tabelas

2.1	Padrão para extrair dois-palavras em documentos [24]	20
5.1	Conjunto de dados	39
5.2	<i>Movie</i> Pang (modelo base)	41
5.3	<i>Airline Food Reviews</i>	42
5.4	<i>Music</i>	42
5.5	<i>Teacher</i>	42
5.6	<i>Post-tagati</i> Economia	43
5.7	<i>Post-tagati</i> Tecnologia	43
5.8	<i>Post-tagati Environment</i>	44
5.9	Melhor Resultado	44

Lista de Figuras

3.1	Ciclo da Aprendizagem ativa [33]	28
3.2	Confiança	29
3.3	Distância	30
3.4	D-confiança [11]	31
4.1	Ilustração de hiperplano $h(x)$	33
4.2	Ilustração de fronteira de decisão em duas dimensões e três dimensões	34
5.1	<i>Movie</i>	44
5.2	<i>Airline</i>	45
5.3	<i>Music</i>	46
5.4	<i>Teacher</i>	47
5.5	<i>Post-tagati</i> Economia	48
5.6	<i>Post-tagati</i> Tecnologia	49
5.7	<i>Post-tagati</i> <i>Environment</i>	50

Capítulo 1

Introdução

O aproveitamento da tecnologia da *Web* pelas empresas para vender os seus produtos e o número de pessoas que usa essa tecnologia para fazer compras *online* está a aumentar cada vez mais. Muitos produtos estão a ser vendidos via *Web*, e cada vez mais pessoas estão a fazer compras por esta via.

Para perceber melhor o que o público pensa sobre um determinado bem ou serviço, normalmente as empresas reservam um espaço próprio para os seus clientes efetivos e potenciais escreverem as suas opiniões e comentários. Estes comentários são muito importantes para as empresas, nomeadamente para a tomada de decisão.

Após o nascimento da *Web* [25], as empresas e os indivíduos passaram a poder colocar informações na *Web* e assim facilmente outras entidades passaram a poder aceder-lhes. A *Web* permite que as empresas possam facilmente interagir com os seus clientes e vice-versa.

O aumento de popularidade de informação no formato texto na *Web*, levou ao aumento da relevância da técnica extração de conhecimento de texto (*text mining*) e extração de opinião (*opinion mining*).

A extração de conhecimento de texto (*text mining*) e a extração de opinião (*opinion mining*) são técnicas utilizadas atualmente para extrair informação de dados com o formato de texto não estruturado. Para limitar o nosso trabalho focalizar-nos-emos mais na extração de opinião (*opinion mining*), em detrimento da extração de conhecimento de texto (*text mining*). A extração de opinião é vulgarmente identificada como a análise de sentimento [25]. Trata-se do processamento de tipo de linguagem que o público utiliza ao comentar determinado bem, serviço ou assunto. A análise de sentimento implica a construção de um sistema para recolher e analisar as opiniões sobre determinado produto que surgem nos foruns da Internet, nos grupos de discussão, blogues, etc.

Um exemplo prático pode ser a decisão sobre a instalação de uma central eléctrica. Suponhamos que um governo quer decidir se deve instalar uma central eléctrica perto de uma cidade [1]. Necessita, obrigatoriamente, das opiniões, ideias e pensamentos das pessoas que moram perto dessa cidade. A maneira mais simples é aproveitar a vantagem da tecnologia de *Web* para criar um fórum de discussão, recolher e analisar as opiniões antes da tomada da decisão. Provavelmente, gerar-se-ão muitos dados e documentos o que faz com que seja difícil lidar com este problema manualmente. Haverá necessidade de encontrar especialistas para analisarem os dados e mesmo estes demorarão muito tempo a analisá-los.

A extração de conhecimento de texto (*text mining*) serviria para resolver este problema, mais especificamente a extração de opinião serviria para semi-automaticamente etiquetar os comentários como a favor ou contra determinada ideia. Até aqui, podemos simplesmente perceber que o problema é de classificação de texto. A classificação de texto é uma técnica de extração de conhecimento de texto (*text mining*) para classificar os documentos baseados nos tópicos.

No ponto de partida, os documentos não são classificados e etiquetados. Etiquetar estes documentos é uma tarefa muito pesada quando feita manualmente. Por esta razão surge a técnica de aprendizagem ativa (*active learning*): estratégia em que o computador seleciona os exemplos para etiquetar. Nesta dissertação aplicamos técnicas de aprendizagem ativa como Confiança (*Confidence*), Distância (*Farthest first*) e D-confiança (*D-confidence*), avaliamo-los num conjunto de dados existentes e etiquetados e aplicamo-los aos conjuntos não etiquetados. O trabalho inclui a seleção de um classificador e um modelo adequado para representação de texto, tendo em mente que este é um problema de extração de opinião.

1.1 Problemas

Assumimos que podemos facilmente obter documentos sobre comentários de determinados assuntos de um grupo de discussão ou blogues. Portanto a ideia é a seguinte: precisamos de encontrar uma técnica capaz de estudar estes documentos. Inicialmente propusemos a técnica da aprendizagem de máquina (*machine learning*) para abordar o problema. Sabemos também que a técnica de aprendizagem de máquina é uma aprendizagem supervisionada.

Na aprendizagem supervisionada, temos um conjunto de treino com tamanho suficiente para construir um modelo de classificação ou regressão. Para construir um modelo de classificação precisamos de um conjunto de treino em que os exemplos estejam etiquetados (têm os respetivos valores da variável objetivo ou classe). No entanto, a disponibilidade de exemplos etiquetados pode ser reduzida pois o custo de especialistas humanos para fazerem essa etiquetagem é elevado. As técnicas de aprendizagem ativa permitem seleccionar criteriosamente e de forma automática os exemplos a etiquetar. Assim, reduzem-se os custos

sem prejudicar significativamente a capacidade preditiva do modelo.

1.2 Hipóteses

O nosso trabalho tem duas hipóteses principais. A nossa hipótese relativamente à extração de opinião (*opinion mining*) é que a utilização de conhecimento linguístico relativamente superficial permite melhorar o resultado obtido com as técnicas de extração de conhecimento de texto nas tarefas de extração de opinião. A hipótese relativamente às técnicas existentes de aprendizagem ativa é que estas permitem reduzir os esforços de etiquetagem por parte do utilizador, mantendo um nível razoável de capacidade preditiva.

1.3 A Nossa Abordagem

Neste trabalho utilizámos técnicas de extração de texto aplicadas ao problema de extração de opinião. Para além dessas técnicas, usámos técnicas linguísticas baseadas em listas de palavras relacionadas com sentimentos e opiniões. Explorámos também o recurso WordNet, em particular a utilização de sinónimos.

Em relação à aprendizagem ativa, aplicámos a Confiança, a Distância e a D-confiança (vejam o capítulo 3) que são os três algoritmos utilizados para lidar com o problema do conjunto de treino que não está etiquetado, ou seja usamos estes algoritmos para preparar o conjunto de treino necessário ao algoritmo de aprendizagem de máquina (*machine learning*). Posteriormente, explicaremos detalhadamente cada um deste algoritmos.

Em termos de metodologia de avaliação, usámos validação cruzada (vejam capítulo 4, sub-subsecção 4.3.1.1) sobre 7 conjuntos de dados. O classificador base utilizado foi o *Support Vector Machine (SVM)* (vejam capítulo 4, secção 4.1).

1.4 Estrutura da Dissertação

Este documento encontra-se organizado de seguinte forma: começamos pela introdução do problema no capítulo 1; os capítulos 2 e 3 descrevem o estado da arte de extração de opinião e técnica de aprendizagem ativa; o capítulo 4 enuncia a metodologia experimental; o capítulo 5 descrevem as experiências realizadas e os resultados alcançados e, por fim, a conclusão e trabalho futuro são discutidos no capítulo 6.

Capítulo 2

Extração de Opinião

Informações digitais na forma texto não estruturado estão a aumentar no ambiente da *Web*. Hoje em dia, muitas pessoas estão interessadas em fazer pesquisas para arranjar uma maneira eficiente para transformar estas informações digitais em informações que sejam úteis para as empresas. Estas informações digitais contêm informação valiosa para as empresas. As empresas, de qualquer modo, precisam destas informações para tomar decisões.

2.1 Extração de Opinião

Extração de opinião é vulgarmente identificada como a análise de sentimento [25]. Trata-se do processamento de tipo de linguagem que o público utiliza ao comentar determinado bem, serviço ou assunto. A análise de sentimento implica a construção de um sistema para recolher e analisar as opiniões sobre determinado produto que surgem nos foruns da Internet, nos grupos de discussão, blogues, etc. Antes da *Web*, quando um indivíduo queria comprar alguma coisa, tipicamente perguntava aos seus amigos ou familiares antes de decidir o que comprar [25]. Quando uma empresa queria recolher as opiniões do público sobre uma determinado assunto ou produto, lançava inquéritos. Hoje, recorrendo-se à *Web* já não é preciso fazer inquéritos. As pessoas emitem opiniões e comentários sobre os produtos e serviços e expressam as suas ideias nos foruns da Internet, grupos de discussão e blogues criando o que coletivamente se designa por conteúdo gerado pelo utilizador (*user generated content*) [24].

Estes comentários contêm informação valiosa para as empresas. Os comentários na *Web* são apresentados normalmente sob a forma de texto não estruturado, facto que associado à quantidade crescente desses comentários torna muito difícil a extração de informação útil, sobretudo se for realizada manualmente por peritos humanos.

A tarefa é tecnicamente desafiante por causa da necessidade de processamento de linguagem natural que lhe é subjacente, mas também assume grande interesse prático [24] por contribuir para a tomada de decisão de muitos agentes económicos. Por exemplo, os gestores de empresas para a definição de várias estratégias necessitam de conhecer a opinião dos consumidores e do público em geral sobre os seus bens ou serviços e os potenciais clientes também querem saber as opiniões de clientes atuais antes de comprar um produto ou contratar um serviço. Ambos, têm interesse em extrair informação útil do conteúdo gerado pelo utilizador para fundamentarem as suas decisões.

Explorar e sumariar estes comentários ou opiniões para os converter em informação útil requer técnicas e mecanismos de extração de dados que permitam resolver, ainda que parcialmente, a complexidade e a grandeza dos conteúdos gerados pelos utilizadores. Há, por conseguinte, necessidade de aperfeiçoar e automatizar as técnicas existentes ou de criar outras novas para extrair dados e constitui-los em informação de modo a satisfazer critérios de eficiência de tempo e de recursos materiais e de eficácia da sua disponibilização aos agentes económicos. A extração de opinião ou análise de sentimento enquadra-se neste contexto.

Liu [24] refere as seguintes três tarefas principais da análise dos documentos que expressam opiniões ou sentimentos:

1. Classificação do sentimento (*Sentiment classification*).

A classificação do sentimento é um problema de classificação. Embora seja análoga à classificação do texto, o modelo subjacente à classificação de sentimento difere do modelo de classificação de texto. O modelo de classificação do texto classifica os documentos num tópico predefinido, por exemplo, política, ciência e desporto. Este modelo prefere as palavras dos tópicos ou dos títulos dos documentos em detrimento das palavras de opinião, por exemplo, ótimo, excelente, terrível, mau, pior.

Os documentos são classificados como positivos ou negativos. A classificação incide sobre o nível mais geral, o próprio documento, e não refere sequer os aspetos que as pessoas gostaram ou desgostaram. Assim por exemplo, da revisão de um produto, resulta apenas a classificação do documento como positivo ou negativo consoante o revisor.

2. Sumarização com base nas propriedades da opinião (*Featured-based opinion mining and summarization*).

A extração atua ao nível da frase para descobrir os detalhes ou propriedades dos aspetos que determinam os sentimentos positivos ou negativos relativos ao objeto em causa. O objeto pode ser um produto, um serviço, um indivíduo ou qualquer outra entidade.

Este modelo, proposto por Hu e Liu [18], formaliza mais o problema da extração de opinião. Identifica as partes importantes da informação e descreve a estrutura do

resultado da sumarização da opinião num texto não estruturado.

Classificar os documentos de texto ao nível de documento ou da frase não nos diz se o autor gosta ou não gosta. Um documento positivo não significa que o autor tem opinião positiva sobre todos os aspetos ou propriedades do objeto. Também um documento negativo não significa que o autor expressa uma opinião negativa sobre todos os aspetos ou propriedades do objeto [26]. No documento de revisão sobre um produto, o autor normalmente descreve os aspetos positivos e negativos, embora a opinião geral possa ser positiva ou negativa.

Para extrair detalhes dos aspetos ou das propriedades, precisamos do modelo de nível de propriedade ou *feature level*. Consideramos três técnicas importantes:

- Identificação das propriedades do objeto.

Por exemplo, na frase "*the Picture quality of this camera is amazing*", o objeto é a câmara e a propriedade é a qualidade da fotografia, "*picture quality*". Em [27] a propõe o modelo supervisionado de extração de padrão. E em [31] utiliza o modelo não supervisionado. A técnica, basicamente, procura substantivos frequentes nas frases como características que são geralmente utilizadas para descrever as propriedades dum objeto.

- Determinação da orientação da opinião.

Procura-se na frase determinar qual a opinião da propriedade do objeto que pode ser positiva, negativa ou neutral. Na frase referida a opinião da propriedade "*Picture quality*" é positiva. Mais uma vez, há muitas técnicas que podem ser utilizadas. Uma abordagem, que produziu bom resultado, baseou-se no léxico [18]. Esta abordagem basicamente utiliza as palavras da opinião na frase para determinar a orientação da opinião na propriedade do objeto.

- Agrupamento em sinónimos.

A mesma propriedade de um objeto poderia ser expressa em diferentes palavras e frases. Esta técnica agrupa estes sinónimos.

3. Extração de frases comparativas e relações (*Comparative sentence and relation mining*).

A extração compara diretamente um objeto contra um ou mais objetos similares. O objetivo é analisar e extrair as relações comparativas dentro das frases sobre determinado objeto. Por exemplo, o computador de marca x é melhor do que o computador de marca y. Em inglês, a relação é normalmente expressa na forma comparativa ou superlativa de um adjetivo ou advérbio. A forma comparativa é utilizada para expressar a superioridade ou inferioridade de um objeto relativamente a outro - o objeto é mais do que ou menos do que outro objeto. A forma superlativa é utilizada

para expressar se um objeto é melhor ou pior do que outro. Esta técnica é melhor explicada por Jindal e Liu [19].

Jindal e Liu [19] enunciaram duas tarefas para extrair informações comparativas:

- *"Dado um conjunto de documentos, identificar as frases comparativas que contêm e classificá-las em diferentes tipos ou classes";*
- *"Extraír relações comparativas das frases identificadas. Isso envolve a extração de entidades e das suas características que estão a ser comparadas, e palavras-chave comparativas".*

A relação expressa-se com: ($\langle relationWord \rangle, \langle features \rangle, \langle entityS1 \rangle, \langle entityS2 \rangle$), Por exemplo, a frase comparativa: *"Canon's optics is better than those of Sony and Nikon"*. A relação extraída é: (*better, optics, Canon, Sony, Nikon*).

Entre estas três tarefas, destacamos a primeira porque vai ser utilizada nesta dissertação para fazer a experiência técnica com o algoritmo de classificação de SVM e como técnica de aprendizagem ativa ou *active learning*. Procuramos a seguir aprofundar a tarefa de classificação de sentimento.

2.2 Classificação de Sentimento

Dado o conjunto de documentos D , um classificador classifica cada documento $d \in D$ em uma de duas classes, Positiva ou Negativa, Sim ou Não, Gostam ou não gostam [25]. Para explicar o conceito utilizamos classe Positiva e Negativa, mas no problema real podemos classificar em outras classes dependendo das suas interpretabilidades. Positiva significa que documento d expressa opinião positiva. Negativa significa que documento d expressa opinião negativa. Por exemplo dada um documento dos comentários sobre instalação de central eléctrica, e o sistema classifica-o automaticamente em classes, positiva ou negativa.

Existem muitas técnicas de classificação que utilizam as formas de aprendizagem de máquina para problemas de classificação [30]. O desenho de algoritmos específicos para a classificação de sentimento, também é muito utilizado sobretudo para explorar as palavras e as frases de opinião juntamente com funções de cálculo de valor [36].

A classificação de sentimento basicamente determina as orientações semânticas das opiniões sobre o objeto em cada documento que satisfaz a assunção referida.

2.2.1 Classificação baseada nas frases de sentimento

As frases de sentimento são as frases que descrevem os objetos. Estas frases contêm um adjetivo ou advérbio, porque segundo várias pesquisas [15, 16, 38] estes dois termos são bons indicadores de subjetividade e de opiniões.

Turney [36] desenhou um algoritmo para lidar com este problema, seguindo a ideia de se basear nas palavras positivas e negativas que existem em cada documento. Turney implementou este algoritmo para classificar a revisão dos clientes.

O algoritmo utiliza um processo de linguagem natural designado como *part-of-speech (POS) tagging*. O *part-of-speech* de uma palavra é uma estrutura linguística que define a sintaxe e a morfologia de palavra. *POS*, em inglês, são geralmente substantivos, verbos, adjetivos, advérbios, pronomes, preposições, conjunções e interjeições. Em [24] explica que Turney utiliza PennTreebank *POS Tags* padrões para identificar as frases.

O algoritmo do Turney consiste nos seguintes em três passos:

Passo 1. Extrair as frases que contêm um adjetivo ou advérbio uma vez que segundo várias pesquisas [15, 16, 38] estes dois termos são bons indicadores de subjetividade e de opiniões.

No entanto, apesar de um adjetivo isolado poder indicar subjetividade, é necessário um contexto suficiente para determinara sua orientação (ou parecer) semântica. Por exemplo, o adjetivo "*unpredictable*" pode ter uma orientação negativa de uma revisão automotiva, numa frase como "*unpredictable steering*", mas poderia ter uma orientação positiva numa revisão de um filme se surgisse numa frase como "*unpredictable plot*". Por isso, o algoritmo extrai duas palavras consecutivas, em que um membro do par é um adjetivo ou advérbio e o outro é uma palavra de contexto. As duas palavras consecutivas que se extraem devem seguir as regras definidas na tabela 2.1. Por exemplo, na primeira linha da tabela 2.1, a primeira palavra tem que ser um adjetivo e a segunda tem que ser um substantivo e a terceira que não se extrairá poderá ser uma qualquer.

Tabela 2.1: Padrão para extrair dois-palavras em documentos [24]

Primeira Palavra	Segunda Palavra	Terceira Palavra (não extrairá)
JJ	NN ou NNS	Qualquer
JJ	JJ	Nao NN nem NNS
NN ou NNS	JJ	Nao NN nem NNS
RB, RBS, RBR	JJ	Nao NN nem NNS
RB, RBS,RBR	VB,VBD,VCN ou VBG	Qualquer

Por exemplo, na frase "*this camera produces beautiful pictures*", extrai-se "*beautiful pictures*".

Passo 2. Utiliza *pointwise mutual information* para calcular a semântica de orientação.

$$PMI(termo1, termo2) = \log \frac{Pr(termo1 \wedge termo2)}{Pr(termo1)Pr(termo2)} \quad (2.1)$$

$Pr(termo1 \wedge termo2)$ é a probabilidade de ocorrerem ambos os termos, termo1 e termo2. E $Pr(termo1)Pr(termo2)$ é a probabilidade de ocorrência dos dois termos se forem estatisticamente independentes. O rácio entre $Pr(termo1 \wedge termo2)$ e $Pr(termo1)Pr(termo2)$ é a medida do grau de dependência estatística entre os termos. O log deste rácio é a quantidade de informação que se adquire sobre a presença de uma das palavras, quando observamos a outra. A orientação semântica ou opinião (SO) de uma frase é calculada com base na sua associação com a palavra referência positiva "excelente" e sua associação com a palavra referência negativa "pior":

$$SO(frase) = PMI(frase, "excelente") - PMI(frase, "pior") \quad (2.2)$$

As probabilidades são calculadas mediante as consultas efetuadas a um motor de busca e coletado número de *hits*. Para cada consulta de pesquisa, um motor de busca geralmente dá o número de documentos relevantes para a consulta, que é o número de *hits*.

$$SO(frase) = \log 2 \frac{hits(frase \text{ Near } "excelente") \cdot hits("péssimo")}{hits(frase \text{ Near } "péssimo") \cdot hits("excelente")} \quad (2.3)$$

Para evitar a divisão por zero, 0,01 é adicionado aos *hits*

Passo 3. Dada uma revisão, o algoritmo calcula a média de todas as frases SO na revisão, e classifica a avaliação como recomendado se a média SO for positiva, caso contrário é não recomendado. Precisa de notar que, esta experiência utilizou *AltaVista Advance Search Engine* para calcular função *Near*, para mais detalha vejam [36].

2.2.2 Usando métodos de classificação de texto

A abordagem mais simples para classificação de sentimento consiste em tratar o problema como se fosse um tema de um problema de classificação de texto. Então, pode usar-se qualquer algoritmo de classificação de texto [30], por exemplo, o naïve Bayesian, SVM, kNN, etc. A abordagem foi experimentada por [30], com as revisões de filme de duas classes, positivos e negativos. Demonstrou-se que o uso de um unigram (um saco de palavras individuais) na classificação resulta num bom desempenho, usando quer o naïve Bayesian quer o SVM.

2.2.3 Classificação utilizando uma função *score*

Dave [9], propuseram este algoritmo, consiste em dois passos:

Passo 1. Calcular o *score* para cada termo em conjunto de treino com equação:

$$\boxed{score(t_i) = \frac{(Pr(t_i|C) - Pr(t_i|C'))}{(Pr(t_i|C) + Pr(t_i|C'))}} \quad (2.4)$$

Em que t_i é um termo e C é a classe, C' é complemento, não C , $Pr(t_i|C)$ é a probabilidade condicionada do termo t_i na classe C é calculada pelo contar o número do termo t_i ocorre na classe C e o divide pelo número total dos termos nas opiniões de classe C . o *score* de um termo é -1 ou 1.

Passo 2. Para classificar o novo documento $d_i = t_1...t_n$, o algoritmo somando os *scores* de todos os termos e utilizando o total para determinar a classe.

$$\boxed{class(d_i) = \begin{cases} C & eval(d) > 0 \\ C' & caso \text{ contrário} \end{cases}} \quad (2.5)$$

Em que,

$$\boxed{eval(d_i) = \sum_j score(t_j)} \quad (2.6)$$

O algoritmo foi aplicado a grandes conjuntos de dados, e os resultados de *accuracy* variam entre (84.6%-88.3%), sem utilizar o *stemming* e retirar *stopwords* [9].

2.3 Técnicas Utilizadas Nesta Dissertação

Pang [30] tratou o problema de extração de opinião com aprendizagem supervisionada ou método de *Machine Learning* (Aprendizagem de Máquina) para classificar os documentos em duas classes: a classe positiva, se o documento expressa um sentimento positivo e a classe negativa se o documento expressa sentimento negativo.

Na representação dos documentos, Pang utilizou o *framework* de *bag-of-word* e avaliou-o com o SVM, Naive Bayes e Maximum Entropy.

A nossa abordagem é um pouco diferente, mas baseia-se na ideia base de Pang [30]. Utilizamos o pré-processamento e representamos os documentos com matriz de indexação de *bag-of-word*, mas com diferentes modelos, nomeadamente Binário, Tf e Tf-Idf. Aproveitamos a lista das palavras positivas e negativas de Liu para calcular os *scores* e juntá-los à matriz de

indexação, extraímos palavras positivas e negativas e procurámos os sinónimos das palavras na base de dados da *WordNet* [2].

2.3.1 Matriz de indexação (*Document-term matrix*)

Tipicamente, os documentos são representados por vetores característicos [14]. O modelo mais utilizado é *bag-of-word* que é muito simples e utiliza todas as palavras nos documentos como característicos e os diferentes números de palavras designados como os números de dimensões.

A forma clássica para organizar os *bag-of-word* é *document-term matrix*. O *Document-term matrix* é uma matriz multidimensional que descreve a frequência dos termos que ocorrem em todos os documentos em que as colunas são as palavras ou termos e as linhas são os documentos.

Há vários esquemas para determinar as entradas das linhas ou valor das linhas, o mais simples é modelo binário, em que 1 significa que este termo ocorre e 0 significa que não ocorre. Um outro esquema é utilizar modelo de Tf (*term frequency*) que representa os valores com base no número de frequências dos termos no documento. O mais utilizado é Tf-Idf, dado o peso à palavra **w** no documento **d**

$$TF - IDFWeight(w, d) = TermFreq(w, d) * \log\left(\frac{N}{DocFreq(w)}\right) \quad (2.7)$$

Em que TermFreq(w, d) é a frequência de uma palavra do documento, N é o número de todos os documentos, DocFreq(w) é o número de documentos que contém a palavra w [14].

2.3.2 Pré-processamentos

A extração de opinião pode ser definida como uma sub-disciplina da linguística computacional, que se concentra em extrair a opinião das pessoas a partir da web ou documento de texto não estruturado [29]. A extração de opinião (*opinion Mining*) também pode ser classificada como um sub tópico de *Text Mining*. Na *Text Mining*, tanto como na extração de opinião, são necessárias algumas técnicas de pré-processamento, como por exemplo, remoção as palavras insignificantes *stopwords*, tokenização (*tokenization*), redução de raízes das palavras (*stemming*), eliminação de pontuação, conversão em minúsculas, remoção de espaços brancos, eliminação de números, redução de termos, *bag-of-word*, *part-of-speech*, entre outros. Estas técnicas encaixam numa fase muito importante tanto na análise de texto como na análise de opinião. Estas técnicas permitem que os nossos documentos sejam interpretáveis em ambiente de computação no nosso caso *R Project*. Além disso,

simultaneamente, reorganiza e reestrutura os dados de forma tabular afim de que seja acessível mais facilmente.

Existem inúmeras técnicas de pré-processamento. Os pré-processamentos têm vantagens e também têm desvantagens: a vantagem é, por exemplo, reduzir o tamanho de dimensão enquanto a desvantagem é a inevitável perda de informação importante. Para limitar o nosso trabalho, a seguir tentamos abordar as técnicas que julgamos mais importantes e utilizamos no nosso trabalho prático.

2.3.2.1 Tokenização (*Tokenization*)

Uma abordagem mais frequente na *Text Mining* (TM) e extração de opinião é a Tokenização. Os documentos, inicialmente, estão em forma de frase ou de parágrafos. Em TM é necessário que estas frases fiquem em partições por palavras ou termos [14]. O conceito mais popular para representar esta abordagem é *bag-of-word*. Representar os documentos numa coleção não-ordenada de palavras, gramática e até ignorando a ordem das palavras. Esta transformação é feita pela tokenização.

2.3.2.2 *Stopwords removal*

Stopwords são as palavras insignificantes numa linguagem que ajudam a construir as frases, mas não representam qualquer conteúdo dos documentos [24]. Artigos, preposições e conjunções e alguns pronomes são candidatos a *stopwords*. Essas palavras são aquelas que são tratadas e removidas na fase de pré-processamento de dados antes de indexar os documentos. Tratamos as *Stopwords* como se fosse a aplicação técnica de seleção de termos ou *Feature Selection* em inglês. Esta técnica é muito importante, porque ajuda a reduzir a dimensionalidade do nosso problema.

2.3.2.3 *Stemming*

Refere-se a um processo de redução das palavras à sua raiz [24]. Raiz de uma palavra é aquela que fica após remoção dos seus prefixos e sufixos. Por exemplo as palavras do inglês, "computer", "computing" e "compute" são reduzidos para "comput" e "walks", "walking" e "walker" são reduzidos para "walk". Claramente que o modo *stemming* aumenta o valor de *recall* e diminui o número de termos a considerar na indexação do documento [24]. No entanto, isso pode influenciar a precisão porque muitos documentos irrelevantes podem ser considerados relevantes [24]. Por exemplo, ambas "cop" e "cope" são reduzidas para raiz "cop". No entanto, se alguém procurar documentos sobre a "police", que contém apenas "cope" é improvável que seja relevante. Embora muitas experiências sejam realizadas pelos

investigadores, ainda não há evidência conclusiva de uma forma ou de outra. Na prática, deve-se experimentar com a coleção de documentos na mão para ver se *stemming* ajuda. O melhor algoritmo de *stemming* é o algoritmo do Porter(1997) [13].

2.3.2.4 Seleção de atributos

- Extração de termos importantes

Uma das maneiras para reduzir a dimensão do nosso problema é a extração dos termos ou *Feature Extraction* [32] que consiste em criar um novo conjunto de dados onde os termos são originalmente de um conjunto anterior. Um exemplo simples e utilizamos no nosso trabalho consiste em extrair palavras positivas e negativas do conjunto original.

- *Document frequency*

O document frequency também é importante na redução da dimensão. A ideia é retirar todos os termos que têm baixa frequência. Sebastiani abordou a ideia com mais detalhe [28, 32].

2.3.2.5 Outras tarefas de pré-processamento

Retirar a pontuação, converter tudo em minúsculas, retirar os espaços brancos, retirar números. Estes pré-processamentos não são menos importantes, principalmente a redução do número de termos ou atributos e a dimensão do problema para que seja tratável computacionalmente e caber na memória do computador.

2.3.3 Técnicas linguísticas

2.3.3.1 Calcular *Score*

No nosso caso, utilizamos a lista PosNeg (palavras positivas e negativas) de Liu e a lista de negação para calcular os *scores* e juntarmos-los à matriz de indexação como um atributo adicional. O *score* é calculado procurando as frequências das palavras da lista de Liu nos documentos. Formalmente, podemos definir como:

$$\boxed{score = N_{palavrapositiva} - N_{palavranegativa}} \quad (2.8)$$

em que, N palavra positiva é a cardinalidade de palavras positivas e N palavra negativa é a cardinalidade de palavras negativas.

2.3.3.2 Extrair palavras positivas e negativas (PosNeg)

No nosso caso, utilizámos a lista PosNeg de Liu e a lista de negação (exemplo: *don't*, *not*, *doesn't*, *didn't* ect.) para extrair palavras positivas e negativas dos documentos.

2.3.3.3 Sinónimos

Esta técnica é mais utilizada na análise semântica para identificar o sinónimo de um termo ou palavra, isto é, identifica as diferentes palavras que contêm o mesmo significado. O exemplo mais prático, neste contexto, consiste em procurar os sinónimos das palavras positivas e negativas na base de dados de *WordNet* [13].

Terminamos a descrição do estado da arte da extração de opinião, incidindo na tarefa e técnica de extração de opinião cuja aplicação descrevemos no capítulo 5. Descrevemos também os pré-processamentos e técnicas utilizados nesta dissertação. A seguir vamos falar sobre aprendizagem ativa.

Capítulo 3

Aprendizagem Ativa

Existem vários esquemas para o problema de classificação, nomeadamente as aprendizagens: supervisionada, não supervisionada, semi-supervisionada e ativa. A aprendizagem supervisionada (*supervised learning*) pressupõe a existência de um conjunto de treino já etiquetado [7,22]. A etiquetagem exaustiva é uma tarefa muito pesada se estivermos perante um grande número de casos. A preparação do conjunto de treino etiquetado não é uma tarefa fácil e associa-se ao elevado custo de tempo e trabalho, principalmente se etiquetarmos manualmente todos os casos do conjunto de treino. Na aprendizagem supervisionada normalmente, selecionam-se os casos já etiquetados aleatoriamente para treinar o modelo. Mas não se resolve o problema dos casos não etiquetados.

A aprendizagem não supervisionada não requer qualquer conjunto etiquetado, mas os utilizadores não podem efetuar agrupamentos à medida das suas necessidades específicas e não têm garantia de que os clusters induzidos estejam alinhados com as classes de aprendizagem [12]

A aprendizagem semi-supervisionada (*semi-supervised learning*) combina as ideias de aprendizagens supervisionada e não supervisionada [7]. Temos de ter, portanto, necessariamente um subconjunto de casos etiquetados e outro subconjunto não etiquetado. Continua a ser necessário, portanto, obter casos etiquetados. Neste caso, podemos aproveitar a técnica de aprendizagem ativa (*active learning*).

3.1 Aprendizagem Ativa (*Active learning*)

A aprendizagem ativa [33] inicia-se com um pequeno conjunto de treino que cresce por aplicação de um processo iterativo que vai etiquetando o conjunto. Ou seja, começa-se por treinar o modelo com conjunto L, sendo L um pequeno conjunto etiquetado e U um conjunto não etiquetado. O conjunto U é maior do que conjunto L. Na primeira iteração, treina-se o

modelo com conjunto L , escolhe-se o caso de U que for mais informativo, isto é, o que estiver associado à menor confiança. Pede-se, então, ao perito humano para etiquetar este caso e depois junta-se ao conjunto L . Repetem-se as iterações até à satisfação de alguns critérios de paragem que foram previamente definidos pelo utilizador ou, até, serem etiquetados todos os casos de conjunto U .

Existem vários tipos de aprendizagem ativa, a diferenciação entre uns e outros reside no processo de escolha do caso mais informativo em cada iteração [33]. Na figura 3.1, mostra-se uma ilustração do aprendizagem ativa baseado no *pool-based*

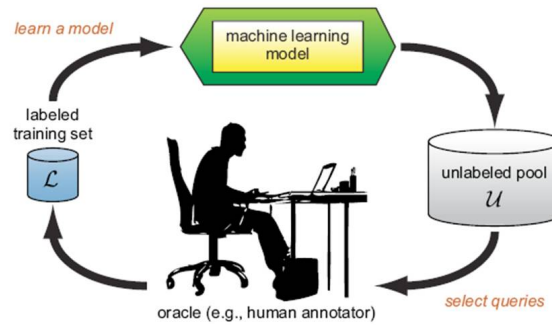


Figura 3.1: Ciclo da Aprendizagem ativa [33]

Sequidamente destacamos três técnicas de aprendizagem ativa, nomeadamente Confiança, Distância e D-confiança. Estes três algoritmos serão implementados na parte prática (capítulo 5).

Antes de explicar os algoritmos de aprendizagem ativa, explicamos primeiro as anotações utilizadas para representar e explicar os algoritmos.

Nas figuras 3.2, 3.3 e 3.4: W é o conjunto de trabalho (*working set*), representativo da amostra do problema. L_i é o subconjunto de W que já está etiquetado na iteração i . U é o subconjunto de W que não está etiquetado. Na iteração i , U_i é o conjunto de diferença entre W e L_i ; C_i é o conjunto das classes conhecidas no L_i ; K é o número de classes, c_k ; h_i representa modelo classificador na iteração i ; q_i representa a consulta (*query*) na iteração i ; $Conf_i(u_j, c_k)$ são as confianças posteriores na classe c_k dado caso u_j , na iteração i . $dist(u_j, c_k)$ é a distância agregada para classe conhecida, calculado por $ClassDist(u_j, c_k)$ baseando-se na distância individual entre cada par dos casos.

3.2 Confiança (*Confidence*)

A ideia de confiança é utilizar a noção de probabilidade para selecionar o caso com a menor probabilidade para ser o próximo a ser etiquetado [10]. Portanto, a ideia é etiquetar os casos

não etiquetados com a previsão da menor probabilidade, isto é a que tem mais incerteza [23]. Intuitivamente, a abordagem de confiança é usar modelo de classificação em cada iteração para calcular a confiança posterior em cada caso etiquetado para cada caso não etiquetado, ou seja, usa os casos etiquetados na iteração i para treinar o classificador, h_i ; e depois calcular a confiança para cada caso não etiquetado, ordenar o resultado de confiança e selecionar o caso com menor confiança para etiquetar e se juntará ao conjunto etiquetado.

```

Algorithm Confidence
(1) Given  $W, L_1$ 
(2)  $i = 1$ 
while not stopping criteria do
    (3)  $U_i = W - L_i$ 
    (4)  $C_i =$  distinct class labels in  $L_i$ 
    (5) learn  $h_i$  from  $L_i$ 
    (6) apply  $h_i$  to  $U_i$  generating  $conf_i(u_j, c_k)$ 
    (7)  $q_i = \text{argmin}_{uj}(conf_i(u_j, c_k))$ 
    (8)  $L_{i+1} = L_i \cup \langle q_i, \text{label}(q_i) \rangle$ 
    (9)  $i++$ 
end while

```

Figura 3.2: Confiança

Resumidamente, na figura 3.2 em cima, os passos mais importantes são 5 e 6, passo 5 é treinar o modelo com o conjunto L_i , enquanto passo 6 para calcular a confiança de cada caso não L_i . Nesta fase, associamos a confiança de cada classe para cada caso. O Passo 7 é para selecionar o caso com menor confiança a ser etiquetado e juntaremos ao conjunto L . Repete o passo 3 até passo 9, até i satisfazer algum critério definido pelo utilizador (critério de paragem). No nosso caso, deixamos até etiquetar todos os casos do conjunto U .

3.3 Distância (*Farthest first*)

Trata-se de um algoritmo que é utilizado para calcular a aproximação do problema de otimização do *clustering K-Centre* [17]. KFF para a aprendizagem ativa [4] consiste no seguinte: Dado o conjunto atual de L , caso etiquetado, escolhemos um caso $x \in U$, que esteja mais distante de L . Usa-se $L \cup (x, y)$ como conjunto, para treinar o modelo do classificador. A próxima consulta (*query*) é o caso mais distante do conjunto observado (mais diferente do conjunto observado).

Resumidamente, na figura 3.3 em cima, começamos por conjunto W e L . No Passo 2 calcula-se a distância entre todos os casos com a função de distância euclidiana. Inicializa-se a iteração no passo 3. No Passo 4 separa-se o conjunto U do conjunto W . Passo 6 é treinar o modelo com o conjunto L_i . Passo 7 e 8 são para selecionar o caso mais distante (mais

```

Algorithm Kernel Farthest-First
(1) Given  $W, L_1$ 
(2) compute distance among instances in  $W$ 
(3)  $i = 1$ 
while not stopping criteria do
  (4)  $U_i = W - L_i$ 
  (5)  $C_i =$  distinct class labels in  $L_i$ 
  (6) learn  $h_i$  from  $L_i$ 
  for ( $u_j \in U_i$ ) do
    for ( $c_k \in C_i$ ) do
      (7)  $ClassDist(u_j, c_k) = Distmin(u_j, c_k)$ 
      (8)  $dist_i(u_j, c_k) = ClassDist(u_j, c_k)$ 
    end for
  end for
  (9)  $q_i = \text{argmin}_{u_j}(dist_i(u_j))$ 
  (10)  $L_{i+1} = L_i \cup \langle q_i, label(q_i) \rangle$ 
  (11)  $i++$ 
end while

```

Figura 3.3: Distância

diferente) do conjunto L_i , baseando no resultado do passo 2. Repete o passo 4 até passo 11, até i satisfazer algum critério definido pelo utilizador (critério de paragem). No nosso caso, deixamos até etiquetar todos os casos do conjunto U .

3.4 D-confiança (*D-confidence*)

Escudeiro e Jorge [11] propuseram uma nova técnica que designaram por D-confiança. Dado o conceito alvo, um conjunto de classes e uma amostra de casos não etiquetados extraída do espaço alvo, procura-se identificar todas classes do espaço alvo, efetuando poucas consultas e minimizar o erro de precisão da aprendizagem. Entretanto no nosso caso, não nos interessa muito a identificação de classes no espaço alvo.

O objetivo do Escudeiro e Jorge [11] é identificar os casos que representam todos os outros e cobrem todas as classes de modo a diminuir o número de consultas ao conjunto de espaço alvo (*working set*). As consultas requerem etiquetagem para um caso específico. Todos os modelos de aprendizagem ativa pressupõem que, pelo menos, no *working set*, cada caso tenham uma classe.

Normalmente, a aprendizagem ativa depende da confiança do classificador para selecionar a consulta [3] e do pressuposto que o conjunto pré-etiquetado cobre todas as classes etiquetadas. No entanto, a técnica de D-confiança [11] é um pouco diferente, O D-confiança, pesa a confiança do classificador como inverso da distância entre o caso em questão e as classes já conhecidas, ou seja calcular o rácio de d-confiança com a confiança sobre a distância. A equação:(3.1) é para calcular o rácio de d-confiança. Espera-se que este desvio favoreça a rápida cobertura do espaço, apresentando uma tendência para explorar áreas desconhecidas.

Como consequência, a convergência torna-se mais rápida do que se utilizássemos apenas a confiança.

$$dConf(u) = \max_k \left(\frac{conf(u_j, c_k)}{dist(u_j, c_k)} \right) \quad (3.1)$$

Em que, $conf(u_j, c_k)$ é a confiança posterior na classe c_k dado caso u_j na iteração i e $dist(u_j, c_k)$ é a distância agregada para classe conhecida, calculado por $ClassDist(u_j, c_k)$ baseando na distância individual entre cada par dos casos. Para mais informações em baixo na figura 3.4 mostramos o algoritmo de D-confiança.

Algorithm 1 D-Confidence algorithm

```

(1) given  $W, L_1$ 
(2) compute distance among instances in  $W$ 
(3)  $i = 1$ 
while not stopping criteria do
  (4)  $U_i = W - L_i$ 
  (5)  $C_i =$  distinct class labels in  $L_i$ 
  (6) learn  $h_i$  from  $L_i$ 
  (7) apply  $h_i$  to  $U_i$  generating  $conf_i(u_j, c_k)$ 
  for ( $u_j \in U_i$ ) do
    for ( $c_k \in C_i$ ) do
      (8)  $dist_i(u_j, c_k) = ClassDist(u_j, c_k)$ 
      (9)  $dconf_i(u_j, c_k) = \frac{conf_i(u_j, c_k)}{dist_i(u_j, c_k)}$ 
    end for
  (10)  $dConf_i(u_j) = \max_{c_k} (dconf_i(u_j, c_k))$ 
  end for
  (11)  $q_i = \underset{u_j}{\operatorname{argmin}}(dConf_i(u_j))$ 
  (12)  $L_{i+1} = L_i \cup \langle q_i, label(q_i) \rangle$ 
  (13)  $i++$ 
end while

```

Figura 3.4: D-confiança [11]

Resumidamente, na figura 3.4, os passos basicamente são dos passos de algoritmo confiança e algoritmo distância. O que difere é no passo 9, em que calcular o rácio de D-confiança por confiança sobre distância, equação:(3.1).

Acabamos de abordar o estado de arte da aprendizagem ativa, começamos basicamente pela ideia geral do conceito aprendizagem ativa até destacamos três algoritmos recentes, nomeadamente Confiança (Confidence), Distância (Farthest first) e D-confiança (D-Confidence). Estes três algoritmos serão implementados na parte prática (capítulo 5). A seguir vamos falar sobre metodologia experimental, em que basicamente falamos sobre algoritmo de classificação SVM, procedimento e medidas de avaliação.

Capítulo 4

Metodologia Experimental

Na parte de prática, utilizamos o SVM como o nosso classificador, taxa de erro (*error rate*) como a nossa medida de avaliação dos erros, validação cruzada como método de avaliação. Já seguir abordamos os significados de cada conceito.

4.1 Máquinas de Vetores Suporte - SVMs

A máquina de vetores suporte (*Support Vector Machines- SVMs*) foi inicialmente introduzida por Vladimir Vapnik [8]. O SVM é considerado o algoritmo mais usado no problema de classificação [39]. O SVM superou os outros algoritmos, por exemplo, o KNN e naive Bayes no problema de classificação de texto [20], Tong [34] implementou o SVM com a aprendizagem ativa (*active learning*) e o resultado foi melhor. Pang [30] compararam o algoritmo SVM com o Naive Bayes e o Maximum Entropy, na classificação de sentimento. O SVM continuou a ser o favorito, pois, o resultado superou os dos dois outros algoritmos.

SVM é um algoritmo de aprendizagem supervisionada em que há dois conjuntos de dados: o conjunto de treino e o conjunto de teste. O conjunto de treino é usado para treinar o modelo ou classificador e o conjunto de teste para testar o modelo. No conjunto de treino as classes são etiquetadas, ou são classes conhecidas, enquanto o conjunto de teste não precisa de classes. A ideia fundamental do SVM é encontrar um hiperplano que separe perfeitamente as duas classes [6] o que é fácil para casos lineares, mas no caso real os problemas nem sempre estão linearmente separados. Os artigos [6,21] explicam melhor o SVM no contexto matemático.

4.1.1 SVMs lineares

Neste problema a ideia é procurar o hiperplano que separa melhor as classes. Tipicamente há muitos hiperplanos, mas escolhemos aquele que tem margem máxima, conhecido por *maximum margin hyperplane*.

Formalmente, dado X como o conjunto de treino com n objetos $x_i \in X$ e com as respectivas etiquetas $y_i \in Y$, em que X é o conjunto de entrada e $Y = \{-1, +1\}$ são as possíveis etiquetas (classes). X é linearmente separável, se é possível separar os objetos das classes -1 e $+1$ por um hiperplano [21].

O hiperplano é calculado por:

$$h(x) = w \cdot x + b \quad (4.1)$$

em que, $w \cdot x$ é o produto escalar entre w e x , $w \in X$ é o vetor normal ao hiperplano e $\frac{2}{\|w\|}$ é a distância do hiperplano em relação à origem, com $b \in \mathbb{R}$.

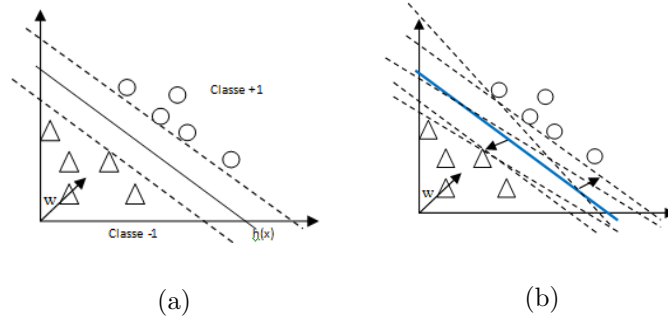


Figura 4.1: (a) Ilustração de hiperplano $h(x)$, que separa linearmente as duas classes $\{-1, +1\}$, (b) O melhor hiperplano é aquele que tem margem máxima e está representado pela linha azul.

4.1.2 SVMs não lineares

Neste problema, SVM mapear o conjunto de treino do seu espaço original para um novo espaço de maior dimensão (multidimensional) [21, 35], depois procura-se o hiperplano, que neste novo espaço, capaz de separar as classes. Calcular o produto interno neste espaço novo é computacionalmente pesado, portanto, utiliza-se outra variante de kernel que em vez de calcular os produtos internos num espaço multidimensional, aproveita o facto de que $K(x; z) = \Phi(x) \cdot \Phi(z)$, portanto simplesmente substituem-se os produtos internos por cálculos do kernel, muito mais simples no espaço de entrada original (e muito menor) [35].

Um kernel é uma função $K()$ tal que para todo $x, z \in X$

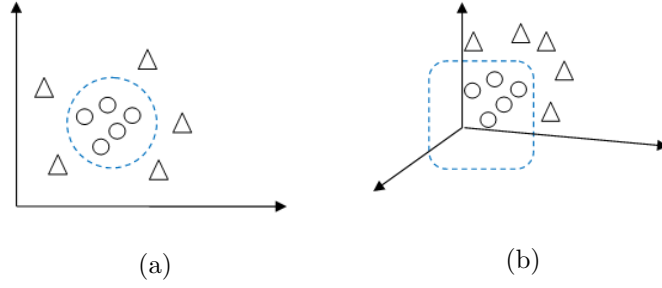


Figura 4.2: (a) Ilustração de conjunto de dados e da fronteira de decisão em duas dimensões, (b) Ilustração do exemplo de transformação de duas dimensões em três dimensões e fronteira de decisão.

$$K(x; z) = \Phi(x) \cdot \Phi(z) \quad (4.2)$$

Em que Φ é mapeamento de espaço original X para um espaço novo com tamanho multidimensional Z . Existem três funções de kernel mais conhecidas:

Gaussian Kernel:

$$K(X_i; X_j) = e^{\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)} \quad (4.3)$$

Kernel Polinomial de grau d :

$$K(X_i; X_j) = (X_i; X_j)^d \quad (4.4)$$

Kernel radial:

$$K(X_i; X_j) = e^{-\gamma \|X_i - X_j\|^2} \quad (4.5)$$

A razão que nos leva a escolher o SVM é muito simples, porque o SVM como um algoritmo de aprendizagem da máquina teve sucesso no problema de classificação especialmente no problema com alta dimensionalidade, como por exemplo classificação de texto [11, 20, 34, 37].

4.2 Medidas de Avaliação

Há muitas medidas de avaliação da classificação, as mais utilizadas são nomeadamente *accuracy*, taxa de erro, *precision* e *recall*, *F-measure* [24]. No nosso caso, utilizamos a taxa de erro.

Taxa de erro é para medir a proporção do erro de previsão do modelo do classificador [21, 35]. Portanto, formalmente, dado conjunto de teste, obtemos a previsão com o modelo de classificador, para calcular a taxa de erro ($L_{0/1}$), precisamos a informação das verdadeiras classes do conjunto do teste.

$$L_{0/1} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} I(\hat{h}_0(X_i), Y_i) \quad (4.6)$$

Em que, $I(\cdot)$ é a função do indicador, se $X=Y$ então $I(X,Y)=0$, caso contrário $I(X,Y)=1$, $\hat{h}_0(X_i)$ valor de previsão do caso i , Y_i é a verdadeira classe do caso i . O valor de ($L_{0/1}$) é variar entre 0 e 1, quanto mais próximo de 0 é considerado melhor erro. Outra maneira de calcular o erro de classificador é utilizando matriz de confusão [21], em que as linhas representam as classes verdadeiras e as colunas representam as classes de previsão. A matriz de confusão M tem dimensão de $k \times k$, k é o número classes. A diagonal desta matriz apresenta os acertos do classificador e os outros elementos são os erros cometidos pelo classificador. Portanto, para calcular o erro é:

$$erro : 1 - \frac{\sum_i M_{i,i}}{\sum_{i,j} M_{i,j}} \quad (4.7)$$

4.3 Procedimento da Avaliação

4.3.1 Procedimento da avaliação da extração de opinião

Depois de construir o modelo de classificação, é necessário que avaliar o seu erro de classificação. Nesta avaliação basicamente é para saber a fiabilidade do nosso modelo. Para medir estes erros é necessário que responder as seguintes perguntas: qual o parâmetro utilizado para medir o erro? -medida de avaliação (vejam na seção 4.2) e como avaliar? -método da avaliação

4.3.1.1 Validação cruzada (*Cross-validation*)

Existem muitos métodos de avaliação da classificação, entre eles, método *Hold-Out* em que dividir aleatoriamente o conjunto em dois subconjuntos, 70% como conjunto de treino e 30% como conjunto de teste [21, 24]. O *Hold-Out* é normalmente utilizado no caso em que o conjunto de dados é razoavelmente grande.

Método *Random Subsampling*, em que aleatoriamente escolher conjunto de teste e conjunto de treino e repetir o várias vezes, no final calcular erro que corresponde à média dos erros em cada repetição. Entretanto o mais interessante para o nosso problema e já usaram no

problema de classificação do texto [5], aprendizagem ativa [10, 12, 34] e extração opinião [30] é método *Cross-Validation* (validação Cruzada). A ideia de validação Cruzada é similar ao *Random subsampling*, essencialmente consiste em k repetições de algumas parte de conjunto de dados como conjunto de treino e o resto para testar [21]. Inicialmente, começa por dividir conjunto de dados em k subconjunto com mesmo tamanho, para cada subconjunto utilizado como conjunto de teste e o resto de $k-1$ como conjunto de treino. Mais uma vez, vai produzir o k erro, o erro final é calculado por média pesada dos k .

4.3.2 Procedimento da avaliação da aprendizagem ativa

Para avaliar os três algoritmos de aprendizagem ativa, no nosso caso, para cada AA (aprendizagem ativa) na primeira iteração selecionamos aleatoriamente dois casos para etiquetar manualmente e depois treinamos o classificador (o nosso caso, utilizamos SVM como o classificador). Em seguida previmos os restos dos exemplos, baseando nos resultados da previsão, começamos por simplesmente selecionar os próximos casos dependendo do critério de seleção ou consulta de cada AA. Já falámos no capítulo 3. Portanto, formalmente como explicámos em cima, utilizamos 5f-cv (*5 folds-Cross validation*) (utilizamos apenas 5f-cv porque os conjuntos de dados são bastante grande, portanto seria muito lenta se utilizar mais *folds*) para avaliar a fiabilidade do nosso modelo. Inicialmente, dividimos aleatoriamente o conjunto de dados em 5 partes dos mesmos tamanhos. Basicamente nós temos 5 iterações (5f-cv), em cada iteração seleciona dois casos pré-etiquetados (conjunto L_i) para treinar o classificador, h_i ; e testá-lo com o conjunto não etiquetado, U_i ; depois baseando-se no método de cada aprendizagem, nomeadamente baseando-se na Confiança, Distância, e D-confiança, seleciona dez casos para juntar com conjunto U. Guardamos sempre a taxa de erro de cada iteração. Portanto até etiquetar todos os casos não etiquetados em U, teríamos n taxa de erro em que n é o número de iteração.

4.4 Os passos necessários para cada técnica utilizada nesta dissertação

Seguidamente apresentamos a explicação de cada técnica utilizada nesta dissertação:

1. Matriz de indexação para representar os documentos;

No nosso caso, antes de implementar a matriz de indexação, fizemos os pré-processamentos - remover *Stopwords*, remover pontuação, *Stemming*, espaços brancos e os *Sparseterms*.

Passos necessários:

- Calcular o *score* para cada documento;

- Implementar o pré-processamento e representar os documentos com matriz de indexação;
- Pré-processamento sem *Sparseterms*;
- Pré-processamento com *Sparseterms* de 97% até 99%;
- Implementar SVM com 5f-cv;

2. Palavras positivas e negativas (PosNeg);

Nesta abordagem, aproveitámos a lista de PosNeg de Liu e a lista das palavras de negação (exemplo: *don't*, *not*, *doesn't*, *didn't* ect.) para extrair os atributos interessantes nos documentos. Além disso, calculámos os *scores*. Utilizamos os pré-processamentos como 1ª abordagem.

Passos necessários:

- Calcular o *score* para cada documento;
- Extrair os atributos interessantes baseados na lista de Liu (positivos e negativos) e lista de negação (exemplo: *don't*, *not*, *doesn't*, *didn't*, ect.);
- Implementar o pré-processamento e representar os documentos com matriz de indexação;
- Pré-processamento sem *Sparseterms*;
- Pré-processamento com *Sparseterms* de 97% até 99%;
- Criar novo conjunto de dados compostos por matriz de indexação, *score* e as etiquetas;
- Implementar SVM com 5f-cv.

3. Sinónimos das palavras positivas e negativas

Nesta abordagem, aproveitamos as palavras extraídas na 2ª abordagem para enriquecer os atributos através da procura dos seus sinónimos na base de dados do *WordNet*.

Passos necessários:

- Calcular o *score* para cada documento;
- Extrair os atributos interessantes baseados na lista de Liu (positivos e negativos) e lista de negação (exemplo: *don't*, *not*, *doesn't*, *didn't* ect.);
- Enriquecer os atributos com os seus sinónimos na base de dados do *WordNet*;
- Implementar pré-processamento e representar os documentos com Tf-Idf;
- Pré-processamento sem *Sparseterms*;
- Pré-processamento com *Sparseterms* de 97% até 99%;
- Criar novo conjunto de dados compostos por matriz de indexação, *score* e as etiquetas;

- Implementar SVM com 5f-cv.

Tivemos a falar sobre o algoritmo SVM, metodologia de avaliação como é que avaliar o nosso resultado, medidas utilizadas e agora seguidamente vamos entrar na parte prática, designado como experiência prática e resultado do capítulo 5. Resumidamente, uma parte sobre conjunto de dados utilizados, resultados dos classificação de texto e resultados dos aprendizagem ativa.

Capítulo 5

Experiências e Resultados

5.1 Conjuntos de Dados

Tabela 5.1: Conjunto de dados

Conjunto de Dados	Número de Casos	Distribuição das Classes
<i>movie-review-data (polarity dataset v1.0)</i>	1400	positiva: 0.5 negativa: 0.5
<i>Teacher Reviews</i>	2893	good: 0.756 poor: 0.244
<i>Music Reviews</i>	651	positive: 0.232 negative: 0.768
<i>Airline Food Reviews</i>	1389	positiva: 0.749 negativa: 0.251
<i>Post-tagati</i> Economia	579	positiva: 0.363 negativa: 0.637
<i>Post-tagati</i> Tecnologia	467	positiva: 0.237 negativa: 0.763
<i>Post-tagati</i> Environment	71	positiva: 0.437 negativa: 0.563

- *Movie-Review-data (polarity dataset v1.0)*

Este conjunto de dados encontra-se disponível no:

(<http://www.cs.cornell.edu/people/pabo/movie-review-data/>). Foi o conjunto de dados que utilizou por Pang [30] para fazer a sua experiência. O conjunto de dados composto por 700 casos positivos e 700 casos negativos.

- *Teacher Reviews*

Conjunto de dados foi utilizado por Brett Drury, encontra-se disponível no: <http://www.goo.gl/IHL6V>, o conjunto de dados composto por 2893 casos, 2188 dos casos *Good* e 701 dos casos *Poor*.

- *Music Reviews*

O conjunto de dados utilizado por Brett Drury, encontra-se disponível no: <http://www.goo.gl/IHL6V>, tem 651 casos, as classes positivas estão com as etiquetas de 3, 4 ou 5 (500 casos) e as classes negativas estão com as etiquetas de 0 ao 2 (151 casos).

- *Airline Food Reviews*

O conjunto de dados utilizado por Brett Drury, encontra-se disponível no: <http://www.goo.gl/IHL6V>, tem 1389 casos, as classes positivas estão com as etiquetas de 7, 8, 9 ou 10 (1040 casos) e as classes negativas estão com as etiquetas de 1, 2 ou 3 (349 casos).

- *Post-tagati Economia*

Este conjunto de dados é o problema do *ePolicy*, composto por 579 casos e originalmente o conjunto de dados estava classificado no *ranking* do seguinte, -2 ao 2. Regularizou-se para duas classes, *ranking* entre -2 até 0 classificamos como Negativo (369 casos) e 1 até 2 como classes Positivo (210 casos).

- *Post-tagati Tecnologia*

Este conjunto de dados é o problema do *ePolicy*, composto por 467 casos e originalmente o conjunto de dados estava classificado no *ranking* do seguinte, -2 ao 2. Regularizou-se para duas classes, *ranking* entre -2 até 0 classificamos como Negativo (356 casos) e 1 até 2 como classes Positivo (111 casos).

- *Post-tagati Environment*

Este conjunto de dados é o problema do *ePolicy*, composto por 71 casos e originalmente o conjunto de dados estava classificado no *ranking* do seguinte, -2 ao 2. Regularizou-se para duas classes, *ranking* entre -2 até 0 classificamos como Negativo (40 casos) e 1 até 2 como classes Positivo (31 casos).

5.2 Resultado das Técnicas de Extração de Opinião

A seguir encontra-se as tabelas com os resultados obtidos com os conjuntos utilizados. Como referimos no capítulo anterior, utilizámos vários modelos de representação dos documentos

na matrix de indexação (Tf, Binário e Tf-Idf). Fizemos vários pré-processamentos, nomeadamente *TermSparse* de 97% ao 99%, Não usar *TermSparse* ou com parâmetro *Default*, Retirar pontuações e Espaços brancos. Utilizámos também as técnicas linguísticas: calcular os *Scores*, utilizar só palavras positivas e negativas (PosNeg) para construir matriz de indexação. Em gerar os resultados do erro variam de um conjunto de dados para outro conjunto de dados. Além disso os melhores resultados são obtidos com diferentes pré-processamentos e parâmetros consoante o conjunto de dados.

Os valores de erro obtido para os vários conjuntos de dados são todos melhores do que prever a classe mais frequente (ver tabela 5.9). São também comparáveis com valores encontrados na literatura, quando disponíveis. os valores NA presentes nas tabelas indicam que a execução do algoritmo não foi possível devido ao elevado número de atributos ou porque se atingem o limite da memória disponível.

5.2.1 *Movie-Review-data (polarity dataset v1.0)*

Este conjunto de dados foi utilizado por Pang [30] na sua experiência. Os resultados da experiência dele foi melhor do que nosso, mas em condições experimentais não exatamente iguais às nossas. O nosso resultado é 0.1829 de erro (utilizando o modelo Tf-Idf, os passos de pré-processamento *Termsparse* (97%), *Stemming* e *Stopwords* e a técnica linguística de *score*). O valor de erro obtido prevendo sempre a classe mais frequente é de 0.5, claramente pior do que o nosso melhor modelo de classificação. O resultado está na tabela 5.2.

Tabela 5.2: *Movie* Pang (modelo base)

Peso /pré-processamento	BoW			PosNeg			Sinónimos		
	Tf	Bin	Tf-Idf	Tf	Bin	Tf-Idf	Tf	Bin	Tf-Idf
Default	NA	NA	NA	0.3436	0.5329	0.5329	0.3436	0.5329	0.5329
TermSparse_99%	0.2607	NA	0.2050	0.2336	0.2043	0.2379	0.2336	0.2043	0.2379
TermSparse_98%	0.2300	0.1943	0.1907	0.2300	0.2264	0.2236	0.2300	0.2264	0.2236
TermSparse_97%	0.2229	0.1986	0.1914	0.2271	0.2129	0.2171	0.2271	0.2129	0.2171
Default+score	NA	NA	NA	0.3107	0.3207	0.3393	0.3107	0.3207	0.3393
TermSparse_99%+score	0.2514	NA	0.2050	0.2271	0.2007	0.2314	0.2271	0.2007	0.2314
TermSparse_98%+score	0.2207	0.1864	0.1893	0.2257	0.2207	0.2193	0.2257	0.2207	0.2193
TermSparse_97%+score	0.2164	0.1871	0.1829	0.2207	0.2071	0.2157	0.2207	0.2071	0.2157

5.2.2 *Airlines Food Reviews*

O erro para conjunto *Airlines Food Reviews* é 0.2036, utilizando o modelo Tf-Idf, os passos de pré-processamento *Termsparse* (98%), *Stemming* e *Stopwords* e a técnica linguística de *score* e palavras Positivas e Negativas. O valor de erro obtido prevendo sempre a classe mais frequente é de 0.25, pior do que o nosso melhor modelo de classificação. O resultado está na tabela 5.3.

Tabela 5.3: *Airline Food Reviews*

Peso /pré-processamento	Tf	BoW		Tf	PosNeg		Tf	Sinónimos	
		Bin	Tf-Idf		Bin	Tf-Idf		Bin	Tf-Idf
Default	0.2513	0.2513	0.2513	0.2513	0.2513	0.2462	0.2513	0.2513	0.2462
TermSparse_99%	0.2455	0.2274	0.2361	0.2101	0.2188	0.2123	0.2101	0.2188	0.2123
TermSparse_98%	0.2166	0.2173	0.2274	0.2087	0.2094	0.2036	0.2087	0.2094	0.2036
TermSparse_97%	0.2188	0.2245	0.2325	0.2195	0.2231	0.2152	0.2195	0.2231	0.2152
Default+score	0.2513	0.2513	0.2513	0.2513	0.2513	0.2462	0.2513	0.2513	0.2462
TermSparse_99%+score	0.2455	0.2274	0.2361	0.2101	0.2181	0.2108	0.2101	0.2181	0.2108
TermSparse_98%+score	0.2166	0.2173	0.2282	0.2094	0.2087	0.2036	0.2094	0.2087	0.2036
TermSparse_97%+score	0.2195	0.2245	0.2325	0.2195	0.2238	0.2173	0.2195	0.2238	0.2173

5.2.3 Music Reviews

O erro para conjunto *Music Reviews* é 0.1538, utilizando o modelo Tf-Idf, os passos de pré-processamento *Termsparse* (97%), *Stemming* e *Stopwords*. O valor de erro obtido prevendo sempre a classe mais frequente é de 0.23, pior do que o nosso melhor modelo de classificação. O resultado está na tabela 5.4.

Tabela 5.4: *Music*

Peso /pré-processamento	Tf	BoW		Tf	PosNeg		Tf	Sinónimos	
		Bin	Tf-Idf		Bin	Tf-Idf		Bin	Tf-Idf
Default	0.2246	0.2323	0.2323	0.2323	0.2323	0.2323	0.2323	0.2323	0.2323
TermSparse_99%	0.2231	0.2015	0.2246	0.2200	0.2277	0.2246	0.2200	0.2277	0.2246
TermSparse_98%	0.2062	0.1846	0.1754	0.2077	0.2169	0.2200	0.2077	0.2169	0.2200
TermSparse_97%	0.2000	0.1769	0.1538	0.2338	0.2323	0.2492	0.2338	0.2323	0.2492
Default+score	0.2246	0.2323	0.2323	0.2323	0.2323	0.2323	0.2323	0.2323	0.2323
TermSparse_99%+score	0.2231	0.2015	0.2246	0.2200	0.2262	0.2277	0.2200	0.2262	0.2277
TermSparse_98%+score	0.2077	0.1846	0.1754	0.2077	0.2169	0.2231	0.2077	0.2169	0.2231
TermSparse_97%+score	0.2000	0.1769	0.1538	0.2354	0.2323	0.2446	0.2354	0.2323	0.2446

5.2.4 Teacher Reviews

O erro para conjunto *Teacher Reviews* é 0.1699, utilizando o modelo Tf-Idf, os passos de pré-processamento *Termsparse* (99%), *Stemming* e *Stopwords* e a técnica linguística de *score* e palavras positivas e negativas. O valor de erro obtido prevendo sempre a classe mais frequente é de 0.24, pior do que o nosso melhor modelo de classificação. O resultado está na tabela 5.5.

Tabela 5.5: *Teacher*

Peso /pré-processamento	Tf	BoW		Tf	PosNeg		Tf	Sinónimos	
		Bin	Tf-Idf		Bin	Tf-Idf		Bin	Tf-Idf
Default	NA	NA	NA	0.2377	0.2429	0.2391	0.2377	0.2429	0.2391
TermSparse_99%	0.9983	0.9983	0.9983	0.1727	0.1768	0.1699	0.1727	0.1768	0.1699
TermSparse_98%	0.9979	0.9983	0.9983	0.1789	0.1799	0.1844	0.1789	0.1799	0.1844
TermSparse_97%	0.9983	0.9983	0.9983	0.1889	0.1920	0.1889	0.1889	0.1920	0.1889
Default+score	NA	NA	NA	0.2377	0.2426	0.2391	0.2377	0.2426	0.2391
TermSparse_99%+score	0.9983	0.9983	0.9983	0.1716	0.1761	0.1699	0.1716	0.1761	0.1699
TermSparse_98%+score	0.9986	0.9983	0.9983	0.1789	0.1792	0.1841	0.1789	0.1792	0.1841
TermSparse_97%+score	0.9983	0.9983	0.9983	0.1896	0.1913	0.1900	0.1896	0.1913	0.1900

5.2.5 *Post-tagati* Economia

O erro para este conjunto de dados é 0.3130, utilizando o modelo Tf-Idf, os passos de pré-processamento *Termsparse* (98%), *Stemming* e *Stopwords*. O valor de erro obtido prevendo sempre a classe mais frequente é de 0.36, pior do que o nosso melhor modelo de classificação. O resultado está na tabela 5.6.

Tabela 5.6: *Post-tagati* Economia

Peso /pré-processamento	BoW			PosNeg		
	Tf	Bin	Tf-Idf	Tf	Bin	Tf-Idf
Default	NA	NA	NA	0.3652	0.3652	0.3652
TermSparse_99%	0.3357	0.3270	0.3165	0.3478	0.3478	0.3304
TermSparse_98%	0.3287	0.3235	0.3130	0.3635	0.3739	0.3530
TermSparse_97%	0.3287	0.3217	0.3287	0.3652	0.3809	0.3461
Default+score	NA	NA	NA	0.3670	0.3670	0.3670
TermSparse_99%+score	0.3357	0.3270	0.3165	0.3496	0.3443	0.3357
TermSparse_98%+score	0.3287	0.3235	0.3130	0.3600	0.3687	0.3548
TermSparse_97%+score	0.3287	0.3217	0.3339	0.3670	0.3826	0.3357

5.2.6 *Post-tagati* Tecnologia

O erro para este conjunto de dados é 0.1935, utilizando o modelo Tf-Idf, os passos de pré-processamento *Termsparse* (99%), *Stemming* e *Stopwords*. O valor de erro obtido prevendo sempre a classe mais frequente é de 0.24, pior do que o nosso melhor modelo de classificação. O resultado está na tabela 5.7.

Tabela 5.7: *Post-tagati* Tecnologia

Peso /pré-processamento	BoW			PosNeg		
	Tf	Bin	Tf-Idf	Tf	Bin	Tf-Idf
Default	0.2387	0.2387	0.2387	0.2387	0.2387	0.2387
TermSparse_99%	0.2129	0.2129	0.1935	0.2452	0.2452	0.2387
TermSparse_98%	0.2194	0.2194	0.2108	0.2344	0.2366	0.2409
TermSparse_97%	0.2215	0.2172	0.2129	0.2409	0.2473	0.2409
Default+score	0.2387	0.2387	0.2387	0.2387	0.2387	0.2387
TermSparse_99%+score	0.2129	0.2129	0.1957	0.2452	0.2473	0.2387
TermSparse_98%+score	0.2194	0.2194	0.2108	0.2344	0.2366	0.2409
TermSparse_97%+score	0.2215	0.2172	0.2129	0.2409	0.2473	0.2430

5.2.7 *Post-tagati* Environment

O erro para este conjunto de dados é 0.4143, utilizando o modelo Binário, os passos de pré-processamento *Termsparse* (97%), *Stemming* e *Stopwords*. O valor de erro obtido prevendo sempre a classe mais frequente é de 0.44, pior do que o nosso melhor modelo de classificação. O resultado está na tabela 5.8.

Tabela 5.8: *Post-tagati Environment*

Peso /pré-processamento	Tf	BoW		Tf	PosNeg	
		Bin	Tf-Idf		Bin	Tf-Idf
Default	0.4286	0.4286	0.4286	0.4286	0.4286	0.4286
TermSparse_99%	0.4286	0.4286	0.4286	0.4286	0.4286	0.4286
TermSparse_98%	0.4286	0.4286	0.4286	0.4286	0.4286	0.4286
TermSparse_97%	0.4571	0.4143	0.4571	0.4571	0.4143	0.4571
Default+score	0.4286	0.4286	0.4286	0.4286	0.4286	0.4286
TermSparse_99%+score	0.4286	0.4286	0.4286	0.4286	0.4286	0.4286
TermSparse_98%+score	0.4286	0.4286	0.4286	0.4286	0.4286	0.4286
TermSparse_97%+score	0.4714	0.4143	0.4571	0.4714	0.4143	0.4571

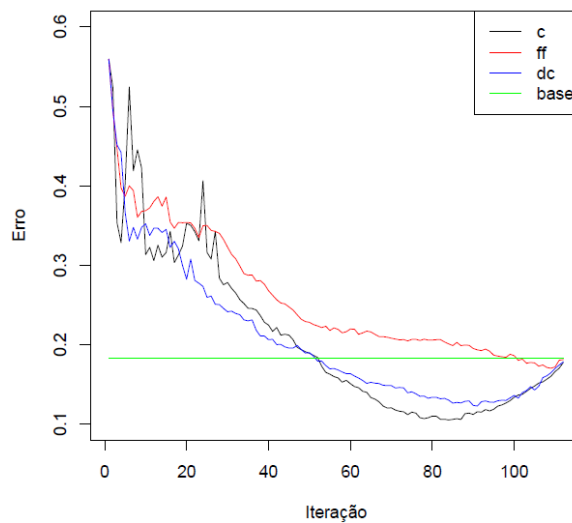
Tabela 5.9: Melhor Resultado

Conjunto de dados	Melhor erro	Peso	Termsparse	Stemming	Stopwords	Score	PosNeg	Erro default
<i>Movie</i>	0.1829	Tf-Idf	97%	Sim	Sim	Não	Não	0.50
<i>Airlines</i>	0.2036	Tf-Idf	98%	Sim	Sim	Sim	Não	0.25
<i>Music</i>	0.1538	Tf-Idf	97%	Sim	Sim	Não	Não	0.23
<i>Teacher</i>	0.1699	Tf-Idf	99%	Sim	Sim	Sim	Sim	0.24
<i>Post-tagati Economia</i>	0.3130	Tf-Idf	98%	Sim	Sim	Não	Não	0.36
<i>Post-tagati Tecnologia</i>	0.1935	Tf-Idf	99%	Sim	Sim	Não	Não	0.24
<i>Post-tagati Environment</i>	0.4143	Bin	97%	Sim	Sim	Não	Não	0.44

5.3 Resultados das Técnicas de Aprendizagem Ativa

Encontram-se em baixo os resultados das experiências utilizando técnicas de aprendizagem ativa (AA), portanto cada subsecção é um problema em que mostra-se um gráfico e as respetivas explicações. As anotações utilizadas nos gráficos são: c como algoritmo de Confiança, dc como algoritmo de D-confiança e ff como algoritmo de Distância.

5.3.1 *Movie*

Figura 5.1: *Movie*

A figura 5.1, representa o resultado da técnica de AA do conjunto de dados *Movie Reviews*, o eixo X representa os erros de previsão, eixo Y representa as iterações (cada iteração seleciona 10 casos). Utilizando o modelo Tf-Idf, os passos de pré-processamento *TermSparse* (97%), *Stemming* e *Stopwords* e a técnica linguística de *Score*.

Como podemos observar na figura 5.1, pode-se concluir que o algoritmo com melhor resultado é o algoritmo Confiança. A Confiança consegue atingir 0.1052 de erro, na 84ª iteração (por volta de 832 casos etiquetados). À medida que o conjunto de exemplos etiquetados cresce o erro tende para o da abordagem supervisionada (*baseline*): 0.1829.

5.3.2 Airline

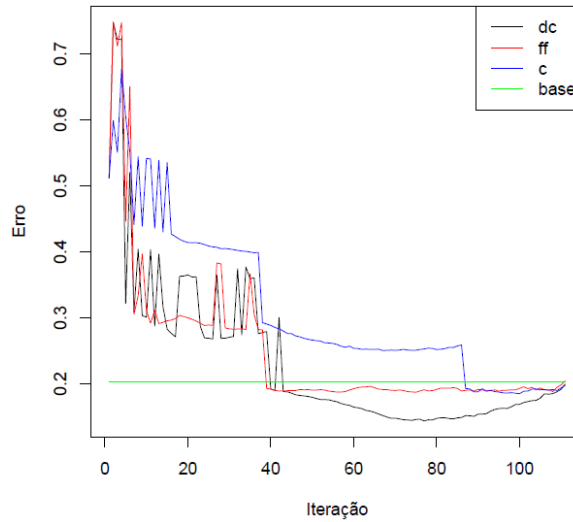


Figura 5.2: *Airline*

A figura 5.2, representa o resultado da técnica de AA do conjunto de dados *Airline Food Reviews*, o eixo X representa os erros de previsão, eixo Y representa as iterações (cada iteração seleciona 10 casos). Utilizando o modelo Tf-Idf, os passos de pré-processamento *TermSparse* (98%), *Stemming* e *Stopwords* e a técnica linguística de *score* e palavras Positivas e Negativas.

Como podemos observar na figura 5.2, o erro tende a baixar com o número de iterações para as três abordagens de AA. Neste conjunto a técnica D-confiança tem melhores resultados. O valor de mínimo de erro atingido é de 0.1449 (D-confiança, iteração 74, ou seja 732 casos). À medida que o conjunto de exemplos etiquetados cresce o erro tende para o da abordagem supervisionada (*baseline*): 0.2036. De qualquer das formas, o erro do *baseline* é atingido com 392 casos (iteração 40) com D-confiança e Distância.

5.3.3 Music

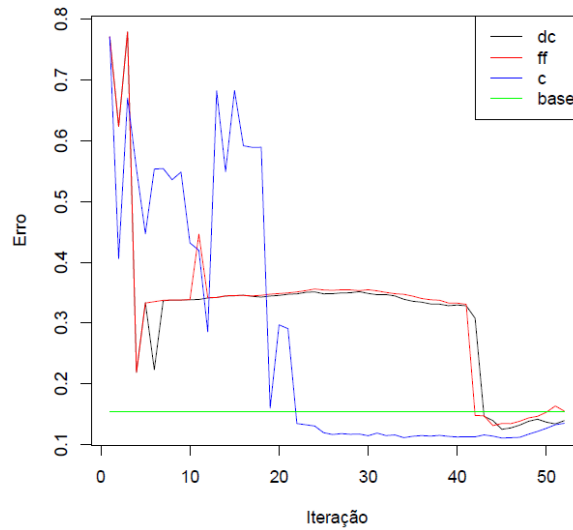


Figura 5.3: *Music*

A figura 5.3, representa o resultado da técnica de AA do conjunto de dados *Music Reviews*, o eixo X representa os erros de previsão, eixo Y representa as iterações (cada iteração seleciona 10 casos). Utilizando o modelo Tf-Idf, os passos de pré-processamento *TermSparse* (97%), *Stemming* e *Stopwords*.

Como podemos observar na figura 5.3, o erro tende a baixar com o número de iterações para as três abordagens de AA, embora com oscilações grandes. Neste conjunto a técnica Confiança tem melhores resultados. O valor de mínimo de erro atingido é de 0.111 (Confiança, iteração 129, ou seja 1282 casos). À medida que o conjunto de exemplos etiquetados cresce o erro tende para o da abordagem supervisionada (baseline): 0.1538. De qualquer das formas, o erro do *baseline* é atingido com 182 casos (iteração 19) com Confiança.

5.3.4 *Teacher*

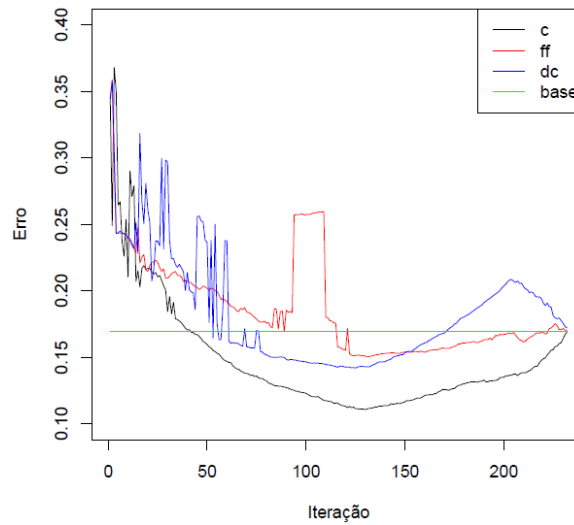


Figura 5.4: *Teacher*

A figura 5.4, representa o resultado da técnica de AA do conjunto de dados *Teacher Reviews*, o eixo X representa os erros de previsão, eixo Y representa as iterações (cada iteração seleciona 10 casos). Utilizando o modelo Tf-Idf, os passos de pré-processamento *Termsparse* (99%), *Stemming* e *Stopwords* e a técnica linguística de *score* e palavras positivas e negativas.

Neste caso também, como podemos observar na figura 5.4, o erro tende a baixar com o número de iterações para as três abordagens de AA, apesar de o fazer de forma pouco suave. Neste conjunto a técnica Confiança tem melhores resultados. O valor de mínimo de erro atingido é de 0.111 (Confiança, iteração 45, ou seja 442 casos). À medida que o conjunto de exemplos etiquetados cresce o erro tende para o da abordagem supervisionada (baseline): 0.1699. De qualquer das formas, o erro do *baseline* é atingido com 422 casos (iteração 43) com Confiança.

5.3.5 *Post-tagati* economia

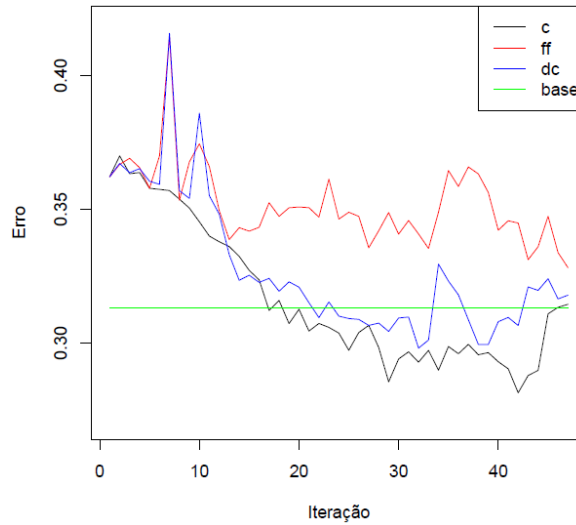


Figura 5.5: *Post-tagati* Economia

A figura 5.5, representa o resultado da técnica de AA do conjunto de dados *Post-tagati* economia, o eixo X representa os erros de previsão, eixo Y representa as iterações (cada iteração seleciona 10 casos). Utilizando o modelo Tf-Idf, os passos de pré-processamento *Termsparse* (98%), *Stemming* e *Stopwords*.

Como podemos observar na figura 5.5, o erro tende a baixar com o número de iterações para as três abordagens de AA. Neste conjunto a técnica Confiança tem melhores resultados. O valor de mínimo de erro atingido é de 0.281 (Confiança, iteração 42, ou seja 412 casos). À medida que o conjunto de exemplos etiquetados cresce o erro tende para o da abordagem supervisionada (baseline): 0.3130. De qualquer das formas, o erro do *baseline* é atingido com 162 casos (iteração 17) com Confiança.

5.3.6 *Post-tagati* tecnologia

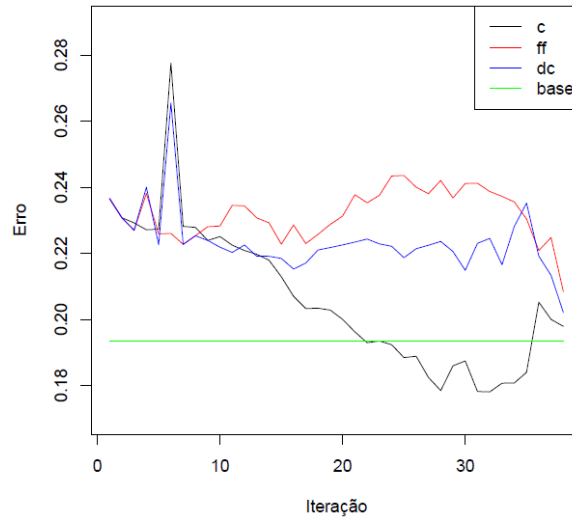


Figura 5.6: *Post-tagati* Tecnologia

A figura 5.6, representa o resultado da técnica de AA do conjunto de dados *Post-tagati* Tecnologia, o eixo X representa os erros de previsão, eixo Y representa as iterações (cada iteração seleciona 10 casos). Utilizando o modelo Tf-Idf, os passos de pré-processamento *Termsparse* (99%), *Stemming* e *Stopwords*.

Como podemos observar na figura 5.6, o erro tende a baixar com o número de iterações para as três abordagens de AA. Neste conjunto a técnica Confiança tem melhores resultados. O valor de mínimo de erro atingido é de 0.178 (Confiança, iteração 28, ou seja 272 casos). À medida que o conjunto de exemplos etiquetados cresce o erro tende para o da abordagem supervisionada (baseline): 0.1935. De qualquer das formas, o erro do *baseline* é atingido com 202 casos (iteração 21) com Confiança.

5.3.7 *Post-tagati environment*

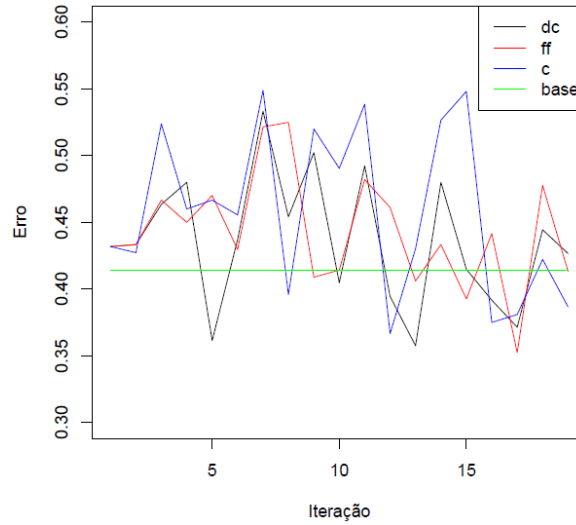


Figura 5.7: *Post-tagati Environment*

A figura 5.7, representa o resultado da técnica de AA do conjunto de dados *Post-tagati Environment*, o eixo X representa os erros de previsão, eixo Y representa as iterações (cada iteração seleciona 3 casos). Utilizando o modelo Binário, os passos de pré-processamento *Termsparse* (97%), *Stemming* e *Stopwords*.

Este conjunto de dados é o conjunto de dados mais pequeno dos outros. Utilizou-se o *query* para cada caso mas não resultou num melhor resultado, portanto optou-se por utilizar *query* de 3 casos.

Como podemos observar na figura 5.7, a técnica D-confiança parece ter melhores resultados. O primeiro mínimo local abaixo do *baseline* é de 0.361 (D-confiança, iteração 5, ou seja 12 casos). À medida que o conjunto de exemplos etiquetados cresce o erro tende para o da abordagem supervisionada (*baseline*): 0.4143. De qualquer das formas, o erro do *baseline* é atingido com 145 casos (iteração 51) com Confiança.

5.3.8 Resumo do resultado de aprendizagem ativa

Na maioria dos problemas, a Confiança produz melhores resultados comparando com a D-confiança e a Distância. Em certos problemas, como se mostra nesta dissertação, a D-confiança tem melhores resultados. Enquanto a Distância não tem melhores resultados para nenhum dos problemas. A confiança tem melhores resultados em problemas: *Movie*, *Music*, *Teacher*, *Post-tagati economia*, *Post-tagati tecnologia*. Enquanto D-confiança tem melhores resultados em problemas: *Airlines* e *Post-tagati environment*.

Os erros de AA ultrapassam os erros de *baseline* (erro de técnica supervisionada), significa que com AA podemos selecionar os melhores casos para conjunto de treino. Algumas experiências (*Movie*, *Airlines*, *Music*, *Teacher*, *Post-tagati* economia e *Post-tagati* tecnologia) sugerem que uma boa seleção de exemplos pode melhorar bastante a capacidade preditiva do modelo baseline. Este resultado pode ser explorado para se obterem melhores classificadores de opinião. Em particular, podemos utilizar critérios de paragem de active learning para decidir qual o melhor sub conjunto de exemplos a utilizar na construção do classificador de opinião [12].

Em conjuntos pequenos como por exemplo conjunto de *Post-tagati environment*, não é fácil observar o efeito da seleção de exemplos feita pela aprendizagem ativa. De qualquer das formas, as técnicas de AA são úteis precisamente em tarefas com um elevado número de exemplos.

Capítulo 6

Conclusão e Trabalho Futuro

6.1 Conclusão

6.1.1 Contribuição de trabalho

A nossa principal contribuição foi a obtenção de resultados experimentais a partir da aplicação de aprendizagem ativa a problemas de extração de opinião. Em particular aplicámos a nova técnica de D-confiança, que já tinha mostrado bons resultados noutro tipo de problemas. Fizemos também um grande número de experiências em extração de opinião combinando a utilização de modelos vetoriais (Binário, Tf e Tf-Idf) com diferentes métodos de pré-processamento e técnicas linguísticas.

6.1.2 Conclusão de trabalho

Neste trabalho, explorámos técnicas de extração de conhecimento de texto e de extração de opinião sobre alguns conjuntos de textos. Também obtivemos resultados experimentais com três técnicas de aprendizagem ativa nesses problemas de extração de opinião.

Em relação à extração de opinião, os resultados variam de um pré-processamento para outro, mas maioritariamente obtém-se melhores resultados com modelo Tf-Idf. Os conjuntos de dados *Movie*, *Airline*, *Music*, *Teacher*, *Post-tagati* economia e *Post-tagati* tecnologia são aqueles que obtêm melhores resultados com Tf-Idf, o único conjunto de dados que obtém um melhor resultado com o Binário é o conjunto de dados *Post-tagati environment*. A utilização de *sparseterm* é muito importante, com esta técnica, reduz-se o número de atributos, o que significa a utilização de menores recursos computacionais. Em certos problemas, ao representar os documentos só com palavras positivas e negativas obtemos melhores resultados do que utilizando todas as palavras. É o caso dos problemas *Airline Food Reviews* e *Teacher*

Reviews. A utilização de extração de sinónimos na base de dados *WordNet* não melhora os resultados na maioria dos casos, embora também não piore.

Entretando, relativamente à aprendizagem ativa (AA), as experiências mostram que, na maioria dos conjuntos de dados podemos reduzir significativamente o número de exemplos necessários para obter um classificador comparável ao que se obtém a partir de todos os exemplos. Os erros de AA ultrapassam os erros de *baseline* (erro de técnica supervisionada), significa que com AA podemos selecionar os melhores casos para conjunto de treino. Algumas experiências (*Movie*, *Airlines*, *Music*, *Teacher*, *Post-tagati* economia e *Post-tagati* tecnologia) sugerem que uma boa seleção de exemplos pode melhorar bastante a capacidade preditiva do modelo baseline. Este resultado pode ser explorado para se obterem melhores classificadores de opinião.

6.2 Trabalho Futuro

- Critério de Paragem.

Nas nossas experiências vimos que a utilização de determinados exemplos pode piorar o erro dos classificadores de opinião. Poderemos então utilizar critérios de paragem de aprendizagem ativa para selecionar subconjuntos dos exemplos etiquetados para que se possa maximizar o desempenho preditivo dos classificadores. No nosso caso, infelizmente, não tivemos oportunidade para investigar este assunto, portanto, colocamo-lo como o trabalho a desenvolver no futuro próximo.

- Protótipo

Colocamos também nos trabalhos futuros, criar um protótipo para classificação de extração de opinião com a técnica de aprendizagem ativa. A ideia é desenhar um sistema que sirva para etiquetar os documentos e classificar os documentos automaticamente.

Apêndice A

Abreviaturas e Acrónimos

5f-cv	5 folds-Cross Validation	N	O numero de todos os documentos
AA	Aprendizagem Ativa	n	Numero de iteração da avaliacao de aprendizagem ativa
bin	Binário	NN	Noun, singular or mass
Bow	Bag of Word	NNS	Noun, plural
C	Classe	PMI	Pointwise Mutual Information
C_i	O conjunto das classes conhecidas no L_i	POS	Part-of-speech
$\text{ClassDist}(u_j; c_k)$	Distância individual entre cada par dos casos	posneg	Positivo e negativo
$\text{Conf}_i(u_j, c_k)$	As confiancas posteriores na classe c_k dado caso u_j	$\text{Pr}(t_i C)$	Probabilidade condicionada do termo t_i na classe C
$d \in D$	Sub documento de Documentos	q_i	Query na iteração i
D-confianca	Distancia-Confiança	RB	Adverb
D	Documentos	RBR	Adverb, comparative
$\text{dist}(u_j; c_k)$	A distância agregada para classe conhecida	RBS	Adverb, superlative
$\text{DocFreq}(w)$	O numero de documentos que contem a palavra w	SO	Orientação sem-Ântica
ePolicy	Engineering the POLicy-making LIfE CYcle	SVM	Support Vector machine
ET	Extração de Texto	$\text{TermFreq}(w, d)$	A frequência de uma palavra do documento
h_i	Modelo de classificador na iteração i	TF-IDF	Term Frequency- inverse document frequency
hits	Numero de acertos	Tf	Term Frequency
$I()$	Função de indicador	t_i	Um termo
i	Iteração	U	Conjunto nao etiquetado
JJ	Adjective	U_i	Conjunto W - Conjunto L_i
K	Numero de Classes	VB	Verb, base form
k	Numero de partição de k-folds Cross Validation	VBD	Verb, past tense
KFF	Kernel Farthest-First	VBG	Verb, gerund or present participle
KNN	K-Nearest-Neighbors	VBN	Verb, past participle
L	Conjunto pequeno	w	Word (palavra)
$L_{0/1}$	Error rate	W	working set
L_i	Conjunto pequeno na iteração i		

Referências

- [1] Engineering the policy-making life cycle, url= <http://www.epolicy-project.eu>. (consultado em julho 2013).
- [2] Wordnet, url=<http://wordnet.princeton.edu/>. (consultado em julho 2013).
- [3] Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1987.
- [4] Yoram Baram, Ran El-Yaniv, and Kobi Luz. Online choice of active learning algorithms. *Journal of Machine Learning Research*, 5:255–291, 2004.
- [5] Núria Bel, Cornelis H. A. Koster, and Marta Villegas. Cross-lingual text categorization. In *ECDL*, pages 126–139, 2003.
- [6] Dustin Boswell. Introduction to support vector machines. 2002.
- [7] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.
- [8] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [9] Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW*, pages 519–528, 2003.
- [10] Mark Dredze and Koby Crammer. Active learning with confidence. In *ACL (Short Papers)*, pages 233–236. The Association for Computer Linguistics, 2008.
- [11] Nuno Filipe Escudeiro and Alípio Mário Jorge. Efficient coverage of case space with active learning. In Luís Seabra Lopes, Nuno Lau, Pedro Mariano, and Luís Mateus Rocha, editors, *EPIA*, volume 5816 of *Lecture Notes in Computer Science*, pages 411–422. Springer, 2009.
- [12] Nuno Filipe Fonseca Vasconcelos Escudeiro. *Semi-automatic classification: using active learning for efficient class coverage*. PhD thesis, Faculdade de Engenharia da Universidade do Porto, 2012.

- [13] Ingo Feinerer, Kurt Hornik, and David Meyer. Text mining infrastructure in r. *Journal of Statistical Software*, 25(5):1–54, 3 2008.
- [14] Ronen Feldman and James Sanger. *The Text Mining Handbook - Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, 2007.
- [15] Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, EACL '97, pages 174–181, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics.
- [16] Vasileios Hatzivassiloglou and Janyce M. Wiebe. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th conference on Computational linguistics - Volume 1*, COLING '00, pages 299–305, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.
- [17] Hochbaum and Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985.
- [18] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In Won Kim, Ron Kohavi, Johannes Gehrke, and William DuMouchel, editors, *KDD*, pages 168–177. ACM, 2004.
- [19] Nitin Jindal and Bing Liu. Mining comparative sentences and relations. In *AAAI*, pages 1331–1336. AAAI Press, 2006.
- [20] Thorsten Joachims. Text categorization with suport vector machines: Learning with many relevant features. In Claire Nedellec and Céline Rouveirol, editors, *ECML*, volume 1398 of *Lecture Notes in Computer Science*, pages 137–142. Springer, 1998.
- [21] Katti Faceli Márcia Oliveira e André Ponce de Leon Carvalho João Gama, Ana Carolina Lorena. *Extração de Conhecimento de Dados Data Mining*. Edições Silabo, 2012.
- [22] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. In *Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pages 3–24, Amsterdam, The Netherlands, The Netherlands, 2007. IOS Press.
- [23] David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *In Proceedings of the Eleventh International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1994.

- [24] Bing Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Data-Centric Systems and Applications. Springer, 2007.
- [25] Bing Liu. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing, Second Edition*. Taylor and Francis Group, Boca, 2010.
- [26] Bing Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012.
- [27] Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: analyzing and comparing opinions on the web. In Allan Ellis and Tatsuya Hagino, editors, *WWW*, pages 342–351. ACM, 2005.
- [28] Matt Flynn Louise Francis. Text mining handbook.
- [29] David Osimo and Francesco Mureddu. Research challenge on opinion mining and sentiment analysis.
- [30] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86, Philadelphia, PA, USA, July 2002. Association for Computational Linguistics.
- [31] Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *HLT/EMNLP*. The Association for Computational Linguistics, 2005.
- [32] Fabrizio Sebastiani. Machine learning in automated text categorization. *CoRR*, cs.IR/0110053, 2001.
- [33] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.
- [34] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2001.
- [35] Luis Torgo. *Data Mining I - Classification Models*. Faculty of Sciences, University of Porto, October 2011.
- [36] Peter D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *ACL*, pages 417–424. ACL, 2002.
- [37] Andreas Vlachos. Active learning with support vector machines. Master’s thesis, University of Edinburgh, 2004.

- [38] Silvia Vázquez and Núria Bel. A classification of adjectives for polarity lexicons enhancement. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet U?ur Do?an, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA).
- [39] Chih wei Hsu, Chih chung Chang, and Chih jen Lin. A practical guide to support vector classification, 2010.