Utilização de Informação Linguística na classificação de documentos em Língua Portuguesa

Teresa Cristina de Freitas Gonçalves

Orientador: Professor Doutor Paulo Miguel Torres Duarte Quaresma

Tese submetida à Universidade de Évora
para obtenção do grau de Doutor em Informática

Departamento de Informática
Universidade de Évora

Novembro de 2007

Utilização de Informação Linguística na classificação de documentos em Língua Portuguesa

Teresa Cristina de Freitas Gonçalves

Orientador: Professor Doutor Paulo Miguel Torres Duarte Quaresma

Departamento de Informática
Universidade de Évora

Novembro de 2007

Resumo

As Tecnologias de Informação actuais e os serviços baseados na Web necessitam de gerir, seleccionar e filtrar quantidades crescentes de informação textual. A classificação de textos permite aos utilizadores, através da navegação sobre hierarquias de classes, consultar mais facilmente o conjunto de textos do seu interesse. Este paradigma é muito eficaz tanto na filtragem de informação como no desenvolvimento de serviços *online* dirigidos para o utilizador.

Como a quantidade de documentos envolvidos nestas aplicações é grande, são necessárias aproximações eficientes e automáticas de classificação. Os modelos padrão de Aprendizagem Automática utilizam a representação saco-de-palavras para induzir a função objectivo de classificação, onde as únicas características do documento são estatísticas sobre as suas palavras. As estruturas da linguagem típicas, como a morfologia, a sintaxe e a semântica são completamente ignoradas no processo de aprendizagem. Por outro lado, a quase totalidade de estudos tem sido realizada sobre textos escritos na língua Inglesa.

Esta tese examina o papel das diversas estruturas linguísticas na classificação de textos, aplicando o estudo à língua Portuguesa. Define-se uma arquitectura modular para a tarefa de classificação de documentos que permite seleccionar o nível de informação linguística utilizado e propõe-se uma representação e um algoritmo de aprendizagem adequados para a informação semântica.

A experimentação mostra que utilizando informação morfológica os resultados podem ser superiores aos da representação padrão e que as estruturas semânticas utilizadas possuem um poder discriminante sobre as classes equivalente ao das estruturas morfológicas.

Abstract

Current Information Technologies and Web-based services need to manage, select and filter increasing amounts of textual information. Text classification allows users, through navigation on class hierarchies, to browse more easily the texts of their interests. This paradigm is very effective both in filtering information as in the development of online end-user services.

As the number of documents involved in these applications is large, efficient and automatic approaches are necessary for classification. Standard Machine Learning approaches use the bag-of-words representation to deceive the classification target function, where the only features are document word statistics. Typical linguistic structures such as morphology, sintax and semantic are completely ignored in the learning process. Moreover, almost all studies have been conducted on texts written in the English language.

This thesis examines the role of various linguistic structures in text classification, applying the study to the Portuguese language. It defines a modular architecture to the classification task that can select the linguistic information level used and proposes a representation and a learning algorithm that uses semantic information.

Experiments show that using morphological information the results can be better than using a standard representation and that the semantic structures used have a discriminating power over categories equivalent to the morphological structures.

Aos meus filhos, Tiago, Carolina e Sofia

Agradecimentos

Gostaria de agradecer profundamente ao meu orientador, Prof. Paulo Quaresma, pelo seu apoio e incentivo. Agradeço a sua disponibilidade e a sua preciosa ajuda, a motivação nos momentos mais difíceis e o apoio que sempre manifestou.

Um agradecimento especial ao Fajé, amigo inesquecível, por me ter dado a conhecer a área de Aprendizagem Automática.

Gostaria também de agradecer aos colegas do Departamento de Informática, em especial à Prof. Irene, sempre boa conselheira, e ao Saias, sempre disposto a resolver os pequenos problemas relativos aos servidores e outras máquinas.

Agradeço, ainda, ao Departamento de Informática da Universidade de Évora pelo apoio dado durante estes anos.

Finalmente, gostaria de agradecer aos meus pais, Loyola e João, sempre disponíveis para ajudar em tudo; aos meus filhos Tiago, Carolina e Sofia pela sua alegria e compreensão; ao Pedro pelo seu amor e paciência.

Índice

Resumo **i**

Abstract **iii**

1 Introdução **1**

1.1 Motivação 2

1.2 Objectivos 2

1.3 Abordagem proposta 3

1.4 Principais contribuições 3

1.5 Organização da dissertação 4

2 Informação linguística **7**

2.1 Informação Morfológica 8

2.1.1 Palavra e morfema 8

2.1.2 Classes gramaticais das palavras 8

2.2 Informação Sintáctica 9

2.2.1 Frase, oração e período 9

2.2.2 Termos da oração 9

2.2.3 Constituintes sintácticos da oração 10

2.3 Informação Semântica 10

2.4 Representação da Informação Linguística 11

2.4.1 Representação morfológica da palavra 12

2.4.2 Representação sintáctica da frase 12

2.4.3 Representação semântica 13

2.5 Ferramentas utilizadas 15

2.5.1 POLARIS 15

2.5.2 PALAVRAS 15

2.5.3 SIN2SEM 17

3 Classificação automática de textos **19**

3.1 Definição 20

3.1.1 Problemas binários e não binários 21

3.2 Aplicações de classificação de texto 22

3.2.1 Indexação de documentos 22

3.2.2 Organização de documentos 23

3.2.3 Filtragem de texto 23

3.2.4 Desambiguação do significado das palavras 24

3.2.5 Classificação hierárquica de páginas web 24

3.3 A aproximação da Aprendizagem Automática 25

3.3.1 Conjuntos de treino, teste e validação 25

3.4 Construção do classificador 27

3.4.1 Representação do documento 27

3.4.2 Indução do classificador 30

3.4.3 Avaliação do classificador 32

3.5 Ferramentas utilizadas 35

3.5.1 WEKA 35

3.5.2 SVMlight 36

3.5.3 SVMlight-TK 36

4 Técnicas de CT **37**

4.1 Unidades básicas de informação 37

4.1.1 Nível sub-palavra 38

4.1.2 Nível palavra 38

4.1.3 Nível multi-palavra 39

4.1.4 Nível semântico 40

4.2 Algoritmos 41

4.2.1 Classificadores probabilísticos 41

4.2.2 Árvores de decisão 43

4.2.3 Regras de decisão 45

4.2.4 Métodos *online* 46

4.2.5 Redes neuronais 47

4.2.6 Método de Rocchio 48

4.2.7 Classificadores baseados em exemplos 49

4.2.8 Máquinas de vectores de suporte 51

4.2.9 Comités de classificadores 51
4.2.10 Outros métodos 53
4.3 Desempenho dos classificadores 53
4.3.1 Unidades de indexação 54
4.3.2 Algoritmos 55
4.3.3 Testes de significância estatística 55
4.4 Classificação de textos em língua Portuguesa 56

5 Núcleos para CT **59**
5.1 Métodos de núcleo 60
5.2 Funções de núcleo e suas propriedades 62
5.2.1 Função de núcleo 62
5.2.2 A matriz de Gram 63
5.2.3 O núcleo e o espaço de características induzido 64
5.2.4 Medidas básicas no espaço definido por um núcleo 67
5.2.5 Operações sobre núcleos 69
5.2.6 Exemplos de núcleos 70
5.3 Máquinas de vectores de suporte 71
5.3.1 Teoria da aprendizagem estatística 71
5.3.2 Classificadores por hiperplanos 74
5.3.3 Classificação com vectores de suporte 77
5.4 Núcleos para texto 79
5.4.1 Núcleos para o espaço de vectores 79
5.4.2 Núcleos para objectos estruturados 81

6 Núcleo para informação semântica **89**
6.1 Representação da informação semântica 90
6.1.1 Estrutura de Representação do Discurso 91
6.2 A forma lógica e os métodos de núcleo 91
6.2.1 A ERD como um grafo 92
6.2.2 Substituição de referentes 93
6.2.3 A Representação de uma ERD 96
6.2.4 A semelhança entre ERDs 96
6.2.5 Representação da forma lógica de um documento 98

7 Avaliação **101**

7.1 Representação do documento 102
7.1.1 Informação morfológica 102
7.1.2 Informação morfo-sintáctica 103
7.1.3 Informação sintáctica 104
7.1.4 Informação semântica 106
7.2 Corpora 109
7.2.1 PGR 109
7.2.2 Folha de S. Paulo 111
7.2.3 Público 111
7.3 Resultados 112
7.3.1 Informação morfológica 113
7.3.2 Informação morfo-sintáctica 118
7.3.3 Informação sintáctica 121
7.3.4 Informação semântica 123
7.4 Avaliação comparativa 126
7.4.1 Diferenças entre representações do mesmo nível 126
7.4.2 Diferenças entre níveis de informação linguística 128
7.4.3 Diferenças entre classes do mesmo corpus 129
7.5 Trabalho relacionado 132

8 Conclusões e Trabalho Futuro **133**
8.1 Conclusões 134
8.2 Trabalho futuro 136

Anexos **139**

A Morfemas e Palavras **141**
A.1 Morfemas gramaticais 141
A.2 Classes gramaticais 142

B Períodos e orações **151**
B.1 Tipos de frase 151
B.2 Classificação de períodos compostos 151
B.3 Funções sintácticas dos elementos da oração 152
B.4 Constituintes sintácticos da oração 154

C Redução de dimensão **157**

C.1 Selecção de termos 157

C.2 Extracção de termos 159

C.3 Ponderação de características 161

D Espaços vectoriais e de Hilbert **163**

D.1 Espaços vectoriais 163

D.2 Normas e produtos internos 165

D.3 Espaços de Hilbert 168

E Etiquetas do PALAVRAS **171**

Referências **177**

Lista de Tabelas

3.1 A tabela de contingência para a classe c_i. 32

7.1 Corpus PGR: as 10 classes mais comuns e o seu número de documentos. 110

7.2 Corpora Publico e Publico9510: número de documentos por classe e a respectiva proporção relativa ao corpus completo. 112

7.3 Número de palavras (#) e médias por documento de símbolos (μ_s) e tipos (μ_t) antes e depois da eliminação de *stopwords* e lematização (corpora PGR, Folha e Publico). 113

7.4 Número de termos obtidos para diferentes valores de limiar (corpora PGR, Folha e Publico). 114

7.5 Valores mínimo, máximo, médio e desvio padrão das micro- e macro-médias (corpora PGR, Folha e Publico). 114

7.6 Corpus PGR: número de experiências que não apresentam diferenças significativas com o melhor resultado. 115

7.7 Corpus PGR: valores das medidas de desempenho para as melhores experiências (no que respeita à cobertura e f_1). 116

7.8 Corpus Folha: valores das medidas de desempenho para algumas das melhores experiências. 116

7.9 Corpus Publico: valores das medidas de desempenho para as melhores experiências. 117

7.10 Valores das medidas de desempenho para as experiências com informação morfológica (corpora Folha2 e Publico9510). 118

7.11 Número de palavras (#) e médias por documento de símbolos (μ_s) e tipos (μ_t) para cada classe gramatical (corpora PGR, Folha e Publico). 118

7.12 Corpus PGR: valores das medidas de desempenho para as experiências com conjuntos de classes gramaticais. 119

7.13 Corpus PGR: valores das medidas de desempenho das melhores experiências com conjuntos de classes gramaticais para diferentes valores de limiar. 120

7.14 Corpus Folha: valores das medidas de desempenho para as experiências com conjuntos de classes gramaticais. 120

7.15 Corpus Folha: valores das medidas de desempenho das melhores experiências com conjuntos de classes gramaticais para diferentes valores de limiar. 121

7.16 Corpus Publico: valores das medidas de desempenho para as experiências com conjuntos de classes gramaticais. 122

7.17 Corpus Folha2: Valores das medidas de desempenho para diversas representações utilizando informação sintáctica. 123

7.18 Corpus Publico9510: Valores das medidas de desempenho para diversas representações utilizando informação sintáctica. 124

7.19 Corpus Folha2: Valores das medidas de desempenho para diversas representações utilizando informação semântica. 125

7.20 Corpus Publico9510: Valores das medidas de desempenho para diversas representações utilizando informação semântica. 125

7.21 Valores das medidas de desempenho para as experiências escolhidas em cada nível de informação linguística. 129

7.22 Corpora PGR, Folha e Publico: valores das medidas de desempenho individuais para o nível de informação morfo-sintáctico. 130

7.23 Corpora Folha2 e Publico9510: valores das medidas de desempenho individuais para o nível de informação semântica. 130

C.1 Principais funções utilizadas na selecção de termos. 159

C.2 Componentes de ponderação de termos mais utilizadas. 162

E.1 PALAVRAS – As etiquetas das classes gramaticais 172

E.2 PALAVRAS – As etiquetas das flexões 172

E.3 PALAVRAS – As etiquetas dos constituintes sintácticos 173

E.4 PALAVRAS – As etiquetas das funções sintácticas 173

E.5 PALAVRAS – As etiquetas dos tipos de frase 173

E.6 PALAVRAS – As etiquetas dos tipos de orações 173

Lista de Figuras

2.1 A árvore sintáctica da frase `A mãe de binóculos observa a filha`. 13

2.2 Duas árvores sintácticas possíveis para a frase `A mãe observa a filha de binóculos`. 14

2.3 Quatro representações semânticas distintas da frase `A mãe observa a filha`. (a) ilustra a frase em Lógica de Primeira Ordem, (b) representa uma Rede Semântica[19], (c) é um diagrama de Dependência Conceptual[20] e (d) uma representação Baseada em Quadros[21]. 14

3.1 A aproximação da AA na classificação de textos: os documentos d_j previamente classificados são transformados numa representação compacta do seu conteúdo que, por sua vez, é apresentada ao aprendiz para induzir um modelo para a classe c_i. Este é posteriormente utilizado para classificar novos documentos d_x. 25

4.1 Uma árvore de decisão; os nós representam atributos, os ramos, os valores possíveis, e as folhas, a classe. 44

4.2 O classificador de Rocchio quando os exemplos positivos ocorrem em grupos separados. Os círculos representam os exemplos positivos e as cruzes os negativos. O círculo maior representa a área de influência do classificador. 49

4.3 A indução de classificadores de vectores de suporte. 52

5.1 Através da função Φ, os exemplos são mapeados para um espaço de características onde as relações não lineares se tornam lineares. A função de núcleo permite encontrar essas relações directamente a partir dos exemplos. 61

5.2 Os dados são processados utilizando uma função de núcleo para criar uma matriz que, por sua vez, é processada por um algoritmo de aprendizagem que produz uma função que reconhece padrões. Esta função é utilizada para classificar novos exemplos. 61

5.3 Exemplo de classificação binária resolvida utilizando três modelos. Os modelos variam em complexidade desde o mais simples (esquerda) que classifica erradamente um grande número de pontos, até ao mais complexo cuja solução é consistente com todos os pontos de treino (mas pode não funcionar tão bem com novos pontos). 72

5.4 Exemplo de uma dimensão VC simples. Existem $2^3 = 8$ formas de associar 3 pontos a duas classes; para pontos representados em $\mathbb{R}^2$, todas as 8 possibilidades podem ser obtidas utilizando hiperplanos separadores, ou seja, esta classe de funções consegue separar os três pontos. Isto não resultaria se existissem 4 pontos, independentemente da sua localização. Assim, a dimensão VC da classe dos hiperplanos separadores em $\mathbb{R}^2$ é 3. 73

5.5 Um exemplo de classificação binária. O hiperplano óptimo está representado por uma linha sólida. Se o problema for separável, existe um vector peso $\mathbf{w}$ e um limiar b para o qual $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0$ $(i = 0, \ldots, m)$. Alterando a escala de tal forma que os pontos mais próximos do hiperplano satisfaçam $|\langle \mathbf{w}, \mathbf{x}_i \rangle + b| = 1$ obtém-se a forma canónica do hiperplano $(\mathbf{w}, b)$, que satisfaz $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$. 76

6.1 Representação gráfica da ERD correspondente à frase `A mãe observa a filha que brinca com a boneca`. 91

6.2 Grafo que ilustra a ERD da frase `A mãe observa a filha que brinca com a boneca`. 92

6.3 Representação gráfica da ERDs correspondente às frases `A mãe observa a filha Carlota` e `A filha Carlota observa a formiga` antes e depois da substituição dos referentes envolvidos em condições do tipo `nome(`x`,y)`. . . . 94

6.4 Representação gráfica da ERD correspondente à frase `A mãe observa a filha que brinca com a boneca` antes e depois da substituição dos referentes envolvidos em condições atómicas do tipo `cond(`x`)`. 95

6.5 Representação gráfica da ERD correspondente à frase `A mãe observa a filha Carlota que brinca com a boneca` antes e depois da substituição dos referentes restringidos por nomes próprios e condições atómicas. 95

6.6 Grafo que ilustra a ERD com substituição de referentes da frase `A mãe observa a filha que brinca com a boneca`. 96

6.7 Grafos que ilustram as ERDs das frases `A mãe observa a filha Carlota` e `A filha Carlota observa a formiga` e o grafo produto directo correspondente. 97

6.8 Matriz de adjacência do grafo de produto representado na Figura 6.7. 97

6.9 Grafo que ilustra a representação das frases `A mãe observa a filha Carlota` e `A filha Carlota observa a formiga` no mesmo documento. 99

7.1 Ferramentas de extracção de informação linguística: exemplo de utilização. . 101

7.2 O documento original e a respectiva representação saco-de-palavras. 103

7.3 Através de espaços a três dimensões é possível visualizar graficamente o conjunto de experiências realizadas com informação morfológica. O ponto marcado corresponde à experiência: eliminação de *stopwords* com a raíz da palavra como termo de indexação, aplicação da função de graduação `imut` e valor de limiar igual a 1 e `tfidf` como medida de ponderação. 104

7.4 O documento original, as árvores sintácticas geradas pela ferramenta PALAVRAS e a respectiva representação árvore-sintáctica. 105

7.5 O documento original e a respectiva representação sequência-de-palavras. . . 106

7.6 A representação estrutura-do-discurso com as frases `A mãe observa a filha Carlota` e `A filha Carlota observa a formiga` no mesmo documento. . . 108

7.7 O documento original, as ERDs geradas e a respectiva representação estrutura-do-discurso. Esta representação do documento utiliza a substituição de nomes próprios e condições atómicas. 108

7.8 Corpus `PGR`: histograma do número de documentos para as 50 classes mais comuns. 110

Capítulo 1

Introdução

Com o rápido crescimento da World Wide Web, a tarefa de classificar documentos em língua natural num conjunto pré-definido de classes semânticas[1] tornou-se um dos métodos chave para organizar informação *online*. Esta tarefa, usualmente designada por classificação de texto, constitui uma pedra basilar numa vasta área de aplicações. Por exemplo, as directorias do Yahoo! classificam páginas Web por tópico, os jornais *online* adaptam-se às preferências de leitura de um utilizador particular e os agentes de encaminhamento de linhas automáticas direccionam as mensagens electrónicas, a partir do seu conteúdo, para o perito apropriado.

No passado, a classificação era realizada manualmente por humanos, mas o crescimento exponencial do número de documentos *online* e a rapidez com que a informação deve ser distribuída criou a necessidade da classificação automática.

As primeiras aproximações à classificação automática de documentos utilizavam técnicas de engenharia do conhecimento[2], onde as regras para classificar automaticamente novos documentos são definidas manualmente por um conjunto de peritos. No entanto, este processo é difícil e moroso e como tal torna-se ineficiente e impraticável em muitas situações. Estas dificuldades são ainda maiores nos casos em que o número de classes é grande ou a sua definição varia ao longo do tempo.

Uma aproximação de Aprendizagem Automática[3] para a construção das regras de classificação pode solucionar estes problemas. Dado um conjunto relativamente pequeno de documentos classificados manualmente, o processo de construção das regras pode ser visto como um problema de aprendizagem supervisionada: o algoritmo recebe

[1] Alguns autores designam a classe semântica por categoria e a tarefa de classificar documentos em classes semânticas por categorização de documentos. Neste trabalho utilizam-se os termos classe e classificação, mas assume-se que a classe incorpora essa informação semântica.

[2] Do inglês, *knowledge engineering*.

[3] Do inglês, *Machine Learning*.

o conjunto de documentos etiquetados e produz, automaticamente, um conjunto de regras.

1.1 Motivação

A escolha do tema deste estudo e da metodologia adoptada prende-se com estas questões. Por um lado, a necessidade de processamento de grandes quantidades de informação conduziu à adopção de técnicas de Aprendizagem Automática para classificar os documentos; por outro lado, a (quase) inexistência de análises ajustadas à língua Portuguesa transportou-o para esta aplicação específica.

Além disso, pretendeu-se analisar a existência de ganhos de desempenho com a utilização de informação linguística na tarefa de classificação de documentos, tal como Lewis and Jones [1997] o fizeram para tarefa de Recuperação de Informação[4].

1.2 Objectivos

Os algoritmos de aprendizagem não interpretam directamente os documentos digitais, sendo necessário transformar cada texto numa representação compacta do seu conteúdo. A mais comum, designada por saco-de-palavras, utiliza uma representação estatística do documento contabilizando, de alguma forma, as palavras que nele aparecem. As estruturas linguísticas (como, por exemplo, a sintaxe e a semântica) típicas dos documentos de língua natural são completamente negligenciadas. Mais ainda, a maioria dos trabalhos de investigação em classificação automática de texto utilizam documentos escritos em Inglês, uma língua anglo-saxónica.

Este trabalho procura atingir os seguintes objectivos:

- por um lado, fazer um estudo sobre a língua Portuguesa, procurando identificar e analisar as suas especificidades e verificar se os resultados obtidos com as representações saco-de-palavras na língua Inglesa são igualmente válidos para uma língua românica;
- por outro, examinar os algoritmos de Aprendizagem Automática e adoptar um que consiga, de alguma forma, considerar, no seu processo de indução, as diversas estruturas linguísticas;
- com base nas análises efectuadas, propôr uma representação dos documentos que

[4]Do inglês, *Information Retrieval*

contenha a sua informação semântica e um algoritmo de classificação automática que a considere;

- finalmente, pesquisar, avaliar e comparar qual a contribuição das estruturas linguísticas, nomeadamente da morfologia, da sintaxe e da semântica, no processo de classificação automática de documentos na língua Portuguesa.

1.3 Abordagem proposta

Para atingir os objectivos escolheram-se corpora escritos na língua Portuguesa com diferentes características: uma colecção pertence a uma área de conhecimento específica, mais concretamente a área jurídica – o corpus `PGR` e a outra possui um carácter mais universal já que contém as notícias de um jornal generalista – o corpus `Publico`. Considerou-se ainda um terceiro corpus, também com notícias de um jornal generalista mas, ao contrário dos primeiros cujos documentos foram redigidos em Portugal, neste os documentos retratam a língua Portuguesa do Brasil – a colecção `Folha`.

Com estes corpora efectuou-se um conjunto de experiências iniciais para avaliar, na língua Portuguesa, as técnicas de representação do documento usualmente utilizadas em Recuperação de Informação. O melhor desempenho obtido nesta representação serviu de ponto de partida para as restantes.

Depois, apreciaram-se as representações linguísticas para os níveis de informação sintáctica e semântica, por um lado, e examinaram-se os algoritmos de Aprendizagem Automática, por outro. Utilizou-se a árvore sintáctica para representar a informação sintáctica e as Estruturas de Representação do Discurso da Teoria da Representação do Discurso para representar a informação semântica. Como algoritmo de classificação escolheu-se aquele que se considerou ser o mais apropriado para utilizar os vários níveis informação linguística – a máquina de vectores de suporte.

Como este algoritmo genérico não suporta directamente a representação semântica, definiu-se e propôs-se uma nova representação dos documentos que contivesse essa informação e que permitisse à máquina de vectores de suporte considerá-la no processo de indução dos classificadores.

1.4 Principais contribuições

Este trabalho elabora um estudo comparativo detalhado sobre a influência de diferentes níveis de informação linguística, nomeadamente o morfológico, o sintáctico e o semântico, no processo de classificação de textos:

- define uma arquitectura modular para a tarefa de classificação de documentos que permite seleccionar o nível de informação linguística a utilizar,
- propõe e avalia uma representação e um algoritmo de aprendizagem automática adequados para a informação semântica, e
- caracteriza as melhores opções de representação do documento em cada e entre os vários níveis de informação linguística considerados.

Este estudo foi direccionado para a língua Portuguesa.

Quanto ao algoritmo proposto, a máquina de vectores de suporte mostrou ser um algoritmo adequado para a classificação de textos em língua Portuguesa, quer aos níveis da eficiência computacional e desempenho alcançados (como referido em outros estudos de classificação de textos), quer na utilização de diferentes representações do documento, nomeadamente no espaço de vectores e estruturadas.

Relativamente à informação morfológica, é possível constatar que, à imagem dos estudos feitos para a língua Inglesa, a representação do documento que produz o melhor desempenho depende do corpus em estudo. No entanto, identificaram-se certas características específicas da língua Portuguesa: a lematização é benéfica (pelo menos para valores de limiar mais baixos) e a etiquetagem morfo-sintáctica pode ser utilizada na redução do número de atributos, sem diminuir o desempenho do algoritmo.

Por outro lado, e de acordo as expectativas iniciais, a utilização da informação sintáctica não revela bons resultados. De facto esta informação, que considera a estrutura sintáctica da frase seria mais adequada, não para classificar os documentos em classes, mas, possivelmente, para identificar os seus autores caso utilizassem diferentes construções sintácticas.

No entanto, a informação semântica com uma representação estruturada do documento mostra possuir um poder discriminante sobre as classes equivalente ao da informação morfológica com uma representação saco-de-palavras.

Esta última constatação é importante já que permite afirmar que se obtêm valores de desempenho equivalentes utilizando diferentes aproximações na representação dos documentos, nomeadamente uma representação estatística das palavras e uma representação que utiliza informação linguística.

1.5 Organização da dissertação

Esta dissertação começa por introduzir as duas áreas de influência do trabalho: estabelece as noções básicas sobre informação linguística e apresenta a área da classificação

automática de textos:

- no Capítulo 2 apresentam-se os vários níveis de informação linguística, referindo as suas unidades básicas de informação, seguido das correspondentes representações linguísticas e das ferramentas de análise linguística utilizadas;
- no Capítulo 3 define-se a tarefa da classificação de texto e referem-se as suas principais aplicações apresentando, de seguida, a aproximação da Aprendizagem Automática e o processo de construção de um classificador. No final, descrevem-se sucintamente as ferramentas utilizadas.

A segunda parte descreve as técnicas "estado da arte" para a construção de classificadores de texto:

- o Capítulo 4 descreve as aproximações que têm sido propostas quer a nível das unidades básicas de informação, quer a nível dos algoritmos de indução de classificadores. Apresenta ainda algumas considerações sobre o desempenho comparativo dos algoritmos e enumera alguns trabalhos de classificação aplicados a textos escritos na língua Portuguesa;
- o Capítulo 5 apresenta os métodos de núcleo, nos quais se inclui o algoritmo de aprendizagem utilizado – a máquina de vectores de suporte. Descreve a função de núcleo e suas propriedades teóricas, introduz o algoritmo e apresenta diversas funções de núcleo que podem ser aplicadas ao texto.

Finalmente, apresenta a abordagem utilizada para incluir a informação semântica na representação dos documentos:

- o Capítulo 6 propõe uma representação do documento que, por um lado, inclui a sua informação semântica e, por outro, é adequada à utilização de métodos de núcleo. Descreve, sucintamente, as Estruturas de Representação do Discurso que reproduzem a forma lógica e formaliza uma representação do documento que, incluindo essa informação, permite a sua utilização pelos métodos de núcleo;
- o Capítulo 7 descreve as representações do documento utilizadas em cada nível de informação linguística, nomeadamente o morfológico, morfo-sintáctico, sintáctico e semântico, apresenta os corpora utilizados, detalha os resultados obtidos e faz uma avaliação comparativa dos mesmos;
- o Capítulo 8 apresenta as conclusões: discute os resultados obtidos e apresenta trabalho futuro com o objectivo de melhorar o desempenho dos classificadores.

Capítulo 2

Informação linguística

A linguagem escrita obedece a uma estrutura específica e compreende vários níveis de informação. É possível identificar pelo menos cinco níveis distintos [Allen 1994; Jurafsky and Martin 2000]:

- **Morfologia.** A morfologia estuda a forma como as palavras são construídas a partir das unidades básicas de significado – os morfemas.
- **Sintaxe.** A sintaxe estuda a forma como as palavras podem ser ordenadas e agrupadas para formar frases correctas e determina qual o papel que cada palavra tem na frase.
- **Semântica.** A semântica preocupa-se com o significado das palavras e como esses significados se conjugam para formar o sentido do texto.
- **Pragmática.** A pragmática estuda a forma como as frases são utilizadas em diferentes situações e como a sua utilização afecta a interpretação da frase.
- **Discurso.** Este nível estuda unidades linguísticas maiores que uma única expressão; preocupa-se com a forma como as frases anteriores afectam a interpretação da frase seguinte. Esta informação é especialmente importante para clarificar pronomes e aspectos temporais da informação expressa.

Estas definições são um pouco difusas já que um facto particular pode incluir aspectos de vários níveis e uma estrutura de representação pode abranger vários níveis simultaneamente.

Este capítulo apresenta a informação linguística contida nos níveis morfológico, sintáctico e semântico apontando, depois, formas de a representar. No final descreve, de forma sumária, as ferramentas de extracção de informação linguística utilizadas.

2.1 Informação Morfológica

A morfologia consiste no estudo da palavra e da sua função na língua. Examina a sua formação isoladamente sem considerar a sua participação na frase.

2.1.1 Palavra e morfema

A **palavra** é uma unidade básica da linguagem falada e escrita. No entanto, a sua unidade mínima de significado é o **morfema**. Por exemplo, a palavra `felizmente` resulta da combinação de dois elementos menores: `feliz` e `-mente`. Cada um deles é um morfema e pode ser utilizado para construir outras palavras: com o morfema `feliz` é possível construir a palavra `felicidade` e o morfema `-mente` desempenha a função gramatical de transformar um adjectivo num advérbio.

Quanto à formação de palavras, podemos dizer que `feliz` é um **morfema livre** pois pode aparecer sozinho para formar uma palavra. Por outro lado, o morfema `-mente` é um **morfema preso** pois nunca pode aparecer sozinho; precisa sempre de ligar-se pelo menos a outro morfema para formar uma palavra.

Tendo em conta o seu significado, e como o exemplo sugere, existem dois tipos de morfemas: o morfema lexical[1] e o morfema gramatical[2]. Intuitivamente, o **morfema lexical** é o morfema "principal" da palavra pois é ele que lhe dá significado (é ele que irmana as palavras da mesma família e lhes transmite uma base comum de significação), enquanto o **morfema gramatical** deriva das relações e categorias existentes na língua.

2.1.2 Classes gramaticais das palavras

As palavras de uma língua podem, segundo a classe gramatical [Cunha and Cintra 1984] a que pertencem, ser: nomes[3], adjectivos, artigos, pronomes, numerais, verbos, advérbios, conjunções, preposições e interjeições.

Para algumas classes, existem ainda subclasses e, consoante a classe a que pertence, a palavra pode apresentar diferentes flexões[4]. Na língua Portuguesa, estas podem ser de: número, género, pessoa, grau, modo, tempo e voz.

É possível relacionar os tipos de morfemas com as classes gramaticais das palavras. São morfemas lexicais os nomes, adjectivos, verbos e advérbios de modo. São morfemas

[1] Também chamado lexema ou semantema e vulgarmente conhecido por radical.
[2] Também conhecido por graema ou formante
[3] Também denominados como substantivos.
[4] Também conhecidas por variações.

gramaticais os artigos, os pronomes, os numerais, as preposições, as conjunções e os demais advérbios, bem como as formas indicadoras de número, género, tempo e modo.

O Anexo A apresenta a classificação dos morfemas gramaticais e descreve, sucintamente, cada uma das classes gramaticais das palavras, indicando as subclasses e flexões permitidas para a Língua Portuguesa.

2.2 Informação Sintáctica

A sintaxe é a parte da gramática que descreve as regras segundo as quais as palavras se combinam para formar frases.

2.2.1 Frase, oração e período

A **frase** é a unidade mínima de comunicação. Pode ser constituída por uma só ou por várias palavras, entre as quais poderá, ou não, existir um verbo. Uma **oração** pode ser definida como um grupo de palavras relacionadas.

Uma frase pode conter uma ou mais orações. Contém apenas uma oração quando apresenta uma só forma verbal (clara ou oculta) ou duas ou mais formas verbais integrantes de uma só locução verbal[5]; contém mais que uma oração, quando há nela mais que um verbo ou locução verbal (claro ou oculto).

O **período** é a frase organizada em uma ou mais orações. No primeiro caso, o período é **simples**; no segundo, é **composto**[6]. O período termina sempre por uma pausa bem definida, que se marca na escrita com ponto, ponto de exclamação, ponto de interrogação, reticências e, algumas vezes, com dois pontos. Esta pontuação esclarece qual a intenção do emissor da frase, determinando qual o seu tipo: declarativa, interrogativa, exclamativa ou imperativa.

2.2.2 Termos da oração

Uma oração é constituída por dois termos essenciais: o sujeito e o predicado. O **sujeito** é o ser sobre o qual se faz uma declaração; o **predicado** é tudo aquilo que se diz do sujeito.

Existem diversos tipos de sujeito: simples ou composto, oculto ou ainda indeterminado.

[5] A locução verbal é o conjunto formado por um verbo auxiliar e um principal.
[6] Também chamado período complexo.

O predicado pode ser nominal, verbal ou verbo-nominal; é constituído pelo verbo e por todos os outros elementos funcionais da oração: complementos, modificadores ou predicativo do sujeito.

Enquanto os **complementos** são elementos obrigatórios quando exigidos pelo verbo, os **modificadores** são facultativos. Os complementos podem ser: directo, indirecto, preposicional e adverbial e os modificadores podem ser preposicional ou adverbial.

O vocativo é outro termo da oração que, sendo isolado do resto da frase e não subordinado a nenhum outro termo, pode relacionar-se com algum deles.

2.2.3 Constituintes sintácticos da oração

Por outro lado, uma oração pode ser dividida em constituintes sintácticos – **os sintagmas**[7]. Estes são formados por grupos de palavras onde, pelo menos uma delas, constitui o seu núcleo[8].

O **núcleo** é a palavra, ou conjunto de palavras, que determina as propriedades sintácticas e combinatórias do sintagma a que pertence.

Todo o constituinte que tem por núcleo um nome, recebe o nome de **sintagma nominal**. Por outro lado, o **sintagma verbal** constitui o predicado; caso o verbo do predicado seja significativo, será o seu núcleo.

Existem ainda os sintagmas adverbial[9], preposicional e adjectivo cujo núcleo é, respectivamente, um advérbio, preposição ou adjectivo.

O Anexo B apresenta os tipos de frases e classificação de períodos compostos e descreve, sucintamente, as funções e constituintes sintácticos da oração.

2.3 Informação Semântica

A estrutura sintáctica de uma frase não reflecte o seu significado. Por exemplo, o sintagma nominal `os braços` pode ter diferentes significados consoante se esteja a falar dos membros superiores do corpo, das peças que fazem parte de uma cadeira ou dos esteiros de um rio. As três interpretações têm a mesma estrutura sintáctica; os diferentes significados surgem da ambiguidade do sentido da palavra `braços`.

[7] Também denominados grupos.
[8] Também conhecido como elemento fundamental
[9] Também conhecido por circunstancial

O facto de `os braços` poderem significar os membros superiores ou esteiros de um rio é referente ao conhecimento da língua e é independente da situação em que a palavra `braços` é utilizada. Depois de saber qual o significado correcto da palavra, por exemplo `os braços do rio`, existe ainda o problema de determinar a que rio o sintagma `os braços` se refere. Aqui, o significado depende da situação em que a frase foi produzida.

Esta diferença permite considerar dois problemas distintos para a identificação do significado: o significado independente do contexto e o significado dependente do contexto [Allen 1994].

Embora grande parte da linguagem seja altamente dependente do contexto, existe uma estrutura semântica considerável que lhe é independente. Muito deste conhecimento semântico pode ser encontrado num dicionário como, por exemplo, as propriedades semânticas básicas das palavras (ou seja, se referem relações ou objectos), os diferentes significados possíveis de cada palavra e que significados se podem combinar para formar estruturas semânticas maiores.

2.4 Representação da Informação Linguística

A maioria das tarefas de processamento da linguagem, se não a sua totalidade, pode ser vista como a resolução de **ambiguidade** num ou mais níveis de informação linguística. Uma unidade de estudo (seja ela a palavra, oração ou período) é ambígua se puderem ser construidas múltiplas representações linguísticas alternativas. Por exemplo, a frase `A mãe observa a filha no parque de binóculos` pode ter os seguintes significados:

a. A mãe observa a filha que está no parque com uns binóculos

b. A mãe observa de binóculos a filha que está no parque

c. A mãe, que está no parque, observa a filha com uns binóculos

d. A mãe, que está no parque de binóculos, observa a filha

Os modelos de representação e os algoritmos de extracção de informação linguística introduzem formas de resolver algumas ambiguidades. Por exemplo, decidir se `lava` é um verbo ou um nome pode ser resolvido através de um algoritmo de etiquetagem morfo-sintáctica[10]; decidir se `planta` é um indivíduo do reino vegetal, a parte inferior do pé ou o desenho em projecção horizontal de uma construção pode ser resolvido através de um algoritmo de desambiguação do significado da palavra[11]. Estas são duas formas importantes de **desambiguação lexical**.

[10] Do inglês, *part-of-speech tagging*.
[11] Do inglês, *word sense desambiguation*.

Por outro lado, decidir, na frase atrás apresentada, se `mãe` e `binóculos` referem a mesma ou entidades diferentes é um exemplo de **desambiguação sintáctica**, que pode ser resolvida através de um algoritmo de análise sintáctica probabilística[12].

2.4.1 Representação morfológica da palavra

A análise morfológica[13] é o processo que permite reconhecer quais os morfemas constituintes de uma palavra e construir uma representação estruturada adequada. Essa representação inclui o **radical** da palavra processada e as suas características morfológicas (classe gramatical, flexão, etc.) Para esta análise é necessário conhecer as regras ortográficas (por exemplo, na língua Portuguesa, o plural constroi-se adicionando o morfema `-s-` ao singular dos substantivos terminados em vogal ou ditongo) e morfológicas (na língua Portuguesa, o substantivo `cais` é invariável) da língua.

Por outro lado, no processamento morfológico de um texto, interessa separar as palavras. Normalmente, elas são separadas umas das outras através de espaços, mas estes não são suficientes: por exemplo, `Porto Santo`, embora contenha um espaço, deve constituir uma entidade única funcionando como uma palavra individual.

Nalgumas aplicações, por exemplo em recuperação de informação, não é necessário fazer uma análise da palavra, mas interessa apenas transformá-la no seu radical. A este processo (retirar as terminações da palavra), dá-se o nome de **radicalização**[14].

Noutras tarefas de processamento da língua interessa saber qual a **raíz**[15] da palavra e não o seu radical. Duas palavras têm a mesma raíz, se tiverem como base a mesma palavra. Por exemplo, embora as palavras `fugiu` e `foge` tenham formas distintas, a sua raíz – `fugir` – é comum. Ao processo de mapear uma palavra com a sua raíz dá-se o nome de **lematização**[16].

Quando as palavras são ambíguas existem várias representações morfológicas possíveis. Neste caso, todas elas devem ser consideradas para posterior desambiguação.

2.4.2 Representação sintáctica da frase

A estrutura sintáctica de uma frase indica a forma como as suas palavras se relacionam entre si: como se agrupam para formar sintagmas, que palavras modificam outras palavras e que palavras são de importância central na frase. Esta estrutura pode

[12] Do inglês, *probabilistic parsing*.
[13] Do inglês, *morphological parsing*.
[14] Correspondente à palavra *stemming*, na língua Inglesa.
[15] Correspondente à palavra *lemma*, na língua Inglesa.
[16] Do inglês, *lemmatization*.

identificar também o tipo de relações existentes entre sintagmas e guardar outras informações passíveis de serem processadas mais tarde.

A maioria das representações sintácticas da linguagem são baseadas na noção de **gramática livre de contexto**[17] (GLC) formalizada por Chomsky [1956] e, de forma independente, por Backus [1959]. Uma GLC consiste num conjunto de regras, onde cada uma reflecte a forma como os símbolos da linguagem podem ser agrupados e ordenados e um léxico de palavras e símbolos. Os símbolos estão divididos em duas classes: os que correspondem às palavras da língua são chamados terminais; os que exprimem grupos ou generalizações são os não-terminais. O léxico é o conjunto de regras que introduz os símbolos terminais. As regras da gramática são construidas de forma hierárquica, permitindo gerar novas regras a partir de outras já construidas.

Uma GLC pode ser vista como um dispositivo que, dada uma frase, gera a estrutura sintáctica correspondente. Esta estrutura é usualmente representada sob a forma de uma árvore – a **árvore sintáctica**. A Figura 2.1 apresenta a árvore sintáctica para a frase `A mãe de binóculos observa a filha`.

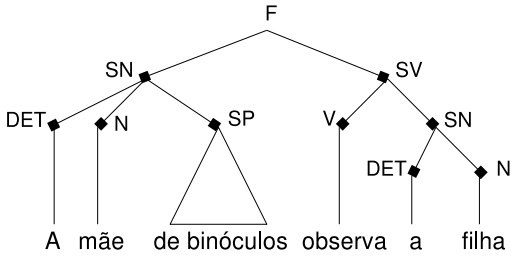

Figura 2.1: A árvore sintáctica da frase `A mãe de binóculos observa a filha`.

A árvore sintáctica fornece detalhes sobre a estrutura dos constituintes (como o sintagma nominal ou verbal) e identifica a classe gramatical de cada palavra.

Numa frase sintacticamente ambígua, não existe apenas uma árvore sintáctica que a representa, como é o caso da frase `A mãe observa a filha de binóculos`. A Figura 2.2 mostra as duas estruturas possíveis. Na primeira a mãe está de binóculos a observar a filha; na segunda é a filha quem tem os binóculos.

2.4.3 Representação semântica

Uma das tarefas chave da interpretação semântica examina que significados da palavra (considerada individualmente) podem ser combinados para obter um significado coerente da frase independente do contexto.

[17] Do inglês, *context-free grammar* (CFG).

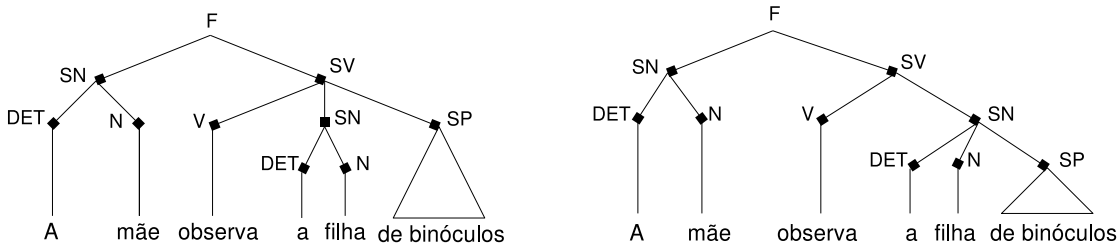

Figura 2.2: Duas árvores sintácticas possíveis para a frase `A mãe observa a filha de binóculos`.

A representação desta informação denomina-se **forma lógica**[18] e pode ser produzida directamente a partir da estrutura sintáctica da frase. A forma lógica retém os significados possíveis das palavras e identifica as relações semânticas com os sintagmas constituintes da frase. Muitas relações semânticas são capturadas utilizando um conjunto abstracto de ligações que podem existir entre o verbo e os seus sintagmas nominais. Ao serem determinadas as relações semânticas apropriadas, alguns significados das palavras são descartados por não se enquadrarem.

A Figura 2.3 ilustra quatro formas distintas de representar o significado da frase `A mãe observa a filha`.

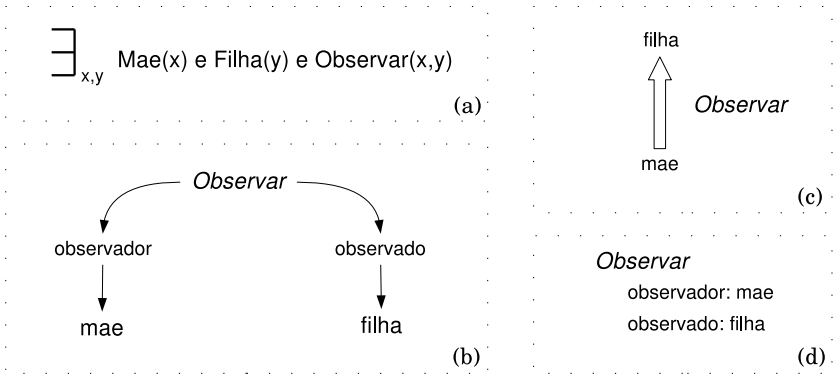

Figura 2.3: Quatro representações semânticas distintas da frase `A mãe observa a filha`. (a) ilustra a frase em Lógica de Primeira Ordem, (b) representa uma Rede Semântica[19], (c) é um diagrama de Dependência Conceptual[20] e (d) uma representação Baseada em Quadros[21].

[18] Do inglês, *logical form*.
[19] Do inglês, *Semantic Network*.
[20] Do inglês, *Conceptual Dependency diagram*.
[21] Do inglês, *Framed-Based representation*.

2.5 Ferramentas utilizadas

Para extrair a informação morfológica, sintáctica e semântica dos textos, foram utilizadas as três ferramentas de análise da língua Portuguesa que se descrevem em seguida.

2.5.1 POLARIS

O POLARIS [Lopes et al. 1994] é uma base de dados lexical para a língua Portuguesa que, dada uma palavra, devolve a sua raíz, classe gramatical e flexões utilizadas.

Quando uma palavra é ambígua, o POLARIS apresenta todas as possibilidades ordenadas pela sua frequência de utilização na língua; quando a palavra não existe na base de dados, a raíz é a própria palavra e a classe gramatical é desconhecida.

Sendo uma base de dados lexical, as entidades como `Porto Santo`, constituídas por mais que uma palavra da língua são consideradas como duas palavras distintas: `Porto` e `Santo`, sendo a análise feita a cada uma dessas palavras individualmente.

2.5.2 PALAVRAS

O PALAVRAS [Bick 2000] faz a análise morfo-sintáctica da frase utilizando um léxico e uma gramática de restrições[22] e gera a árvore sintáctica correspondente. Foi desenvolvido no contexto do projecto VISL (Visual Interactive Syntax Learning)[23] do *Institute of Language and Communication* da *University of Southern Denmark*.

É um sistema híbrido multi-nível: o léxico do analisador morfológico constitui a entrada para o primeiro nível da gramática de restrições, enquanto a saída do último nível é convertida numa árvore sintáctica através de gramáticas de estrutura de sintagma[24] específicas, que utilizam funções sintácticas (em vez de palavras) como terminais.

O seu analisador morfológico é baseado num léxico com 70000 lexemas e 15000 nomes. O resultado é processado através das cerca de 7500 regras que compõem a gramática de restrições para desambiguação morfológica e sintáctica (e alguma semântica).

O sistema produz sempre um resultado, mesmo em presença de frases incorrectas ou incompletas e tem uma baixa percentagem de erros: na determinação da classe gramatical da palavra é inferior a 1% e na geração da árvore sintáctica o erro ronda os 3–4% [Bick 2003].

[22] Do inglês, *constraint grammar*.
[23] Pode ser consultado em `http://www.visl.sdu.dk/`
[24] Do inglês, *phrase structure grammar*.

Gramática de restrições

Uma gramática de restrições é um paradigma metodológico para o processamento de língua natural introduzido nos anos 90 por Karlsonn et al. [1995]. As **regras**, escritas por linguistas, são dependentes do contexto e compiladas numa gramática que atribui etiquetas gramaticais às palavras e a outros símbolos do texto. As etiquetas atribuídas incluem a raíz, a flexão e derivação, a função sintáctica e dependência, a valência e o tipo semântico. Cada regra adiciona, remove, selecciona ou substitui uma etiqueta ou conjunto de etiquetas num determinado contexto da frase.

A cada etiqueta ou conjunto de etiquetas de uma palavra num ponto qualquer da frase podem ser adicionadas condições locais (distâncias definidas) ou globais (distâncias indefinidas). As condições para uma mesma regra podem ser interligadas (ou seja, podem ser condicionadas entre si), negadas ou bloqueadas consoante as palavras ou etiquetas que interferem.

As gramáticas de restrições contêm tipicamente milhares de regras que são aplicadas em passos progressivos cobrindo, em cada um, níveis mais avançados de análise. Em cada nível, primeiro são aplicadas **regras seguras**[25] e só depois as **regras heurísticas**, não sendo permitido a nenhuma regra remover a última etiqueta de determinado tipo. Desta forma obtém-se um elevado grau de robustez.

Um exemplo

Como exemplo, apresenta-se de seguida a estrutura gerada pelo sistema PALAVRAS ao lhe ser apresentada a frase `A mãe observa a filha Carlota`.

```
STA:fcl
=SUBJ:np
==>N:art('o' <artd> F S)        A
==H:n('mãe' <Hfam> F S) mãe
=P:v-fin('observar' PR 3S IND)  observa
=ACC:np
==>N:art('o' <artd> F S)        a
==H:n('filha' <Hfam> F S)       filha
==N<:prop('Carlota' M/F S)      Carlota
=.
```

[25] Do inglês, *safe rules*.

Nesta análise é possível observar que o PALAVRAS identifica:

- uma frase declarativa, identificada pela etiqueta `STA`;
- o sujeito (`SUBJ`), `a mãe`, e o predicado (`P`), `observa a filha`;
- o sintagma nominal (`np`) que constitui o sujeito é formado por um artigo (`art`), `a`, um nome (`n`), `mãe`;
- o verbo finito (`v-fin`), `observar`, que está no presente do indicativo (`PR`, `IND`) da 3ª pessoa do singular (`3S`);
- um complemento directo (`ACC`) que é formado por um sintagma nominal, `a filha Carlota`, cujo núcleo (`H`) é um nome, `filha`, caracterizado pelo nome próprio (`prop`) `Carlota`.

A estrutura hierárquica, ou seja, o nível de profundidade em que cada elemento se encontra na árvore sintáctica, é representada através do carácter `=`. Assim, o artigo `a` e o substantivo `mãe`, estão no segundo nível da árvore, sendo o constituinte `np` o seu ancestral imediato (nível 1).

O Anexo E descreve as etiquetas do PALAVRAS para a classificação de palavras e frases e para as funções e constituintes sintácticos das orações.

2.5.3 SIN2SEM

SIN2SEM é uma aplicação que transforma a árvore sintáctica obtida pelo PALAVRAS na forma lógica correspondente, utilizando Estruturas de Representação de Discurso[26] (ERD). As ERDs constituem a linguagem de representação semântica da Teoria de Representação do Discurso[27] [Kamp and Reyle 1993].

A cada árvore sintáctica corresponde uma ERD. Esta estrutura é representada através de um predicado Prolog com dois termos – o primeiro, possui a lista de referentes do discurso[28] e o segundo, o conjunto de condições da ERD[29]. Para cada referente é mantida a informação disponibilizada pelo PALAVRAS acerca da entidade que representa.

Embora esta teoria possua mecanismos que consideram o significado de uma frase no contexto em que esta é produzida, a aplicação SIN2SEM constrói as ERD considerando, de forma independente, o significado da frase em análise.

Esta aplicação foi construída no contexto de uma tese de mestrado [Salgueiro 2004] e utilizada no âmbito do CLEF-2004[30] [Quaresma et al. 2005].

[26] Do inglês, *Discourse Representation Structure* (*DRS*).
[27] Do inglês, *Discourse Representation Theory (DRT).*
[28] Do inglês, *discourse referent.*
[29] Do inglês, *DRS conditions.*
[30] CLEF: Cross-Language Evaluation Form.

Teoria de Representação do Discurso

A **Teoria de Representação do Discurso** (TRD) proposta por Kamp and Reyle [1993] é uma das teorias de semântica dinâmica que apareceram nos últimos 20 anos e cujo principal interesse consiste em considerar o significado dependente do contexto.

A arquitectura da TRD baseia-se no tratamento de **pressupostos** [Soames 1984]. Um pressuposto linguístico é um requisito que a frase impõe no contexto em que é utilizada. Quando um contexto não satisfaz os pressupostos impostos pela frase, é alterado ou actualizado para um novo contexto que os satisfaça.

Na TRD, as representações semânticas são especificadas em termos de uma linguagem sobre **Estruturas de Representação do Discurso** (ERDs). Estas são formadas por dois elementos básicos: um conjunto de Referentes do Discurso (RD), geralmente referido como o seu "universo", e um conjunto de condições. Intuitivamente, os **referentes** identificam as entidades referidas no discurso, enquanto as **condições** representam as restrições (propriedades, relações) associadas a essas entidades.

Um exemplo

Como exemplo, apresenta-se a estrutura gerada pelo sistema SIM2SEM ao lhe ser apresentada a árvore sintáctica representada na página 16.

```
drs(
    [def-A-'<hfam>'-f,def-B-'<hfam>'-f],
    [mãe(A),observar(A,B),filha(B),nome(B,'Carlota')]
)
```

Nesta DRS existem dois referentes, `A` e `B`. Ambos representam entidades do género feminino e a sua classificação semântica é `<hfam>` (ser humano da classe família)[31]; As restrições sobre as entidades são representadas através de 4 condições: a entidade `A` refere a mãe e `B` refere a filha, a segunda condição representa a acção de `observar` onde a entidade `A` representa o sujeito, e a entidade `B` o objecto e a quarta indica que o nome próprio de `B` é `Carlota`.

[31] Do inglês, *familiar human being*.

Capítulo 3

Classificação automática de textos

Nos últimos anos, as tarefas de gestão automática do conteúdo dos documentos (conhecida, de uma forma geral, por Recuperação de Informação) ganhou um estatuto proeminente na área dos sistemas de informação devido, em grande parte, ao aumento da disponibilidade de documentos em forma digital e à consequente necessidade do seu acesso de forma flexível por parte dos utilizadores. A **classificação de textos**, a actividade que consiste em etiquetar textos de língua natural com classes temáticas pertencentes a um conjunto pré-definido, é uma de tais tarefas.

Tendo aparecido nos anos sessenta, só no início dos anos noventa, com o aumento do interesse aplicacional e a disponibilidade de hardware mais poderoso, é que a classificação de textos se tornou numa sub-área principal dos sistemas de informação. Nos dias de hoje, é utilizada em vários contextos, desde a indexação automática de textos baseada num vocabulário controlado até à filtragem de documentos, geração automática de meta-dados, desambiguação do significados das palavras, preenchimento de catálogos hierárquicos de recursos Web e, em geral, qualquer aplicação que requeira a organização ou selecção de documentos.

Até finais dos anos oitenta, a aproximação mais popular à classificação de texto, pelo menos do ponto de vista operacional, era a da engenharia do conhecimento. Esta aproximação consiste em definir, manualmente, um conjunto de regras que codificam o conhecimento pericial sobre como classificar documentos num conjunto de classes. A partir dos anos 90 esta aproximação perdeu popularidade, especialmente na comunidade de investigação, a favor da aproximação de **Aprendizagem Automática** que utiliza um processo de inferência genérico e constrói, de forma automática, um classificador. As vantagens desta aproximação são a obtenção de uma exactidão[1] comparável à obtida por peritos do domínio e uma poupança considerável de potencial humano, já que não é necessária qualquer intervenção, quer de engenheiros de conhecimento quer

[1] Do inglês, *accuracy*.

de peritos, para a construção do classificador ou para alterá-lo para outro conjunto de classes.

Hoje em dia, a classificação de texto (CT) é uma disciplina no cruzamento entre as áreas de Recuperação de Informação (RI) e Aprendizagem Automática (AA), e como tal, partilha um conjunto de características com outras tarefas tais como a extracção de informação/conhecimento de textos e mineração de textos[2].

Este capítulo começa por definir o problema de CT descrevendo, em seguida, as tarefas mais importantes onde tem sido aplicado. A secção 3.3 apresenta as ideias principais da aproximação da AA à área da classificação e a secção 3.4 descreve os diversos passos para a construção de um classificador de texto. Finalmente, apresentam-se as ferramentas utilizadas na realização deste trabalho.

3.1 Definição

A classificação de textos pode ser definida como a tarefa de associar um valor Booleano a cada par $\langle d_j, c_i \rangle \in \mathcal{D} \times \mathcal{C}$, onde $\mathcal{D}$ é o domínio dos **documentos** e $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$ é o conjunto de **classes** pré-definidas. Um valor V associado a $\langle d_j, c_i \rangle$ indica que d_j pertence a c_i, enquanto um valor F indica que d_j não pertence a essa classe.

Mais formalmente, a tarefa consiste em aproximar a **função objectivo** desconhecida $\breve{\phi} : \mathcal{D} \times \mathcal{C} \rightarrow \{V, F\}$ (que descreve como os documentos devem ser classificados) através de uma função $\phi : \mathcal{D} \times \mathcal{C} \rightarrow \{V, F\}$ chamada **classificador** (um tipo de regra, hipótese ou modelo) tal que $\breve{\phi}$ e ϕ "coincidam o mais possível". A definição e medição deste grau de coincidência (a que chamaremos **desempenho**[3]) será apresentada na secção 3.4.3.

Este trabalho assume que:

- as classes são etiquetas simbólicas, não existindo nenhum conhecimento adicional acerca do seu significado para ajudar na construção do classificador;
- a atribuição dos documentos às classes utiliza apenas conhecimento **endógeno**, ou seja aquele que pode ser extraído a partir do documento em si. Isto significa que, apenas é utilizado o texto do documento sem outra informação como a data, fonte de publicação ou o seu tipo.

Estas suposições produzem os seguintes efeitos:

- os algoritmos utilizados são completamente genéricos e não dependem da disponibilidade de recursos específicos que podem ser difíceis de obter ou estar

[2]Do inglês, *text mining*.
[3]Do inglês, *effectiveness*.

simplesmente indisponíveis;

- a pertença de um documento a uma classe é não determinística já que, ao basear-se apenas no conhecimento endógeno, a classificação depende da semântica do documento que, como já foi referido, é uma noção subjectiva.

3.1.1 Problemas binários e não binários

Os problemas de classificação podem ser divididos consoante a cardinalidade do conjunto de classes $\mathcal{C}$. Se $|\mathcal{C}| = 1$ estamos em presença de um **problema binário** – um documento pertence ou não à classe em questão. Se $|\mathcal{C}| > 1$ o problema pode ser multi-etiqueta ou multi-classe. É **multi-etiqueta** quando um documento pode pertencer a mais que uma classe; é **multi-classe**, quando pertence a apenas uma das classes possíveis. Se existir uma taxonomia associada a $\mathcal{C}$ estamos em presença de um problema de classificação hierárquica.

Embora existam sistemas de CT desenvolvidos especificamente para problemas multi-etiqueta [McCallum 1999; Schapire and Singer 2000] e multi-classe [Lee et al. 2001; Weston and Watkins 1998], é comum decompô-los em múltiplos problemas independentes de classificação binária. Nos problemas multi-etiqueta constroem-se $|\mathcal{C}|$ classificadores binários independentes, um para cada classe de $\mathcal{C}$. Esta simplificação, no entanto, requer que as classes sejam estocasticamente independentes entre si, ou seja, quaisquer duas classes c' e c'' o valor de $\breve{\phi}(d_j, c')$ não pode depender do valor de $\breve{\phi}(d_j, c'')$ e vice-versa.

Se o resultado (saída) do classificador binário for um número real (em vez de um valor Booleano), a construção do classificador multi-classe utilizando $|\mathcal{C}|$ classificadores binários pode ser feita de forma semelhante ao caso multi-etiqueta: classificar um documento consiste em executar os $|\mathcal{C}|$ classificadores, sendo escolhida a classe correspondente ao classificador que produzir o maior valor. Este esquema, designado por **um-contra-todos** (UCT)[4], é conceptualmente simples e foi descoberto várias vezes, de forma independente, por diferentes investigadores [Rifkin and Klautau 2004].

Outra aproximação possível para resolver problemas multi-classe utilizando classificadores binários utiliza ideias da teoria de códigos de correcção de erros[5] na escolha do conjunto de classificadores binários a treinar e um método para combinar os diferentes classificadores [Crammer and Singer 2002; Dietterich and Bakiri 1995; Fürnkranz 2002]. Um desses métodos, conhecido como método de todos-os-pares ou **todos-contra-todos** (TCT)[6] utiliza $\binom{N}{2}$ classificadores binários; cada um é construído, não com

[4]Do inglês, *one-vs-all* (OVA).
[5]Do inglês, *error correcting code theory*.
[6]Do inglês, *all-pairs* ou *all-vs-all* (AVA).

o conjunto completo dos documentos, mas a partir daqueles pertencentes a um par de classes [Allwein et al. 2000]. A classificação de um documento pode, entre outros métodos, ser feita por maioria de votos (estratégia *max-win*) [Friedman 1996] ou utilizar as probabilidades derivadas pelos classificadores binários para estimar a probabilidade conjunta de todas as classes [Hastie and Tibshirani 1998].

No problema da classificação hierárquica, embora também existam sistemas que utilizam a taxonomia da classificação [Koller and Sahami 1997; Sun and Lim 2001], é comum transformá-lo numa aproximação multi-classe utilizando cada nível da taxonomia como um novo problema [Dumais and Chen 2000] ou "planificando" o espaço das classes [Sasaki and Kita 1998], onde cada folha da taxonomia é uma classe.

3.2 Aplicações de classificação de texto

A classificação automática de documentos remonta ao início dos anos sessenta com o trabalho de Maron [1961] sobre classificação probabilística de texto. Desde aí tem sido utilizada num número variado de aplicações. Esta secção descreve sucintamente algumas delas. Note-se, no entanto, que a fronteira entre as diferentes aplicações não é rígida e que algumas podem ser consideradas casos especiais de outras. Outras aplicações não descritas incluem a classificação do discurso através da combinação do reconhecimento da fala com a CT [Myers et al. 2000; Schapire and Singer 2000], a classificação de documentos multimédia através da análise de legendas textuais [Sable and Hatzivassiloglou 2000], a identificação do autor de textos de autores desconhecidos ou disputados [Forsyth 1999], a identificação da língua de textos de língua desconhecida [Cavnar and Trenkle 1994], a identificação automática do género do texto [Kessler et al. 1997] e a classificação automática de ensaios [Larkey 1998].

3.2.1 Indexação de documentos

A primeira utilização dos classificadores automáticos de texto e aquela que gerou mais investigação nos tempos iniciais foi a indexação automática de documentos para sistemas de RI tendo como base um dicionário controlado[7] [Borko and Bernick 1963; Field 1975; Gray and Harley 1971; Hamill and Zamora 1980; Heaps 1973; Hoyle 1973; Maron 1961]. A indexação associa, a cada documento, uma ou mais palavras-chave ou frases-chave que descrevem o seu conteúdo. As palavras e frases pertencem a um conjunto finito – o dicionário controlado – que, muitas vezes, consiste num *thesaurus* temático hierárquico. Este trabalho costuma ser feito por pessoas treinadas na indexação.

[7] Do inglês, *controlled dictionary*.

Se as entradas no dicionário forem vistas como classes, a indexação de texto é uma instância da tarefa de CT. Existem vários classificadores de texto concebidos explicitamente para a indexação de documentos como, por exemplo, os descritos em [Fuhr and Knorz 1984; Robertson and Harding 1984; Tzeras and Hartmann 1993].

A indexação automática com dicionário controlado está intimamente ligada ao tópico de construção automática de meta-informação. Nas bibliotecas digitais interessa etiquetar documentos com meta-informação que os descreve segundo vários aspectos. Alguma desta meta-informação é temática, tendo como função descrever a semântica do documento através de códigos bibliográficos, palavras-chave ou frases-chave. A construção desta informação pode ser vista como um problema de indexação com dicionário controlado e assim ser resolvido através técnicas de CT.

3.2.2 Organização de documentos

A indexação com um dicionário controlado é um caso específico do problema de organização e arquivação de documentos. Por exemplo, nos escritórios de um jornal, os anúncios "classificados" devem ser categorizados, antes da sua publicação, segundo o esquema adoptado pelo jornal. Embora muitos jornais tratem esta tarefa manualmente, ela pode ser realizada através de um sistema automático que escolhe a classe mais adequada para um dado anúncio. Neste caso, uma restrição típica é que a cada anúncio seja associada exactamente uma classe. Aplicações semelhantes são a organização de patentes em classes [Larkey 1999] para tornar a sua pesquisa mais eficiente, a colocação automática de notícias nas secções apropriadas (por exemplo, `Politica`, `Economia` ou `Tecnologia`) [Gonçalves et al. 2006] ou a organização automática de artigos de uma conferência em sessões.

3.2.3 Filtragem de texto

A filtragem de texto[8] consiste em classificar uma colecção dinâmica de textos (um fluxo de documentos) enviados de forma assíncrona pelo produtor para os consumidores de informação [Belkin and Croft 1992]. Na área das notícias existe um caso típico, onde o produtor é uma agência de notícias (por exemplo a Reuters ou Lusa) e o consumidor é um jornal [Hayes and Andersen 1990]. Neste caso, o sistema deve bloquear a entrega de documentos que não interessem ao consumidor; por exemplo, para um jornal desportivo, todas as notícias que não digam respeito a desporto.

A filtragem pode ser vista como um problema de classificação binária – documentos relevantes e não relevantes. Adicionalmente, o sistema de filtragem pode classificar

[8] Do inglês, *text filtering*.

os documentos que interessam em classes; para o exemplo anterior, todas as notícias desportivas são relevantes e podem ser classificadas segundo o desporto a que dizem respeito.

De forma semelhante, um filtro de correio electrónico pode ser treinado para descartar mensagens de lixo [Androutsopoulos et al. 2000; Drucker et al. 1999] e classificar as restantes num conjunto de classes de interesse para o utilizador [Cohen 1996].

3.2.4 Desambiguação do significado das palavras

A desambiguação do significado da palavra[9] consiste em encontrar o significado correcto para uma palavra ambígua (homónima ou homógrafa) num determinado texto. Esta tarefa é importante num grande número de aplicações, incluindo a representação de língua natural (ver secção 2.4) ou a indexação de documentos pelo significado da palavra (em vez da palavra em si) na área de RI. Pode ser vista como uma tarefa de classificação multi-classe se considerarmos o contexto das ocorrências da palavra como documentos e os seus significados como classes (ver, por exemplo, [Gale et al. 1993; Hearst 1991]).

Outros exemplos de possível resolução de ambiguidades da língua natural utilizando a CT automática são a correcção de erros ortográficos sensível ao contexto, ligação do sintagma preposicional[10], a etiquetagem morfo-sintáctica e escolha de palavras na tradução automática [Roth 1998].

3.2.5 Classificação hierárquica de páginas web

A CT obteve recentemente um grande interesse devido à sua utilização em aplicações Internet. Uma possível é a classificação automática de páginas Web num catálogo hierárquico. Se os documentos estiverem catalogados desta forma, pode ser mais simples encontrar o que se procura: navega-se primeiro na hierarquia de classes executando, depois, a pesquisa sobre uma ou várias classes de interesse.

Esta área de aplicação da CT tem duas particularidades: o hipertexto existente nos documentos e a estrutura hierárquica do conjunto de classes. As hiperligações constituem uma fonte de informação rica, já que podem ser compreendidas como indicadores de relevância entre a página ligada e a que liga. Técnicas que exploram esta característica num contexto de CT foram apresentadas em [Attardi et al. 1998; Chakrabarti et al. 1998; Gövert et al. 1999].

Como já referido, o problema da estrutura hierárquica pode ser resolvido através da

[9] Do inglês, *word sense disambiguation*.
[10] Do inglês, *prepositional phrase attachment*.

decomposição da classificação original num conjunto de problemas mais pequenos, correspondendo cada um a uma decisão num nó interno da hierarquia. Técnicas que utilizam esta aproximação foram apresentadas em [Dumais and Chen 2000; Koller and Sahami 1997; Ruiz and Srinivasan 1999].

3.3 A aproximação da Aprendizagem Automática

A partir do início dos anos noventa a aproximação da AA na construção de classificadores de texto ganhou popularidade, tornando-se, eventualmente, dominante, pelo menos na comunidade de investigação. Nesta aproximação, um processo indutivo genérico (também chamado aprendiz) constrói, de forma automática, um classificador para uma classe c_i através da observação das características de um conjunto de documentos previamente classificados por um perito. A partir destas características, o processo indutivo define aquelas que um novo documento d_x deve ter para poder ser classificado como pertencente a c_i.

No entanto, os documentos não são interpretados directamente pelo aprendiz ou classificador; é necessário realizar um processo de indexação que transforma um texto d_j numa representação compacta do seu conteúdo. A Figura 3.1 ilustra a aproximação de AA na construção de classificadores de texto.

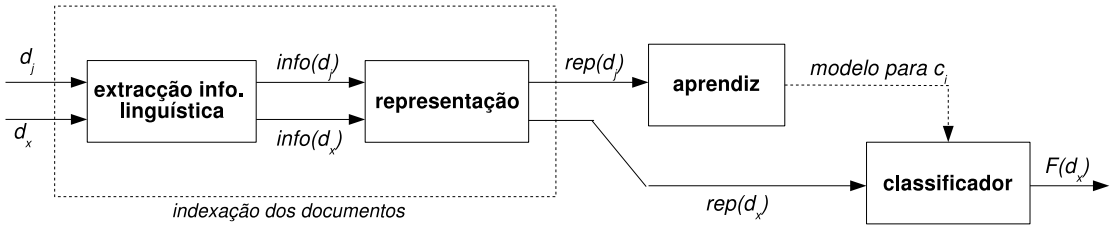

Figura 3.1: A aproximação da AA na classificação de textos: os documentos d_j previamente classificados são transformados numa representação compacta do seu conteúdo que, por sua vez, é apresentada ao aprendiz para induzir um modelo para a classe c_i. Este é posteriormente utilizado para classificar novos documentos d_x.

Na terminologia da AA, o problema da classificação é uma actividade de aprendizagem supervisionada, já que o processo é guiado (supervisionado) pelo conhecimento das classes e pelos exemplos previamente classificados.

3.3.1 Conjuntos de treino, teste e validação

Esta aproximação depende da existência de um conjunto inicial $\Omega = \{d_1, \ldots, d_{|\Omega|}\}$ de documentos previamente classificados no conjunto de classes $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$ em que o

sistema vai operar, ou seja, os valores da função $\breve{\phi} : \mathcal{D} \times \mathcal{C} \rightarrow \{V, F\}$ são conhecidos para todos os pares $\langle d_j, c_i \rangle \in \Omega \times \mathcal{C}$, com $\Omega \subset \mathcal{D}$. Um documento d_j é um exemplo positivo de c_i se $\breve{\phi}(d_j, c_i) = V$ e um exemplo negativo de c_i se $\breve{\phi}(d_j, c_i) = F$. Usualmente, o conjunto Ω dos documentos designa-se por **corpus** ou colecção.

Para avaliar o desempenho de um classificador, e antes da sua construção, costuma dividir-se o corpus inicial em dois conjuntos, não necessariamente do mesmo tamanho:

- um **conjunto de treino-e-validação** $TV = d_1, \ldots, d_{|TV|}$. Este é o conjunto de documentos cujas características são utilizadas na construção indutiva do classificador;
- um **conjunto de teste** $Te = d_{|TV+1|}, \ldots, d_{|\Omega|}$. Este conjunto é utilizado com a finalidade de testar o desempenho do classificador. Cada documento de Te é fornecido ao classificador, e as suas decisões $F(d_j, c_i)$ são comparadas com as do perito $\breve{\phi}(d_j, c_i)$; a medida do desempenho do classificador é baseada no número de vezes que o valor de $F(d_j, c_i)$ concorda com o valor de $\breve{\phi}(d_j, c_i)$.

Os documentos em Te não podem, de forma alguma, participar na construção indutiva do classificador; se tal acontecesse os valores experimentais seriam, provavelmente, irrealisticamente melhores e a avaliação não teria valor científico. Num sistema operacional, depois da avaliação, o classificador é, tipicamente, re-treinado com o corpus inicial Ω para melhorar o desempenho. Isto significa que os resultados da avaliação anterior constituem uma estimativa conservadora do desempenho real, já que o classificador final é treinado com mais dados que o classificador avaliado. A esta técnica dá-se o nome de **aproximação treino-e-teste**.

Outra técnica possível é a **validação-cruzada com k-pastas**[11], onde k classificadores diferentes $F_1, \ldots, F_k$ são induzidos através da partição inicial do corpus em k conjuntos distintos $Te_1, \ldots, Te_k$, aplicando iterativamente a aproximação treino-e-teste aos pares $\langle TV_i = \Omega - Te_i, Te_i \rangle$. O valor do desempenho final é obtido através do cálculo da média dos valores obtidos pelo desempenho individual dos k classificadores $F_1, \ldots, F_k$.

Em ambas as técnicas, e para optimizar o classificador, é comum afinar os parâmetros internos do aprendiz testando quais os valores que produzem melhor desempenho. Para tornar esta optimização possível, o conjunto $\{d_1, \ldots, d_{|TV|}\}$ na aproximação treino-e-teste, é ainda dividido num **conjunto de treino** $Tr = \{d_1, \ldots, d_{|Tr|}\}$ sobre o qual o classificador é induzido e um **conjunto de validação** $Va = \{d_{|Tr+1|}, \ldots, d_{|TV|}\}$, no qual são realizados repetidos testes com o objectivo de optimizar os parâmetros; na aproximação da validação cruzada é feita a adaptação óbvia. Pelas mesmas razões que não se testa um classificador com os documentos com o qual foi treinado, também

[11] Do inglês, *k-fold cross-validation*.

não se testa o classificador com os documentos com os quais foi optimizado; isto é, os conjuntos de teste e validação devem, também eles, ser distintos.

Dado um corpus Ω, define-se, **generalidade**[12] $g_\Omega(c_i)$ da classe c_i, como a percentagem de documentos que pertence a c_i:

$$g_\Omega(c_i) = \frac{|d_j \in \Omega | \breve{\phi}(d_j, c_i) = V|}{|\Omega|}$$

A generalidade dos conjuntos de treino g_{Tr}, validação g_{Va} e teste g_{Te} de uma classe c_i são definidos de forma idêntica, substituindo, na equação, Ω por Tr, Va e Te, respectivamente.

3.4 Construção do classificador

A construção de um classificador de texto inclui, essencialmente, três fases. Em primeiro lugar é necessário encontrar uma representação dos documentos, transformando-os num conjunto de características (ou atributos) que os descrevem. Esta representação, é depois utilizada no processo de indução do classificador. No final é feita a avaliação do seu desempenho.

3.4.1 Representação do documento

Os documentos de texto não podem ser interpretados directamente por um algoritmo de aprendizagem ou pelo classificador; é necessário realizar um processo de indexação que transforma um texto d_j numa representação compacta do seu conteúdo nos documentos de treino, validação e teste.

Para encontrar tal representação são, tipicamente, realizados os seguintes passos:

1. **Extracção de informação relevante**. A escolha da representação depende do que for considerada a unidade de significado textual. Tipicamente é utilizada a palavra como unidade básica, mas podem ser construidas características mais complexas, utilizando formas estruturadas (i.e. expressões multi-palavra) ou adicionando informação linguística (morfológica, sintáctica ou semântica).

2. **Normalização**. Este passo, quando utilizado, aplica a radicalização/lematização às palavras, aglomerando as que têm a mesma raíz/lema na mesma característica.

[12] Do inglês, *generality*.

3. **Redução de dimensão**[13]. Neste passo, a representação básica do documento é modificada retirando características irrelevantes ou inapropriadas e/ou introduzindo novas características por combinação das originais.

Extracção de informação relevante

Como referido no Capítulo 2, a mesma palavra pode ter significados diferentes em frases distintas e a mesma frase pode ter diferentes significados dependendo do orador, da audiência e da situação. O **contexto** constitui, assim, um problema fundamental ao lidar com a língua natural, já que possui uma influência substancial no significado de um excerto de texto (ver [Blair 1992]).

Em CT, diferentes aproximações para representar documentos reconhecem ou ignoram estas dependências em alguma extensão e podem ser estruturadas de acordo com o nível de análise do texto:

- **sub-palavra**: a palavra é decomposta, podendo ser analisada a sua morfologia;
- **palavra**: a palavra é a unidade básica, podendo ser estudada a sua informação lexical;
- **multi-palavra**: a unidade básica é a oração e a sua informação sintáctica pode ser analisada;
- **semântico**: o significado do texto é analisado;
- **pragmático**: o significado do texto é analisado de acordo com o contexto e situação em que ocorre.

Às unidades básicas de cada nível dá-se o nome de **termos de indexação**[14]. Por exemplo, no nível palavra os termos de indexação referem palavras, enquanto no nível multi-palavra, podem referir orações ou mesmo frases.

Em RI utiliza-se, na maior parte das vezes, o **modelo do espaço de vectores**[15] (MEV) para representar os documentos. Neste modelo, um documento d_j é representado através de um vector de pesos $\vec{d_j} = \langle w_{1j}, \ldots, w_{|\mathcal{T}|j} \rangle$, onde $\mathcal{T}$ é o conjunto de termos (atributos ou características) que ocorrem pelo menos uma vez em pelo menos um documento do corpus Ω e onde $0 \leq w_{kj} \leq 1$ representa uma medida da contribuição do termo t_k para o documento d_j. Usualmente, o conjunto $\mathcal{T}$ dos termos que ocorrem no corpus, designa-se por **dicionário**.

[13] Do inglês, *dimensionality reduction*.
[14] Do inglês, *indexing terms*.
[15] Do inglês, *vector space model (VSM)*.

A versão mais simples do MEV que utiliza a palavra como termo de indexação, é a representação saco-de-palavras[16]. Um saco é um conjunto no qual são permitidos elementos repetidos; além da presença do termo considera-se, também, a sua frequência. Assim, nesta representação, um documento é representado pelas palavras que contém, sendo ignorada a ordem das palavras e a pontuação.

A secção 4.1 descreve os níveis de representação de documentos, referindo os principais trabalhos desenvolvidos em cada um deles.

Normalização

Este passo costuma ser aplicado quando a unidade básica de informação é a palavra e tradicionalmente, aplica-se a radicalização. A lematização é importante nas línguas com uma morfologia muito rica, como a língua Portuguesa, onde centenas de formas diferentes são derivadas a partir da mesma raíz; pode ser utilizada como método de normalização se a raíz de cada palavra for disponibilizada através da análise linguística.

A aplicação da normalização em CT é controversa. Embora existam trabalhos que indicam que a sua aplicação diminui o desempenho [Baker and McCallum 1998], outros adoptam-na [Larkey and Croft 1996; Ng et al. 1997; Schütze et al. 1995; Wiener et al. 1995; Yang and Liu 1999] já que ela reduz a dimensão do espaço de termos e o nível de dependência entre termos.

Redução de dimensão

Neste passo de pré-processamento, modifica-se a representação básica do documento removendo termos irrelevantes ou inapropriados e/ou introduzindo novos termos por combinação dos originais.

As principais motivações são a protecção contra o sobre-ajustamento[17] e o aumento da eficiência computacional. No entanto, ao não considerar alguns termos corre-se o risco de remover informação potencialmente útil sobre o significado dos documentos. É pois claro que, para obter um desempenho óptimo, o processo de redução deva ser realizado com cuidado.

Os classificadores sobre-ajustados aos dados possuem bons níveis de desempenho quando reclassificam os exemplos de treino mas, quando aplicados a novos exemplos, os seus níveis de desempenho são muito baixos. Por outro lado, por vezes, a eficiência computacional só pode ser obtida através da redução da dimensão do espaço de carac-

[16] Do inglês, *bag-of-words*.
[17] Do inglês, *overfitting*.

terísticas já que muitos algoritmos de aprendizagem não conseguem manipular espaços de características de grandes dimensões, alguns impondo, até, restrições de tempo e memória.

Existem duas formas distintas de redução de dimensão consoante a tarefa é realizada local ou globalmente, ou seja, realizada de forma independente para cada classe ou para o conjunto de todas as classes. Na **redução de dimensão local**, para cada classe c_i, são escolhidos $|\mathcal{T}'| \ll |\mathcal{T}|$ termos para a construção do classificador responsável pela classe c_i. Neste caso, existem diferentes subconjuntos de vectores $\vec{d_j}$ para cada classe. Na **redução de dimensão global** são escolhidos $|\mathcal{T}'| \ll |\mathcal{T}|$ termos que são utilizados na construção dos classificadores para todas as classes de $\mathcal{C}$. Usualmente, esta distinção não influencia o tipo de técnica utilizada na redução de dimensão já que a maioria das técnicas podem ser utilizadas em ambos os casos.

Uma segunda distinção, ortogonal à anterior, divide as técnicas de redução de dimensão consoante a forma como é encontrada a "nova" representação dos documentos. Se a nova representação consiste num subconjunto dos termos originais, o processo designa-se por **selecção de termos**; se os termos forem obtidos por combinação ou transformação dos originais, designa-se por **extracção de termos**. Ao contrário da primeira distinção, aqui são utilizadas técnicas muito distintas. Várias funções têm sido propostas tanto da área da teoria da informação como da álgebra linear e os seus méritos têm sido testados experimentalmente avaliando o desempenho do classificador.

A **ponderação de características** pode ser considerada uma forma "leve" de selecção já que não remove certas dimensões de informação mas ajusta a sua influência relativa. O anexo C apresenta as técnicas mais usuais de redução de dimensão, assumindo uma representação "saco-de-palavras". No entanto, a sua maioria pode ser aplicada a outras representações de forma semelhante.

3.4.2 Indução do classificador

O processo de classificação apresentado na secção 3.1 designa-se por **classificação "rígida"**, porque existe uma decisão de V ou F para cada par $\langle d_j, c_i \rangle$. Embora uma automatização completa do processo de CT exija este tipo de classificação, uma automatização parcial pode ter requisitos diferentes. Por exemplo, dado um documento d_j, um sistema pode simplesmente posicionar as classes de $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$ de acordo com a sua adequação a d_j, sem tomar uma decisão rígida para qualquer uma. Alternativamente, dada uma classe c_i, um sistema poderá posicionar os documentos do conjunto $\mathcal{D}$ de acordo com a sua adequação estimada a c_i. Estas formas de classificação são, por vezes, denominadas, respectivamente, por classificação por **graduação-de-classes**[18] e

[18] Do inglês, *category-ranking categorization*.

classificação por **graduação-de-documentos**[19] [Yang and Liu 1999].

Na classificação por graduação, a construção de um classificador para a classe $c_i \in \mathcal{C}$ consiste, de uma forma geral, na definição de uma função $VC_i : \mathcal{D} \rightarrow [0, 1]$ que, dado um documento d_j, devolve o seu **valor de classificação**[20]. Este valor é um número entre 0 e 1 que representa a evidência de d_j pertencer à classe c_i. Na classificação por graduação-de-documentos, os documentos são ordenados de acordo com o seu valor VC_i para cada classe c_i; na classificação por graduação-de-classes e para um determinado documento d_j, posicionam-se as diferentes classes de $C = c_1, \ldots, c_{|C|}$ de acordo com seus valores VC_i.

Um classificador "rígido" pode ser construído de duas formas distintas. A primeira consiste na definição de uma função $VC_i : \mathcal{D} \rightarrow \{V, F\}$; a segunda consiste na definição de uma função $VC_i : \mathcal{D} \rightarrow [0, 1]$ análoga ao caso da classificação por graduação, seguida da definição de um **valor de limiar**[21] τ_i tal que um valor de $VC_i(d_j) \geq \tau_i$ é interpretado como V e um valor de $VC_i(d_j) < \tau_i$ é interpretado como F.

A determinação do valor de limiar τ_i (que pode depender das restrições impostas pela aplicação) pode ser feita de forma analítica ou experimental. O método analítico designa-se por **valor de limiar probabilístico**[22] e pode ser utilizado quando o algoritmo disponibiliza estimativas de probabilidades. O cálculo experimental pode ser realizado utilizando o valor de limiar VC ou corteS[23] [Cohen and Singer 1999; Schapire et al. 1998; Wiener et al. 1995], o valor de limiar proporcional ou corteP[24] [Iwayama and Tokunaga 1995; Larkey 1998; Lewis 1992a; Lewis and Ringuette 1994; Wiener et al. 1995] ou o valor de limiar fixo ou corteR[25] [Field 1975; Lam et al. 1999]. Lewis [1992a] e Yang and Liu [1999] apresentam e testam os diferentes métodos.

Para calcular o valor de classificação têm sido utilizados diversos algoritmos, alguns tradicionalmente utilizados em AA, outros em RI.

Diferentes algoritmos de indução implementam diferentes paradigmas que permitem a pesquisa em diferentes espaços de hipóteses. Existem aproximações quantitativas e simbólicas, algoritmos baseados em distâncias, classificadores lineares e não lineares, classificadores que se adaptam quando lhe são apresentados novos exemplos e classificadores que atrasam o processamento até ser necessário classificar um exemplo.

A secção 4.2 descreve diversos algoritmos que implementam alguns destes paradigmas, referindo os principais trabalhos desenvolvidos em cada um deles.

[19] Do inglês, *document-ranking categorization*.
[20] Do inglês, *categorization status value, CSV*.
[21] Do inglês, *threshold value*.
[22] Do inglês, *probability thresholding*.
[23] Do inglês, *CSV thresholding* ou *Scut*.
[24] Do inglês, *proportional thresholding* ou *Pcut*.
[25] Do inglês, *fixed thresholding* ou *Rcut*.

3.4.3 Avaliação do classificador

Normalmente, a avaliação experimental de um classificador mede, não a sua eficiência mas, o seu desempenho, ou seja, a sua capacidade de tomar as decisões de classificação acertadas. Podem ser calculadas um conjunto de medidas de desempenho que fornecem uma medida da distância entre a classificação humana (presente no conjunto de teste) e aquela produzida pelo sistema automático.

Precisão e cobertura

O desempenho do sistema de classificação é, na maior parte das vezes, medido em termos das noções clássicas de RI de **precisão**[26] (π) e **cobertura**[27] (ρ), adaptadas ao caso da CT. Estas medidas são definidas relativamente a uma classe particular c_i, sendo depois calculada uma média para obter os valores de π e ρ para o conjunto de todas as classes:

- π_i, é definida como sendo a probabilidade condicional $P(\breve{\phi}(d_x, c_i) = V | \phi(d_x, c_i) = V)$. Representa a probabilidade dos documentos d_x pertencerem a c_i, quando assim são classificados.

- ρ_i, é definida como $P(\phi(d_x, c_i) = V | \breve{\phi}(d_x, c_i) = V)$. Representa a probabilidade do classificador decidir correctamente os documentos d_x pertencentes a c_i.

Estas probabilidades devem ser entendidas como probabilidades subjectivas (já que medem a expectativa do sistema se comportar correctamente quando classifica um documento em c_i) e podem ser estimadas através da **tabela de contingência**[28] da classe c_i, num determinado conjunto de teste. A Tabela 3.1 esquematiza-a, onde FP_i (falsos positivos) representa o número de documentos do conjunto de teste que foram incorrectamente classificados em c_i; FN_i (falsos negativos), VP_i (verdadeiros positivos) e VN_i (verdadeiros negativos) são definidos de forma similar.

		Perito	
		Sim	**Não**
Classificador	**Sim**	VP_i	FP_i
	Não	FN_i	VN_i

Tabela 3.1: A tabela de contingência para a classe c_i.

[26] Do inglês, *precision*.
[27] Do inglês, *recall*.
[28] Do inglês, *contingency table*.

Estimativas da precisão e cobertura relativas à classe c_i podem ser obtidas pelas fórmulas

$$\hat{\pi}_i = \frac{VP_i}{VP_i + FP_i} \qquad \hat{\rho}_i = \frac{VP_i}{VP_i + FN_i}$$

As estimativas de π e ρ, podem ser obtidas através da micro-média[29] ou macro-média[30]. Na **micro-média** somam-se as decisões individuais; a tabela de contingência "global" (obtida somando os valores das tabelas de contingência de todas as classes c_i) é utilizada para calcular a precisão ($\hat{\pi}^{\mu}$) e cobertura ($\hat{\rho}^{\mu}$) médias. Na **macro-média**, a precisão e cobertura são calculadas para cada classe c_i, fazendo depois a média para obter a precisão ($\hat{\pi}^{M}$) e cobertura ($\hat{\rho}^{M}$) globais.

$$\hat{\pi}^{\mu} = \frac{VP}{VP + FP} = \frac{\sum_{i=1}^{|\mathcal{C}|} VP_i}{\sum_{i=1}^{|\mathcal{C}|}(VP_i + FP_i)} \qquad \hat{\rho}^{M} = \frac{\sum_{i=1}^{|\mathcal{C}|} \hat{\pi}_i}{|\mathcal{C}|}$$

$$\hat{\rho}^{\mu} = \frac{VP}{VP + FN} = \frac{\sum_{i=1}^{|\mathcal{C}|} VP_i}{\sum_{i=1}^{|\mathcal{C}|}(VP_i + FN_i)} \qquad \hat{\rho}^{M} = \frac{\sum_{i=1}^{|\mathcal{C}|} \hat{\rho}_i}{|\mathcal{C}|}$$

Estas duas medidas podem gerar valores muito distintos, especialmente se as classes tiverem diferentes níveis de generalidade, ou seja, se existirem classes com muitos e outras com poucos exemplos de treino positivos.

Outras medidas de desempenho

A **exactidão** ($\hat{A}$) e o **erro** ($\hat{E}$) são medidas alternativas a π e ρ muito utilizadas na literatura de AA. No entanto, isto não acontece em CT, uma vez que o grande valor do denominador torna estas medidas muito mais insensíveis a uma variação no número de decisões correctas ($VP + VN$) que π e ρ.

$$\hat{A} = \frac{VP + VN}{VP + VN + FP + FN} \qquad \hat{E} = \frac{FP + FN}{VP + VN + FP + FN} = 1 - \hat{A}$$

Além disso, Yang and Liu [1999] mostra que, ao utilizar a exactidão como medida de avaliação e se o problema for de baixa generalidade, o classificador trivial de rejeição[31], ($\phi(d_j, c_i) = F$ para todos os documentos d_j e classes c_i) tende a ter um melhor desempenho que qualquer classificador não trivial; por outro lado, a sintonização de parâmetros (no conjunto de validação) de um classificador não trivial resulta na escolha

[29] Do inglês, *micro-average*.
[30] Do inglês, *macro-average*.
[31] Do inglês, *trivial rejection classifier*.

de valores que fazem o classificador comportar-se de forma muito parecida com aquele classificador trivial.

Medidas de desempenho combinadas

A precisão e a cobertura não fazem sentido se forem consideradas de forma independente uma da outra. Por exemplo, o classificador trivial de aceitação[32] ($\phi(d_j, c_i) = V$ para todos os documentos d_j e classes c_i) tem cobertura máxima ($\rho = 1$); no entanto, a sua precisão é, tipicamente, muito baixa. Por outro lado, a prática de RI diz-nos que valores altos de precisão, podem ser obtidos à custa de baixos valores de cobertura.

Na prática, sintonizando os valores de limiar τ_i, a função $VC_i : D \rightarrow \{V, F\}$ torna-se mais liberal (melhorando ρ_i em detrimento de π_i) ou mais conservativa (melhorando π_i em detrimento de ρ_i) [Riloff and Lehnert 1994]. Um classificador deve, desta forma, ser avaliado através de uma medida de desempenho combinada que seja função de π e ρ. Várias medidas têm sido propostas; as mais frequentes são:

- **precisão-média-de-11-pontos**. Cada valor de limiar τ_i é, sucessivamente, fixado em valores para os quais ρ_i seja $0.0, 0.1, \ldots, 0.9, 1.0$; para estes 11 valores diferentes, calcula-se π_i e utiliza-se a sua média como medida de desempenho. Note-se que esta medida não faz sentido para funções binárias de VC_i já que, neste caso, não é possível variar ρ_i.

- **ponto de equilíbrio**[33]. A medida de desempenho é o valor para o qual π é igual a ρ. É obtido por um processo análogo ao anterior: variando os valores de limiar τ_i, calcula-se π em função de ρ e o ponto de equilíbrio é identificado como o valor de ρ (ou π) para o qual o gráfico intersecta a linha $\rho = \pi$. Se não existir um valor de τ_i para o qual π e ρ sejam exactamente iguais, τ_i recebe o valor para o qual π e ρ sejam os mais próximos e calcula-se um *ponto de equilíbrio interpolado* utilizando a média dos valores de π e ρ.

- **função** f_β. A medida de desempenho é o valor da função f_β para um $0 \leq \beta \leq +\infty$, onde

$$f_\beta = \frac{(\beta^2 + 1)\pi\rho}{\beta^2\pi + \rho}$$

  Nesta fórmula, β pode ser visto como o grau relativo de importância atribuído a π e ρ. Se $\beta = 0$, f_β coincide com π; se $\beta = +\infty$, f_β coincide com ρ. Tipicamente, utiliza-se um valor de $\beta = 1$, dando igual importância a π e ρ. Moulinier and

[32] Do inglês, *trivial acceptor classifier*.
[33] Do inglês, *breakeven point*.

Ganascia [1996] e Yang and Liu [1999] mostram que o valor do ponto de equilíbrio de um classificador ϕ é sempre menor ou igual que o seu valor de f_1.

Depois de escolhida a medida de desempenho, o classificador pode ser sintonizado (valores de limiar e outros parâmetros) para que o desempenho resultante seja máximo. A sintonização de um parâmetro p (seja ele um valor de limiar ou outro) é, normalmente, feita de forma experimental: para o conjunto de validação, são realizadas experiências com diferentes valores do parâmetro p, com os outros parâmetros p_k fixos (um valor por omissão[34] para parâmetros p_k ainda não sintonizados e o valor escolhido para parâmetros p_k já sintonizados). No final do processo, o valor que conduzir a um melhor desempenho é escolhido para p.

3.5 Ferramentas utilizadas

Para construir os diversos classificadores utilizados neste trabalho foram utilizadas três ferramentas que se descrevem de seguida.

3.5.1 WEKA

WEKA [Witten and Frank 2005] é um software livre desenvolvido pela *New Zealand's Waikato University*. É composto por uma colecção de algoritmos de AA para tarefas de mineração de dados[35] e comporta ferramentas de pré-processamento de dados, classificação, regressão, agrupamento, regras de associação e visualização. Utiliza uma representação dos dados no modelo de espaço de vectores e permite a utilização de vectores esparsos.

Tem sido utilizado numa vasta área de aplicações. Entre outras, existem aplicações relacionadas com texto (classificação e agrupamento de textos, desambiguação do significado de palavras, extracção de frases-chave), séries temporais, biologia e medicina.

Neste trabalho, utilizou-se a ferramenta WEKA para estudar o comportamento de alguns classificadores de AA na classificação de documentos escritos na Língua Portuguesa. Experimentaram-se classificadores probabilísticos (algoritmo Naïve de Bayes [John and Langley 1995]), árvores de decisão (implementação do algoritmo C4.5 [Quinlan 1993]) e máquina de vectores de suporte (utilizando o algoritmo de optimização sequencial mínimo[36] [Platt 1999]) para construir modelos que utilizam uma representação saco-de-palavras (ver secção 7.1.1).

[34] Do inglês, *default value*.
[35] Do inglês, *data mining*.
[36] Do inglês, *sequential minimal optimisation algorithm*.

3.5.2 SVMlight

SVM^{light} [Joachims 1999] é uma implementação da Máquina de Vectores de Suporte [Vapnik 1995] para problemas de classificação e regressão. Os algoritmos de optimização estão descritos em [Joachims 2002, 1999]. Esta implementação utiliza o modelo de espaço de vectores e, como é escalável nos requisitos de memória, pode ser utilizado em problemas com vários milhares de vectores de suporte de forma eficiente.

Este software tem sido utilizado numa vasta área de problemas incluindo classificação de texto, reconhecimento de imagens, bioinformática e aplicações médicas.

3.5.3 SVMlight-TK

SVM^{light}-TK [Moschitti 2004] é uma extensão ao software SVM^{light} que utiliza núcleos de convolução[37] para representar estruturas em árvore. Estes núcleos medem a semelhança entre duas árvores em termos das suas subestruturas comuns.

A ferramenta implementa dois tipos de núcleos, o núcleo de subárvores[38] [Vishwanathan and Smola 2003] e o núcleo de subconjuntos de árvores[39] [Collins and Duffy 2002] (secção 5.4.2). Intuitivamente, um núcleo de subárvores, para a subárvore com raíz no nó n da árvore A, contabiliza todos os n-descendentes até às folhas; o núcleo de subconjuntos de árvores além das anteriores, considera árvores cujas folhas são nós internos da árvore com raíz no nó n [Moschitti 2006b].

Esta ferramenta permite conjugar representações em árvore com as representações clássicas do espaço de vectores aplicando dois métodos distintos: somando os pares correspondentes (soma sequencial[40]) ou somando todas as árvores com todos os vectores (soma todos-vs-todos[41]).

A utilização destes núcleos mostrou bons resultados na graduação de árvores sintácticas [Collins and Duffy 2002] e na classificação de argumentos predicativos [Moschitti 2006a; Zhang and Lee 2003].

Neste trabalho foi utilizada para construir a máquina de vectores de suporte para os níveis de informação sintáctico e semântico (ver secções 7.1.3 e 7.1.4).

[37] Do inglês, *convolution kernels*.
[38] Do inglês, *subtree kernel*.
[39] Do inglês, *subset tree kernel*.
[40] Do inglês, *sequential summation*.
[41] Do inglês, *all-vs-all summation*.

Capítulo 4

Técnicas para construção de classificadores de texto

Como referido na secção 3.4, a construção de um classificador de texto implica a selecção do processo de indexação do documento, do método de determinação do valor de limiar e do algoritmo de indução do classificador. Estas escolhas não são independentes umas das outras já que, por exemplo, algumas técnicas de indução são apenas adaptáveis a algumas unidades básicas de informação e o tipo de aplicação pode definir qual o tipo de algoritmo de indução e qual o método para determinar do valor de limiar.

A maioria dos trabalhos de CT utiliza a palavra como termo de indexação e é aplicada a textos escritos na língua Inglesa. Este Capítulo descreve, sucintamente, as aproximações que têm sido propostas, quer ao nível dos termos de indexação, quer dos algoritmos de indução utilizados. A secção 4.3 apresenta algumas considerações sobre o desempenho comparativo dos diversos classificadores de texto e a secção 4.4 enumera algumas aplicações de CT para a língua Portuguesa.

4.1 Unidades básicas de informação

Como referido na secção 3.4.1, ao lidar com a língua natural, o contexto possui um papel fundamental e diferentes aproximações para representar o texto utilizam ou não essa informação. De uma forma geral, quanto mais alto o nível de representação, maior é o detalhe capturado; no entanto, a produção automática de representações de maior detalhe tem uma complexidade acrescida. Por exemplo, a produção de uma representação semântica requer um conhecimento substancial acerca do domínio de interesse e os métodos estado da arte apenas a conseguem aproximadamente.

Esta secção descreve os diferentes níveis de representação do documento, começando

pelo mais simples, cujo termo de indexação é a sub-palavra e terminando no nível semântico onde o significado da frase é analisado. Representações que utilizam a estrutura pragmática do texto ainda não foram exploradas.

4.1.1 Nível sub-palavra

Neste nível, a palavra é decomposta, podendo, ou não, ser analisada a sua morfologia. A representação mais popular é o n-**Grama**[1] em que os termos de indexação são cadeias de n caracteres construídas a partir do texto utilizando o modelo de espaço de vectores. Por exemplo, a representação 3-Grama (trigrama) da palavra `livro` é `_li`, `liv`, `ivr`, `vro`, `ro_`. Utilizando o modelo do espaço de vectores cada documento é representado através de um vector de n-Gramas, obtendo-se, uma representação saco-de-n-Gramas.

Esta representação modela, de forma natural, a semelhança entre palavras mas pode ser enganadora: embora palavras distintas como `computador` e `computadores` partilhem a maioria dos trigramas, o mesmo acontece entre as palavras `computador` e `comutador`.

Uma vantagem da representação n-Gramas é fornecer alguma robustez contra erros de ortografia. Além disso, a construção de uma representação n-Grama de um documento é muito simples, mesmo para línguas com muitas formas compostas (como o Alemão). Resultados experimentais podem ser encontrados em [Cavnar and Trenkle 1994; Huffman 1995; Neumann and Schmeier 1999].

4.1.2 Nível palavra

As palavras são muitas vezes, mesmo sem considerar o contexto, unidades de significado com pouca ambiguidade. Nas representações cujo termo de indexação é a palavra assume-se que as palavras homógrafas têm pouco impacto na representação do documento como um todo.

A simplicidade das representações baseadas na palavra constitui uma vantagem substancial sobre outras representações. Desenhar um algoritmo que decompõe um texto em palavras é muito simples: um que divida uma cadeia de caracteres em palavras utilizando os espaços brancos produz, usualmente, resultados satisfatórios para as línguas que não utilizam nomes compostos, como o Inglês ou Português.

A representação mais utilizada, vulgarmente apelidada por aproximação **saco-de-palavras**, transforma a cadeia de caracteres do documento num conjunto de palavras, registando sua frequência através de um método de ponderação de características (ver Anexo C.3) e ignorando toda a estrutura do documento.

[1] Do inglês, *n-Gram*.

Esta representação torna a CT acessível à maioria dos algoritmos de AA que requerem que cada exemplo (documento de texto d_j) seja descrito por um vector de dimensão fixa onde cada componente do vector representa um atributo (palavra) cujo valor (por exemplo, o número de vezes que a palavra ocorre no documento) representa o peso (w_{kj}) que a palavra tem na semântica do documento. Formalmente, o documento é representado pelo vector $\overrightarrow{d_j} = \langle w_{1j}, \ldots, w_{|\mathcal{T}|j} \rangle$, onde $\mathcal{T}$ é o conjunto de palavras consideradas.

Embora existam alguns trabalhos que consideram a ordem das palavras nos documentos (por exemplo, os sistemas FOIL [Cohen 1995] e SLEEPING EXPERTS [Cohen and Singer 1999]), a representação saco-de-palavras mostrou ser muito efectiva em RI e CT (ver, por exemplo, [Lewis 1992a]) e constitui a base para maioria dos trabalhos de CT. A radicalização é aplicada em alguns trabalhos [Larkey and Croft 1996; Ng et al. 1997; Schütze et al. 1995; Wiener et al. 1995; Yang and Liu 1999] e embora a maioria aplique uma lista de *stopwords*, alguns não o fazem [Joachims 2002; Lewis et al. 1996; Nigam et al. 2000; Riloff 1995]; uns utilizam pesos binários [Apté et al. 1994; Koller and Sahami 1997; Lewis and Ringuette 1994; Li and Jain 1998; Moulinier and Ganascia 1996; Schapire and Singer 2000; Schütze et al. 1995; Sebastiani et al. 2000] mas, uma grande parte, utiliza a medida $tfidf$ (ver Anexo C.3) normalizada.

4.1.3 Nível multi-palavra

As representações deste nível geralmente utilizam termos de indexação que incorporam informação sintáctica. A representação mais comum, normalmente designada por **Indexação de Sintagmas Sintácticos**[2], utiliza os sintagmas nominais como estrutura sintáctica [Dumais et al. 1998; Lewis 1992a,b].

Noutros trabalhos, são exploradas outras estruturas sintácticas. Por exemplo, são utilizadas sequências de palavras [Mladenić and Grobelnik 1998], conjuntos de radicais de n palavras (retirando *stopwords*) [Caropeso et al. 2001], pares `(radical, etiqueta-ms)` [Basili et al. 1999], pares `(palavra, conceito)` analisados ao nível da frase [Riloff and Lehnert 1994], sintagmas que satisfazem certos padrões (por exemplo, "`nome verbo-auxiliar compl-directo`") [Fürnkranz et al. 1998] e pares de palavras adjacentes radicalizadas (retirando *stopwords*) [Schütze et al. 1995].

Basili et al. [2004] extraem diversas características linguísticas, nomeadamente as etiquetas morfo-sintácticas, nominais complexos, nomes próprios e significados das palavras e utilizando diversas combinações dessas características experimentam três algoritmos de classificação (Rocchio, Rocchio parametrizado e máquinas de vectores de suporte) sobre 4 corpora em duas línguas.

[2] Do inglês, *Syntactic Phrase Indexing*.

Outras aproximações para gerar termos de indexação multi-palavra baseiam-se em métodos estatísticos, sendo analisada a co-ocorrência de padrões. Um grupo de palavras é considerado um termo de indexação se co-ocorrer frequentemente [Croft and Lewis 1990; Fagan 1987] ou se tiver uma distribuição similar à classe em questão [Baker and McCallum 1998]. Fuhr et al. [1991] e Tzeras and Hartmann [1993] utilizam o pré-processamento sintáctico para gerar os candidatos a termos de indexação, a partir dos quais filtram "bons" sintagmas utilizando métodos estatísticos; Lodhi et al. [2002] utilizam, como termos de indexação, sequências ordenadas de k caracteres não necessariamente contíguos.

4.1.4 Nível semântico

Os classificadores de texto só conseguem trabalhar de forma óptima se for possível capturar, de forma suficiente, a semântica dos documentos; no entanto, não é ainda possível extrair de forma automática a semântica de um texto e representá-la de forma operacional.

Até certo ponto, a semântica de um documento pode ser capturada utilizando taxonomias e linguagens de indexação com um vocabulário controlado. São exemplos, o *thesaurus* MeSH[3] (*Medical Subject Headings*) para a medicina, a classificação decimal Dewey[4] para bibliotecas e as classes reservadas do Yahoo![5] para a *World Wide Web*. Em qualquer destas taxonomias os documentos são colocados manualmente em uma ou mais classes e a sua estrutura hierárquica permite fazer inferências de relações é-do-tipo[6]. No entanto, esta indexação de documentos não está, normalmente, disponível e a afectação dos documentos às classes é, em si, um problema de CT.

A lógica de predicados e as redes semânticas [Brachman and Schmolze 1985] oferecem linguagens de representação poderosas e são muito operacionais. No entanto, representar totalmente os documentos na sua forma lógica requer uma tradução manual, tornando-as impraticáveis para a maioria das aplicações.

As Estruturas Representação do Discurso, apresentadas na secção 2.4.3, permitem obter uma representação semântica dos documentos. Este trabalho utiliza essa representação para criar classificadores de texto.

Os poucos trabalhos de classificação que utilizam (alguma) informação semântica não utilizam representações do documento estruturadas; Basili et al. [2004] utilizam algoritmos de desambiguação do significado da palavra para introduzir informação semântica

[3]Disponível em `http://www.nlm.nih.gov/mesh/`
[4]Disponível em `http://www.oclc.org/dewey/`
[5]Disponível em `http://docs.yahoo.com/docs/info/reserved.html`
[6]Do inglês, *is-a relationship*.

e Basili et al. [2005] utilizam o WordNet para medir a semelhança das palavras que utilizam como termos de indexação.

4.2 Algoritmos

Na secção 3.4.2, um classificador para a classe $c_i \in \mathcal{C}$ foi definido utilizando a função $VC_i(d_j)$ cujo valor representa a evidência de d_j dever ser classificado em c_i. Diferentes algoritmos utilizam diferentes aproximações para o cálculo de $VC_i(d_j)$. De seguida, apresentam-se aqueles que têm sido utilizados em CT, assumindo que estamos em presença de um problema de classificação "rígida". Pelo contexto, é possível determinar como e quando podem ser adaptados à classificação por graduação.

4.2.1 Classificadores probabilísticos

Para calcular $VC_i(d_j)$, os **classificadores probabilísticos** utilizam $P(c_i|\vec{d_j})$, a probabilidade de um documento representado pelo vector de termos $\vec{d_j} = \langle w_{1j}, \ldots, w_{|\mathcal{T}|j} \rangle$ pertencer à classe c_i. Para calcular esta probabilidade, aplicam o teorema de Bayes (o espaço de eventos é representado pelo espaço dos documentos)

$$P(c_i|\vec{d_j}) = \frac{P(c_i)P(\vec{d_j}|c_i)}{P(\vec{d_j})} \tag{4.1}$$

onde $P(\vec{d_j})$ é a probabilidade de um documento retirado aleatoriamente ser representado pelo vector $\vec{d_j}$ e $P(c_i)$ é a probabilidade de um documento retirado aleatoriamente pertencer à classe c_i.

Embora a estimação de $P(\vec{d_j}|c_i)$ seja problemática, torna-se simples ao assumir que quaisquer duas coordenadas do vector documento $w_{k'j}$ e $w_{k''j}$ (quando vistas como variáveis aleatórias) são estatisticamente independentes uma da outra. Esta suposição de independência é representada pela equação

$$P(\vec{d_j}|c_i) = \prod_{k=1}^{|\mathcal{T}|} P(w_{kj}|c_i) \tag{4.2}$$

Esta simplificação é utilizada pela maioria das aproximações probabilísticas de CT [Joachims 1998; Koller and Sahami 1997; Larkey and Croft 1996; Lewis 1992a; Lewis and Gale 1994; Li and Jain 1998; Robertson and Harding 1984] para construção de classificadores. Estes denominam-se, usualmente, por classificadores **naïve de Bayes** já que assumem ingenuamente esta independência que, na prática, não se verifica.

Uma das aproximações naïve de Bayes mais conhecidas é o classificador **binário independente** [Robertson and Spark Jones 1976], que resulta da utilização de vectores binários para representar os documentos. Neste caso, se escrevermos p_{ki} para representar $P(w_{kx} = 1|c_i)$, os factores $P(w_{kj}|c_i)$ da equação anterior podem ser escritos como

$$P(w_{kj}|c_i) = p_{ki}^{w_{kj}}(1-p_{ki})^{1-w_{kj}} = (\frac{p_{ki}}{1-p_{ki}})^{w_{kj}}(1-p_{ki}) \tag{4.3}$$

Como em CT o espaço de documentos está dividido em duas classes, c_i e o seu complemento $\bar{c}_i$, tal que $P(\bar{c}_i|\vec{d_j}) = 1 - P(c_i|\vec{d_j})$, ao substituir as Equações (4.2) e (4.3) na Equação (4.1) e aplicar o logaritmo, obtém-se

$$\log P(c_i|\vec{d_j}) = \log P(c_i) + \sum_{k=1}^{|\mathcal{T}|} w_{kj} \log \frac{p_{ki}}{1-p_{ki}} + \sum_{k=1}^{|\mathcal{T}|} \log(1-p_{ki}) - \log P(\vec{d_j}) \tag{4.4}$$

$$\log P(1-c_i|\vec{d_j}) = \log(1-P(c_i)) + \sum_{k=1}^{|\mathcal{T}|} w_{kj} \log \frac{p_{k\bar{i}}}{1-p_{k\bar{i}}} + \sum_{k=1}^{|\mathcal{T}|} \log(1-p_{k\bar{i}}) - \log P(\vec{d_j}) \tag{4.5}$$

onde $p_{k\bar{i}}$ representa $P(w_{kx} = 1|\bar{c}_i)$. Ao subtrair a Equação (4.5) da Equação (4.4), obtém-se uma única equação

$$\log \frac{P(c_i|\vec{d_j})}{P(1-c_i|\vec{d_j})} = \log \frac{P(c_i)}{1-P(c_i)} + \sum_{k=1}^{|\mathcal{T}|} w_{kj} \log \frac{p_{ki}(1-p_{k\bar{i}})}{p_{k\bar{i}}(1-p_{ki})} + \sum_{k=1}^{|\mathcal{T}|} \log \frac{1-p_{ki}}{1-p_{k\bar{i}}} \tag{4.6}$$

Esta equação é monotónica crescente com $P(c_i|\vec{d_j})$ e, como tal, pode ser utilizada directamente como $VC_i(d_j)$. Definir um classificador para a classe c_i requer, basicamente, a estimação dos $2|\mathcal{T}|$ parâmetros $\{p_{1i}, p_{1\bar{i}}, \ldots, p_{|\mathcal{T}|i}, p_{|\mathcal{T}|\bar{i}}\}$ a partir do conjunto de treino; é por isso que os classificadores probabilísticos também se designam por classificadores **paramétricos**.

A classificação de um documento, em geral, não requer o cálculo da soma das $|\mathcal{T}|$ parcelas como pode parecer à primeira vista: se considerarmos como valor inicial de VC_i os termos constantes da Equação 4.6, o valor final é calculado adicionando os valores que involvam os termos existentes no documento. Como os vectores documento são, tipicamente, muito esparsos, o cálculo é computacionalmente eficiente.

O classificador binário independente é apenas uma das variantes da aproximação naïve de Bayes. Lewis [1998] apresenta as diversas variações deste algoritmo básico. Existem aproximações que:

- relaxam a restrição de vectores binários. Estes classificadores utilizam variadas distribuições estatísticas de variáveis inteiras, tais como distribuições de Poisson

e a binomial negativa, para considerar a frequência dos termos [Fuhr 1989; Fuhr et al. 1991; Salton and Buckley 1988].

- introduzem a normalização do tamanho do documento. Esta normalização é importante já que o valor de $\log \frac{P(c_i|\vec{d_j})}{1-P(c_i|\vec{d_j})}$ tende a ser mais extremo (muito alto ou muito baixo) para grandes documentos (nos quais $w_{kj} = 1$ para muitos valores de k), independentemente da sua ligação semântica com c_i. Ao utilizar um modelo multinomial (no qual os eventos são termos e não documentos) o classificador considera naturalmente o tamanho do documento [Baker and McCallum 1998; Chakrabarti et al. 1998; Guthrie et al. 1994; Joachims 2002; McCallum et al. 1998]. Por outro lado, ocorrências do mesmo termo no mesmo documento são consideradas independentes – uma suposição ainda menos plausível que aquela que considera termos diferentes independentes um do outro.

- relaxam a suposição de independência. Estes classificadores são caracterizados pela estimação dos parâmetros ser mais difícil e por um custo computacional muito mais elevado [Koller and Sahami 1997].

O algoritmo naïve de Bayes implementado na ferramenta WEKA denomina-se Flexible Bayes [John and Langley 1995] e permite, além de atributos discretos, a utilização de atributos contínuos. Este algoritmo, além de utilizar a aproximação mais comum para a estimação dos parâmetros de cada tipo de atributos (para os discretos, estima as probabilidades; para os contínuos, assume uma distribuição normal dos atributos estimando a média e desvio padrão) permite, para os contínuos, a utilização de um método de núcleo (ver Capítulo 5) que aproxima distribuições mais complexas que a distribuição normal.

4.2.2 Árvores de decisão

Os métodos de indução probabilística são, por natureza, quantitativos e, como tal, têm sido algumas vezes criticados por não serem facilmente interpretados pelas pessoas. Uma classe de algoritmos que não sofre deste problema são os algoritmos simbólicos, entre os quais, as árvores e regras de decisão são os exemplos mais importantes.

Um classificador de texto, representado por uma **árvore de decisão** é uma árvore, na qual os nós internos representam termos, os ramos representam os valores que os termos podem ter e as folhas representam classes. Estes algoritmos, classificam um documento d_j verificando, recursivamente (a partir da raíz), o valor do termo indicado no nó e percorrendo o ramo que possui o valor do termo em d_j até chegar a uma folha; o documento pertence à classe etiquetada na folha. Se for utilizada uma representação binária do documento, as árvores são binárias; se os termos forem atributos contínuos

os testes são feitos sobre intervalos de valores. A Figura 4.1 ilustra uma árvore de decisão.

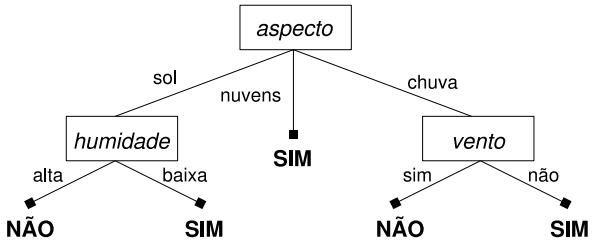

Figura 4.1: Uma árvore de decisão; os nós representam atributos, os ramos, os valores possíveis, e as folhas, a classe.

A indução de uma árvore de decisão para a classe c_i é feita aplicando uma estratégia de divisão e conquista[7]: verifica se todos os exemplos do conjunto de treino têm a mesma etiqueta (c_i ou $\bar{c}_i$); em caso negativo, selecciona um termo t_k, dividindo o conjunto de treino Tr em subconjuntos de documentos com o mesmo valor para t_k e colocando cada subconjunto numa subárvore distinta. Este processo é repetido recursivamente para cada uma das subárvores até que todas as folhas só contenham exemplos pertencentes à mesma classe, que é então escolhida como etiqueta da folha.

O passo chave do algoritmo consiste na escolha do termo t_k sobre o qual é feita a partição dos documentos. Esta escolha é, geralmente, feita de acordo com um critério de informação mútua ou entropia.

Uma árvore construída desta forma está sujeita a sobre-ajustamento, já que alguns ramos podem ser excessivamente específicos para o conjunto de treino. Assim sendo, a maioria dos algoritmos de indução de árvores de decisão incluem um método para fazer crescer a árvore e outro para a podar[8] (remover ramos demasiado específicos), onde a capacidade para classificar correctamente todos os exemplos de treino é trocada por uma maior generalidade. Existem diversas variações deste esquema básico de indução de árvores de decisão (ver, por exemplo [Mitchell 1997]).

A maioria dos trabalhos de CT utiliza algoritmos padrão para indução de árvores de decisão. Os mais populares são o ID3 (os atributos são nominais) utilizado em [Fuhr et al. 1991], o C4.5 (uma extensão do ID3, que, além de outros melhoramentos, permite atributos nominais e contínuos) utilizado em [Cohen and Hirsh 1998; Cohen and Singer 1999; Joachims 1998; Lewis and Gale 1994] e o C5 (versão comercial do C4.5) utilizado em [Li and Jain 1998]. Dumais et al. [1998], Lewis and Ringuette [1994] e Weiss et al. [1999] utilizam algoritmos de indução de árvores de decisão mais específicos.

Além de serem estudadas como ferramenta de classificação principal, as árvores de

[7]Do inglês, "*divide and conquer*".
[8]Do inglês, *prunning*.

decisão têm sido utilizadas como classificador de comparação[9] [Cohen and Singer 1999; Joachims 1998] e como membro de comités de classificadores [Li and Jain 1998; Schapire and Singer 2000; Schapire et al. 1998; Weiss et al. 1999] (ver secção 4.2.9).

A ferramenta WEKA implementa o algoritmo C4.5 [Quinlan 1993].

4.2.3 Regras de decisão

Um classificador para a classe c_i que utiliza **regras de decisão** é uma regra na forma normal disjuntiva (FND), ou seja, fórmulas condicionais (cláusulas) do tipo

se $\langle fórmula\ Booleana\ FND \rangle$ **então** c_i **senão** $\bar{c}_i$

Os antecedentes denotam a presença ou ausência de termos no documento e o consequente denota a decisão de classificá-lo ou não na classe c_i. As regras de decisão são similares às árvores de decisão do ponto de vista de poderem codificar qualquer função Booleana. No entanto, uma das suas vantagens é tenderem gerar regras mais compactas.

Os métodos de indução de regras tentam seleccionar, do conjunto de regras de cobertura (aquelas regras que classificam correctamente todos os exemplos de treino) a "melhor" de acordo com um **critério de minimização**. Enquanto as árvores de decisão são geralmente induzidas utilizando uma estratégia divisão-e-conquista do geral para o particular, as regras FND são, muitas vezes, induzidas do particular para o geral.

No início da indução do classificador para c_i, cada exemplo de treino é visto como uma fórmula $\eta_1, \ldots, \eta_n \rightarrow \gamma_i$, onde $\eta_1, \ldots, \eta_n$ são os termos do documento e γ_i é igual a c_i ou $\bar{c}_i$ conforme o documento seja um exemplo positivo ou negativo de c_i. Este conjunto de cláusulas é já um classificador FND para c_i mas está, certamente, sobre-ajustado. O algoritmo de indução emprega então um processo de generalização na qual a regra é simplificada através de um conjunto de operações (por exemplo, removendo premissas das cláusulas ou juntando cláusulas) que maximizam a sua compactação sem afectar a propriedade de cobertura do algoritmo. No fim deste processo, é aplicado um algoritmo de poda, à semelhança das árvores de decisão.

Os algoritmos de indução de regras variam muito em termos de métodos, heurísticas e critérios empregues na generalização e poda. CHARADE [Moulinier and Ganascia 1996], DL-ESC [Li and Yamanishi 1999], RIPPER [Cohen 1995; Cohen and Hirsh 1998; Cohen and Singer 1999], SCAR [Moulinier and Ganascia 1996] e SWAP-1 [Apté et al. 1994] são alguns algoritmos utilizados em CT.

[9] Do inglês, *baseline classifier*.

Enquanto estes métodos utilizam regras da lógica proposicional, existem outros que induzem regras de lógica de primeira ordem. Estes, designam-se como algoritmos de **programação lógica indutiva**[10]. Cohen [1995] comparou estes dois tipos de indução de regras numa aplicação de CT (utilizando o RIPPER e a sua versão de lógica de primeira ordem, o FLIPPER), concluindo que o poder adicional da lógica de primeira ordem apenas produz benefícios modestos.

4.2.4 Métodos *online*

Um **classificador linear** para a classe c_i é uma representação de c_i em termos de um vector $\vec{c_i} = \langle w_{1i}, \ldots, w_{|\mathcal{T}|i} \rangle$ pertencente ao mesmo espaço a $|\mathcal{T}|$-dimensões dos documentos, onde $VC_i(d_j)$ corresponde ao produto interno $\sum_{k=1}^{|\mathcal{T}|} w_{ki} w_{kj}$ entre $\vec{d_j}$ e $\vec{c_i}$. Ao aplicar a normalização co-seno (ver Anexo C.3), o produto interno entre os dois vectores corresponde à semelhança co-seno

$$S(c_i, d_j) = cos(\alpha) = \frac{\sum_{k=1}^{|\mathcal{T}|} w_{ki} \cdot w_{kj}}{\sqrt{\sum_{k=1}^{|\mathcal{T}|} w_{ki}^2} \cdot \sqrt{\sum_{k=1}^{|\mathcal{T}|} w_{kj}^2}}$$

que representa o co-seno do ângulo α que separa os dois vectores.

Os métodos de indução de classificadores lineares são muitas vezes divididos em duas classes: métodos *batch* e métodos *online*. Os métodos *batch* constroem o classificador analisando todo o conjunto de treino de uma só vez; os métodos *online* constroem o classificador assim que observam o primeiro documento de treino refinando-o, incrementalmente, à medida que forem examinando novos documentos.

Um exemplo simples de um método *online* é o **perceptrão**[11], proposto para aplicações de CT por Schütze et al. [1995] e Wiener et al. [1995] e utilizado depois por Dagan et al. [1997] e Ng et al. [1997]. Neste algoritmo, o classificador para c_i é inicializado fixando todos os pesos w_{ki} com o mesmo valor positivo. Quando um exemplo de treino d_j (representado pelo vector $\vec{d_j}$ de pesos binários) é examinado, o classificador construído até ao momento classifica-o e o resultado é examinado. Se estiver correcto nada é feito, mas se estiver errado, os pesos do classificador são modificados: se d_j for um exemplo positivo de c_i, então os pesos w_{ki} dos termos activos (os termos t_k para os quais $w_{kj} = 1$) são promovidos incrementando-os de um valor fixo $\alpha > 0$ (chamado **coeficiente de aprendizagem**[12]); se d_j for um exemplo negativo de c_i, então os mesmos pesos são despromovidos decrementando-os do mesmo valor α. É de notar que, quando o classificador atinge um nível de desempenho razoável, o facto de um

[10] Do inglês, *inductive logic programming*.
[11] Do inglês, *perceptron*.
[12] Do inglês, *learning rate*.

peso w_{ki} ser muito baixo significa que t_k contribui negativamente para o processo de classificação e, como tal, pode ser eliminado da representação.

O perceptrão é um algoritmo aditivo de actualização de pesos[13]. Os algoritmos Positive Winnow [Dagan et al. 1997] e Ballanced Winnow [Dagan et al. 1997; Ragas and Koster 1998] são variantes multiplicativas deste algoritmo, que possuem dois coeficientes de aprendizagem para promover e despromover os pesos ($\alpha_1 > 1$ e $0 < \alpha_2 < 1$). O algoritmo Ballanced Winnow possui, ainda, dois pesos w_{ki}^+ e w_{ki}^- para cada termo t_k.

Outros exemplos de métodos *online* para construção de classificadores de texto são o Widrow-Hoff e a sua refinação chamado Exponentiated Gradient, ambos utilizados em [Lewis et al. 1996] e o algoritmo Sleeping Experts [Cohen and Singer 1999; Ragas and Koster 1998], uma versão do algoritmo Ballanced Winnow.

4.2.5 Redes neuronais

Uma **rede neuronal** é uma rede de unidades básicas, onde as unidades de entrada representam normalmente os termos, as unidades de saída representam as classes e os pesos dos arcos que ligam as unidades entre diferentes níveis representam as relações de dependência entre as unidades. Para classificar um documento d_j, colocam-se nas unidades de entrada os pesos w_{kj} dos termos; a sua activação é propagada através da rede e o valor das unidades de saída determina a decisão da classe. Uma forma tradicional de treinar uma rede neuronal utiliza um algoritmo de retro-propagação[14] do erro, que altera os parâmetros da rede quando um exemplo de treino é mal classificado, de maneira eliminar ou minimizar esse erro.

O perceptrão é o tipo de classificador neuronal mais simples [Dagan et al. 1997; Ng et al. 1997]. Outros classificadores lineares neuronais que implementam um tipo de regressão logística foram propostos e testados por Schütze et al. [1995] e Wiener et al. [1995] com muito bons resultados de desempenho.

Uma rede neuronal não linear é uma rede em que existem mais "camadas" de unidades, o que permite à rede aprender interacções entre termos de nível superior [Lam et al. 1999; Ruiz and Srinivasan 1999; Schütze et al. 1995; Weigend et al. 1999; Wiener et al. 1995; Yang and Liu 1999]. Os estudos comparativos entre redes não lineares e as redes lineares correspondentes revelam que se obtiveram nos primeiros melhoramentos muito pequenos [Wiener et al. 1995] ou nulos [Schütze et al. 1995].

[13] Do inglês, *additive weight-updating algorithm*.
[14] Do inglês, *backpropagation*.

4.2.6 Método de Rocchio

Um classificador linear **baseado-em-perfis**[15] para a classe c_i, utiliza um **perfil** (ou documento protótipo) extraído a partir do conjunto de treino para representar c_i. Estes classificadores têm vantagens óbvias em termos de interpretabilidade, já que o perfil é mais facilmente compreendido por um humano do que, por exemplo, uma rede neuronal. A indução de um classificador linear é, muitas vezes, precedido por uma redução do espaço de termos; neste caso, um perfil para c_i é uma lista de termos pesados cuja presença ou ausência é a mais adequada para discriminar a classe.

O **método de Rocchio** é utilizado para induzir classificadores lineares baseados em perfis. Utiliza uma adaptação para CT da fórmula de Rocchio da reacção de relevância[16] no modelo de espaço de vectores. É, talvez, o único método de CT cujas raízes residem exclusivamente na área de RI e não na área de AA. Esta adaptação foi proposta por Hull [1994] e a partir daí tem sido utilizada em muitos trabalhos, como objecto de investigação [Ittner et al. 1995; Joachims 1997; Moschitti 2003a; Ragas and Koster 1998; Sable and Hatzivassiloglou 2000; Schapire et al. 1998; Singhal et al. 1997], como classificador de comparação [Cohen and Singer 1999; Galavotti et al. 2000; Joachims 1998; Lewis et al. 1996; Schapire and Singer 2000; Schütze et al. 1995] ou como membro de um comité de classificadores [Larkey and Croft 1996] (ver secção 4.2.9).

O método de Rocchio calcula o classificador $\vec{c_i} = \langle w_{1i}, \ldots, w_{|\mathcal{T}|i} \rangle$ para a classe c_i através da fórmula

$$w_{ki} = \beta \cdot \sum_{\{d_j \in POS_i\}} \frac{w_{kj}}{|POS_i|} - \gamma \cdot \sum_{\{d_j \in NEG_i\}} \frac{w_{kj}}{|NEG_i|}$$

onde w_{kj} é o peso do termo t_k no documento d_j, $POS_i = \{d_j \in Tr | \breve{\phi}(d_j, c_i) = V\}$ e $NEG_i = \{d_j \in Tr | \breve{\phi}(d_j, c_i) = F\}$. Nesta fórmula, β e γ são parâmetros de controlo que permitem dar importâncias relativas aos exemplos positivos e negativos. Por exemplo, se $\beta = 1$ e $\gamma = 0$ (como utilizado em [Dumais et al. 1998; Hull 1994; Joachims 1998; Schütze et al. 1995]), o perfil de c_i é o centróide dos seus exemplos de treino positivos. De uma forma geral, o classificador de Rocchio recompensa a proximidade de um documento de teste ao centróide dos exemplos de treino positivos e a sua distância ao centróide dos exemplos de treino negativos.

Este método é de fácil implementação e também muito eficiente, uma vez que consiste, basicamente, em calcular médias de pesos. Por outro lado, o seu desempenho pode ser fraco se os documentos de uma classe ocorrerem em grupos separados, já que o seu centróide pode estar longe de qualquer dos grupos, como retratado na Figura 4.2.

[15] Do inglês, *profile based*.
[16] Do inglês, *relevance feedback*.

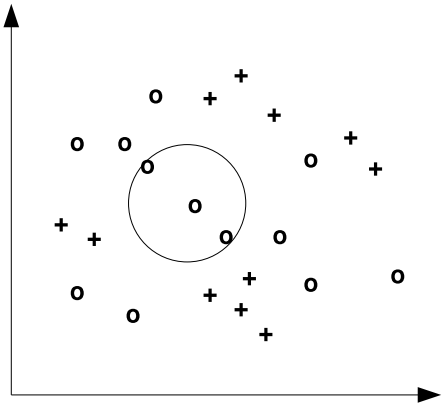

Figura 4.2: O classificador de Rocchio quando os exemplos positivos ocorrem em grupos separados. Os círculos representam os exemplos positivos e as cruzes os negativos. O círculo maior representa a área de influência do classificador.

Alguns trabalhos introduziram alterações ao método original modificando a forma como é extraído o perfil da classe: o conjunto NEG_i pode ser substituído pelo subconjunto $QPOS_i$ dos quase-positivos, definido como "os mais positivos, entre todos os negativos" [Schapire et al. 1998]. O subconjunto $QPOS_i$ é obtido verificando quais, do conjunto de exemplos negativos, são mais próximos do centróide dos exemplos positivos.

Este algoritmo, que pode ser considerado um método de zona de inquirição[17] [Singhal et al. 1997], quando utilizado em conjunto com métodos de ponderação mais eficazes, originou desempenhos comparáveis com aqueles obtidos com o método de estimulação (secção 4.2.9), um método considerado estado-da-arte para a classificação de texto [Sebastiani 2002]. Embora possua uma eficiência inferior ao algoritmo original, os seus tempos de treino são 60 vezes mais rápidos o método de estimulação [Schapire et al. 1998].

O classificador de Rocchio parametrizado[18] [Moschitti 2003a], que utiliza um modelo para selecção automática dos parâmetros, obteve desempenhos equiparáveis ao método de Schapire et al. [1998], mantendo a complexidade temporal do algoritmo original.

4.2.7 Classificadores baseados em exemplos

Um classificador baseado em exemplos não constrói uma representação explícita da classe, mas depende das etiquetas dos documentos de treino similares ao documento de teste. Estes métodos são também conhecidos por **aprendizes preguiçosos**[19] uma vez que atrasam a generalização dos dados de treino até aparecer um exemplo de teste.

[17] Do inglês, *query zoning*.
[18] Do inglês, *parameterized Rocchio classifier*.
[19] Do inglês, *lazy learners*.

As primeiras experiências na área de CT com algoritmos baseados em exemplos foram de realizadas por Creecy et al. [1992] e Masand et al. [1992]; outros exemplos incluem [Joachims 1998; Lam et al. 1999; Larkey 1998, 1999; Li and Jain 1998; Yang and Liu 1999; Yang and Pedersen 1997].

O algoritmo **k-vizinhos-mais-próximos**[20] é uma das implementações mais conhecidas de classificadores baseados em exemplos. Para decidir se d_j deve ser classificado em c_i, o algoritmo verifica se os k documentos de treino mais parecidos com d_j foram classificados em c_i. Em caso afirmativo para uma maioria suficientemente grande dos exemplos, decide-se por uma classificação positiva, senão decide-se negativamente.

Yang and Chute [1994] utilizam uma versão deste algoritmo pesada pela distância: o facto de um documento mais-próximo pertencer a c_i é pesado pela sua semelhança com o exemplo de teste. A classificação de um documento resume-se ao cálculo de

$$VC_i(d_j) = \sum_{d_z \in Tr_k(d_j)} RS(d_j, d_z) \cdot \|\breve{\phi}(d_z, c_i)\|$$

onde $RS(d_j, d_z)$ é uma medida de relacionamento semântico entre o documento de teste d_j e o documento de treino d_z, $Tr_k(d_j)$ é conjunto dos k documentos d_z que maximizam $RS(d_j, d_z)$ e

$$\|\alpha\| = \begin{cases} 1 & \text{se } \alpha = V \\ 0 & \text{se } \alpha = F \end{cases}$$

Larkey and Croft [1996] utilizam uma medida probabilística e Yang and Chute [1994] uma medida baseada em vectores para medir o relacionamento semântico dos documentos.

A construção de um classificador k-vizinhos-mais-próximos involve também a determinação do valor de k que indica quantos documentos de treino devem ser considerados para calcular $VC_i(d_j)$; k é normalmente determinado experimentalmente através de um conjunto de validação.

É de notar que este algoritmo, ao contrário dos classificadores lineares não subdivide o espaço de documentos de forma linear e, como tal, não sofre do problema mencionado no final da secção 4.2.6.

Existem outras técnicas de vizinhos-mais-próximos utilizadas em CT. O sistema WHIRL [Cohen and Hirsh 1998] estende a tecnologia das bases de dados relacionais com "junções baseadas em semelhança"[21] e Galavotti et al. [2000] redefine $\|\alpha\|$ de maneira a pesar negativamente os documentos d_z similares a d_j que não pertencem a c_i. Lam

[20] Do inglês, *k-nearest neighbours*.
[21] Do inglês, *similarity-based soft joins*.

and Lo [1998] apresentam uma combinação dos métodos baseados em exemplos e de perfis utilizando instâncias generalizadas[22] em vez de documentos de treino.

4.2.8 Máquinas de vectores de suporte

O método de **máquinas de vectores de suporte**[23] foi introduzido em CT por Joachims [1998, 1999] e depois utilizado por Drucker et al. [1999], Dumais et al. [1998], Dumais and Chen [2000], Klinkenberg and Joachims [2000], Taira and Haruno [1999] e Yang and Liu [1999]. Em termos geométricos, pode ser visto como uma tentativa de encontrar, entre todas as superfícies $\tau_1, \tau_2, \ldots$ do espaço a $|\mathcal{T}|$-dimensões que separam os exemplos positivos dos negativos (denominadas **superfícies de decisão**[24]), a superfície τ_i que tem a margem mais larga, ou seja, aquela cuja propriedade de separação é invariante com respeito à mais larga translação possível de τ_i.

Este conceito é de mais fácil percepção quando os exemplos positivos e negativos são linearmente separáveis. Neste caso, as superfícies de decisão são hiperplanos a $(|\mathcal{T}|-1)$-dimensões. Num problema a 2-dimensões, como o caso da Figura 4.3, várias linhas podem ser escolhidas como superfícies de decisão; este método escolhe o elemento do meio do conjunto mais "largo" de linhas paralelas, definido como o conjunto no qual a distância máxima entre dois elementos do conjunto é a maior. É de notar que esta superfície de decisão é determinada apenas por um pequeno conjunto de exemplos de treino, denominados **vectores de suporte**. Este método também é aplicável quando os exemplos positivos e negativos não são linearmente separáveis (ver secção 5.3).

A ferramenta WEKA implementa o algoritmo de optimização sequencial mínima[25] [Platt 1999] para treinar o classificador de vectores de suporte.

4.2.9 Comités de classificadores

Este método baseia-se na ideia de que k peritos, se combinados apropriadamente, podem produzir um melhor desempenho que as suas opiniões individuais. Na CT, a ideia é aplicar k classificadores diferentes $\phi_1, \ldots, \phi_k$ à mesma tarefa de decidir se o documento d_j deve ser classificado em c_i e, depois, combinar os seus resultados de forma apropriada. Estes classificadores são, então, caracterizados pela escolha de (i) k classificadores e (ii) uma função de combinação.

Para obter um bom desempenho, os classificadores que formam o comité devem ser

[22] Do inglês, *generalized instances*.
[23] Do inglês, *support vector machine (SVM)*.
[24] Do inglês, *decision surfaces*.
[25] Do inglês, *sequential minimal optimization algorithm*.

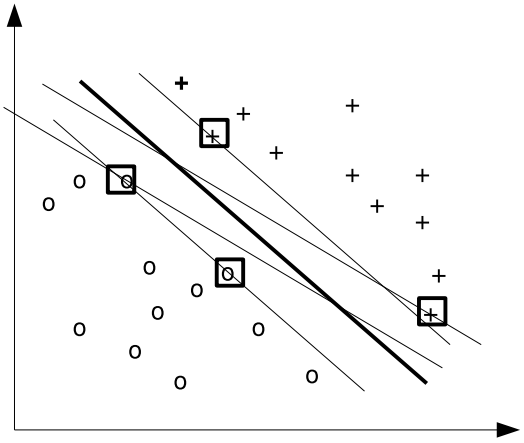

Figura 4.3: A indução de classificadores de vectores de suporte.

o mais independentes possível: podem ser diferentes na representação do documento, do método de indução utilizado ou ambos. Em CT, a maioria dos trabalhos utiliza diferentes algoritmos de indução sobre a mesma representação dos documentos.

Para combinar os resultados, existem diversas aproximações. A mais simples é a **maioria-de-votos**[26], que considera as opiniões binárias dos k classificadores e decide por aquela que obtiver pelo menos $\frac{k+1}{2}$ votos [Li and Jain 1998; Liere and Tadepalli 1997]. Outra regra consiste na **combinação linear pesada**[27] onde a soma pesada dos valores individuais de VC_i de cada classificador constitui o valor final de VC_i. A **selecção de classificador dinâmica**[28] e a **combinação adaptativa de classificadores**[29] são variações da combinação linear pesada que consideram os l exemplos de validação mais semelhantes a d_j [Li and Jain 1998].

Os resultados obtidos em CT não são conclusivos. Em [Larkey 1998], os comités obtiveram melhores desempenhos que o melhor classificador individual, mas em [Li and Jain 1998] este resultado apenas foi obtido (com uma pequena margem) utilizando a técnica da combinação adaptativa de classificadores.

No **método de estimulação**[30] [Schapire and Singer 2000; Schapire et al. 1998], os k classificadores $\phi_1, \ldots, \phi_k$ que formam o comité são obtidos através do mesmo método de indução (chamado **aprendiz fraco**[31]). Nesta técnica, os k classificadores são treinados sequencialmente e não de forma independente como nos comités descritos anteriormente. Desta forma, o classificador ϕ_i pode ter em consideração o desempenho dos

[26] Do inglês, *majority voting*.
[27] Do inglês, *weighted linear combination*.
[28] Do inglês, *dynamic classifier combination*.
[29] Do inglês, *adaptive classifier combination*.
[30] Do inglês, *boosting*.
[31] Do inglês, *weak learner*.

$\phi_1, \ldots, \phi_{i-1}$ classificadores, concentrando-se em classificar correctamente os exemplos mal classificados por eles; o classificador final é obtido através de uma combinação linear pesada dos k classificadores assim construídos. Testes realizados em [Schapire and Singer 2000; Schapire et al. 1998] indicam que o AdaBoost, um algoritmo de estimulação direccionado para maximizar a micro-médias que e que utiliza árvores de decisão de um nível, possui um desempenho equivalente aos melhores desempenhos obtidos Sebastiani [2002]. Sebastiani et al. [2000] apresenta um algoritmo baseado em comités de subcomités de classificadores que melhora o desempenho e, especialmente, a eficiência do algoritmo AdaBoost.

Li and Jain [1998], Myers et al. [2000] e Weiss et al. [1999], que utilizam outras aproximações baseadas em métodos de estimulação, conseguiram desempenhos equivalentes.

4.2.10 Outros métodos

Entre outros, existem ainda trabalhos em CT que utilizam aproximações baseadas em modelos de regressão [Fuhr and Pfeifer 1994; Ittner et al. 1995; Lewis and Gale 1994; Schütze et al. 1995; Yang and Chute 1994], redes de inferência Bayesianas[32] [Dumais et al. 1998; Lam et al. 1997; Tzeras and Hartmann 1993], algoritmos genéticos [Clack et al. 1997; Masand 1994] e modelação da entropia máxima[33] [Manning and Schütze 1999].

4.3 Desempenho dos classificadores

Para testar os algoritmos de CT é costume utilizar alguns corpora disponíveis publicamente. Os mais comuns, qualquer deles escritos na língua Inglesa, são o `Reuters`, `Ohsumed` e `20 Newsgroups`. Entre eles, o corpus `Reuters` é o mais utilizado, sendo constituído por um conjunto de notícias classificadas em classes relacionadas com economia. Apresenta diversas versões, mas a `Reuters-21578`[34] é considerada a versão padrão. No corpus `Ohsumed` (construído por Hersh et al. [1994]) os documentos são títulos ou títulos com resumo de revistas médicas e as classes são termos do *thesaurus* MeSH e no corpus `20 Newsgroups` (construído por Lang [1995]) os documentos são mensagens colocadas no grupo de notícias Usenet e as classes são os próprios grupos.

Os resultados experimentais publicados na literatura permitem fazer algumas considerações sobre desempenho comparativo dos métodos de CT apresentados. No entanto, convém referir que as comparações tendem a ser de confiança só quando as experiências

[32] Do inglês, *Bayesian inference networks*.
[33] Do inglês, *maximum entropy modelling*.
[34] Disponível em `http://www.daviddlewis.com/resources/testcollections/reuters21578`.

são realizadas pelo mesmo autor em condições cuidadosamente controladas. Se foram realizadas por diferentes autores, algumas escolhas, muitas vezes exteriores ao algoritmo (por exemplo, diferentes escolhas de pré-processamento, redução de dimensão ou valores dos parâmetros dos algoritmos) podem influenciar os resultados experimentais.

A maioria os trabalhos referidos nesta secção utilizam o corpus `Reuters` na experimentação. Como tal, as considerações apontadas não são absolutas e finais, já que um corpus particular pode ter características muito diferentes daquelas existentes na colecção `Reuters` e classificadores diferentes podem responder de forma diferente a essas características. A língua na qual estão escritos os documentos constitui uma característica muito particular que pode influenciar o desempenho do classificador, sendo esse estudo um dos objectivos deste trabalho.

4.3.1 Unidades de indexação

Várias experiências [Baker and McCallum 1998; Dumais et al. 1998; Lewis 1992a; Tzeras and Hartmann 1993] reportam que a utilização de representações mais sofisticadas do que a palavra não revelam melhores desempenhos. Lewis [1992a] argumenta que embora as representações baseadas em sintagmas tenham qualidades semânticas superiores, possuem qualidades estatísticas inferiores às representações que utilizam a palavra, já que possuem mais termos, mais sintagmas sinónimos ou quase sinónimos, menor consistência de afectação (por não se associarem termos sinónimos em diferentes documentos) e menor frequência dos termos nos documentos.

Basili et al. [1999] e Basili et al. [2004] referem que o processamento de língua natural pode ser utilizado para melhorar a classificação de texto, mas os resultados não são impressionantes: a informação sintáctica parece apenas melhorar o desempenho de algoritmos de classificação de texto débeis (algoritmos que geram desempenhos inferiores) e a informação semântica produz apenas desempenhos timidamente melhores.

Ao aplicar a representação de sequências não contíguas de k caracteres a uma colecção com poucos documentos, Lodhi et al. [2002] obtiveram resultados comparáveis à representação saco-de-palavras; no entanto, com colecções maiores, esta semelhança já não foi visível: o nível palavra suplantou a sua representação multi-palavra. Uma justificação possível indicada é esta representação fornecer ligações semânticas entre palavras que a representação saco-de-palavras vê como distintas; em colecções maiores, onde existe informação suficiente para apreender a relevância das palavras, estas ligações já não se revelam tão importantes.

Outros trabalhos, como por exemplo [Fürnkranz et al. 1998; Mladenić and Grobelnik 1998; Tan et al. 2002], reportam que a utilização de termos de indexação mais complexos que a palavra originam melhores classificadores. No entanto, essas melhorias são

mínimas, foram realizadas com poucos documentos, classificadores débeis ou tarefas específicas o que não permite afirmar que a utilização de informação linguística é benéfica para o processo de classificação de documentos.

4.3.2 Algoritmos

Sebastiani [2002] resume os resultados experimentais publicados sobre diferentes métodos de CT em várias versões do corpus `Reuters` e apresenta algumas considerações sobre o seu desempenho comparativo:

- Os classificadores baseados em métodos de estimulação, máquinas de vectores de suporte, métodos baseados em exemplos e métodos de regressão apresentam os melhores desempenhos;
- As redes neuronais e os classificadores lineares *online* mostram bons resultados, embora um pouco inferiores que os métodos atrás mencionados;
- O método de Rocchio e os classificadores naïve de Bayes revelam desempenhos inferiores. No entanto, as variações ao algoritmo de Rocchio original introduzidas por Schapire et al. [1998] (utilização de quase-positivos) e Moschitti [2003a] (classificador de Rocchio parametrizado) posicionam o algoritmo entre os melhores;
- É difícil concluir algo sobre árvores de decisão. Embora tenha sido referido que as árvores de decisão não apresentam bons resultados [Cohen and Singer 1999; Joachims 1998; Lewis and Gale 1994; Lewis and Ringuette 1994], Dumais et al. [1998] apresenta uma versão de árvores de decisão com desempenhos quase idênticos ao sistema com melhor desempenho (máquina de vectores de suporte).

4.3.3 Testes de significância estatística

A maioria dos trabalhos referidos não valida os resultados obtidos através de testes de significância estatística. Estes testes são úteis para verificar até que ponto os resultados experimentais suportam a afirmação de que um classificador ϕ' é melhor que outro classificador ϕ'' ou para verificar o quanto uma diferença na construção da experiência afecta o desempenho medido num sistema ϕ.

Hull [1994] e Schütze et al. [1995] foram dos primeiros a validar os seus resultados utilizando testes ANOVA e de Friedam. Yang and Liu [1999] definem um conjunto completo de testes de significância, alguns para aplicar a micro-médias e outros a macro-médias de medidas de desempenho, aplicando-os sistematicamente na comparação entre cinco classificadores distintos. Outros exemplos de testes de significância em CT foram

aplicados em [Cohen 1995; Cohen and Hirsh 1998; Joachims 1997; Koller and Sahami 1997; Lewis et al. 1996; Wiener et al. 1995].

4.4 Classificação de textos em língua Portuguesa

Nos últimos anos outros investigadores têm desenvolvido trabalhos sobre classificação de textos escritos na língua Portuguesa. De seguida referem-se sucintamente alguns deles.

Langie [2004] e Moraes and Lima [2007] utilizam o algoritmo dos k-vizinhos-mais-próximos num processo de classificação hierárquica de documentos com uma representação saco-de-palavras. Ambos utilizam a mesma hierarquia de classes desenvolvida por Langie mas enquanto o primeiro utiliza um corpus com relativamente poucos documentos (2896 textos jornalísticos), o corpus do último tem 26606 textos do corpus PLN-BR CATEG.

Oliveira et al. [2007] utilizam redes neuronais com uma representação saco-de-palavras para classificar um conjunto de 3281 relatórios de objectivos de empresas brasileiras em 764 classes, um subconjunto das actividades económicas reconhecidas pela lei Brasileira.

A informação morfo-sintáctica gerada pelo PALAVRAS é utilizada por Silva and Vieira [2007] na selecção de termos de indexação de documentos da Folha de S. Paulo (o corpus `Folha` descrito em 7.2.2). Utilizando uma representação saco-de-palavras comparam algoritmos de árvores de decisão e máquina de vectores de suporte e referem que, ao contrário do primeiro, na máquina de vectores de suporte o aumento do número de termos tem uma influência de contínua redução dos erros.

Por outro lado, Camargo [2007] utiliza o mesmo tipo de informação morfo-sintáctica para comparar os algoritmos naïve de Bayes e máquina de vectores de suporte, aplicando estes algoritmos de aprendizagem aos corpora `Folha` e `Teses`. Este último corpus é composto por 475 documentos com o resumo de teses de pós-graduação que estão distribuidos por 5 classes. Os resultados obtidos indicam que o algoritmo naïve de Bayes gera classificadores com percentagem de erros mais baixas, pelo menos em representações com um pequeno número de termos; no entanto não foram feitos testes de significância entre as diferenças.

Rigo et al. [2007] classificam documentos relacionados com linguagens de programação orientada por objectos, utilizando informação linguística descrita em ontologias do domínio. Esta ontologia foi criada manualmente a partir de conceitos sobre a estrutura dos documentos e do próprio domínio; foram identificados os termos mais frequentes, além de sinónimos e possíveis hierarquias entre conceitos. Com a utilização de heurísticas e da ontologia são identificadas as construções relevantes que permitem a

classificação de documentos nesse domínio. Para a classificação foram utilizados 46 documentos: 37 documentos jornalísticos da Folha de S. Paulo, 4 de um livro didáctico de ciências e 2 tutoriais sobre linguagens de programação orientada por objectos.

Aires et al. [2004] classificam páginas Web escritas em Português do Brasil num conjunto de 7 classes. Estas correspondem a intenções de pesquisa do utilizador durante a navegação (por exemplo a pesquisa de uma definição, um estudo ou uma notícia sobre determinado assunto). Utiliza as árvores de decisão da ferramenta WEKA e baseia-se em 46 características estilísticas dos documentos; o corpus é composto por 511 documentos distribuidos equitativamente pelas 7 classes.

Por outro lado, Pavelec et al. [2006] utilizam atributos estilométricos da língua Portuguesa para identificar a autoria de documentos digitais. A abordagem utiliza máquinas de vectores de suporte aplicada a um corpus de 150 textos jornalísticos escritos por 10 autores. A extracção dos atributos estilométricos baseia-se na análise sintáctica, utilizando ocorrências normalizadas de conjunções.

Capítulo 5

Núcleos para classificação de texto

Em CT, e como referido na secção 4.3.2, os melhores desempenhos têm sido obtidos pelos classificadores baseados em métodos de estimulação, máquinas de vectores de suporte, métodos baseados em exemplos e métodos de regressão. Algumas variações do método de Rocchio e de árvores de decisão também apresentaram resultados equivalentes.

Neste estudo escolheu-se, como algoritmo de classificação, a máquina de vectores de suporte (MVS). Este algoritmo insere-se num conjunto mais vasto de métodos de análise de padrões[1] – os **métodos de núcleo**[2]. Esta aproximação coloca os dados num espaço de características[3] apropriado e utiliza algoritmos baseados em álgebra linear, geometria e estatística para descobrir padrões nos dados transformados. Este espaço é especificado de forma natural mas indirecta, através de uma **função de núcleo**[4].

A escolha deste algoritmo justifica-se pelos seguintes factos:

- os métodos de regressão e de estimulação são pouco eficientes na fase de construção do classificador;
- a construção de uma árvore de decisão embora mais rápida que os métodos anteriores é muito mais lenta que um classificador baseado em máquina de vectores de suporte;
- os métodos baseados em exemplos são pouco eficientes na fase de classificação;
- o método de Rocchio (modificado) embora possua uma eficiência computacional

[1] Do inglês, *pattern analysis methods*.
[2] Do inglês, *kernel methods*.
[3] Do inglês, *feature space*.
[4] Do inglês, *kernel function*.

equiparada à MVS, é considerado um método com fracas bases teóricas[5] [Joachims 2002; Moschitti 2003b].

Mais ainda, por constituir um método de núcleo, a utilização de exemplos estruturados é mais simples já que não exige alterações ao funcionamento do algoritmo.

Este Capítulo começa por apresentar os métodos de núcleo, enumerando, de seguida, as propriedades teóricas que caracterizam as funções de núcleo e os espaços de características induzidos. A secção 5.3 introduz as máquinas de vectores de suporte e a secção 5.4 mostra como o modelo do espaço de vectores desenvolvido pela comunidade de RI pode ser reinterpretado por funções de núcleo e apresenta núcleos para dados estruturados.

Os conceitos básicos sobre álgebra linear são apresentados no Anexo D; as demonstrações dos teoremas e propriedades relativos aos núcleos encontram-se em [Shawe-Taylor and Cristianini 2004, Capítulo 3].

5.1 Métodos de núcleo

A estratégia adoptada pelos métodos de núcleo consiste em mapear os dados para um espaço de características onde os padrões[6] existentes possam ser descobertos através de relações lineares[7]. Este processo é realizado de forma modular já que cada um dos passos é realizado por componentes distintos.

A componente de mapeamento dos dados é definida implicitamente através de uma **função de núcleo**. Ela depende dos dados específicos do problema e do conhecimento acerca dos padrões que se espera encontrar e pode ser construída a partir de diferentes tipos de dados.

Neste novo espaço de características, são então encontrados padrões lineares através de **algoritmos de aprendizagem** genéricos, robustos e eficientes. São robustos porque têm bases estatísticas firmes que lhe conferem resistência ao sobre-ajustamento e eficientes porque requerem uma quantidade de recursos computacionais de ordem polinomial com o tamanho e número de exemplos, mesmo quando a dimensão do novo espaço cresce exponencialmente.

A Figura 5.1 mostra um exemplo entre as relações existentes nos dados no espaço de exemplos e no novo espaço de características.

[5] Do inglês, *theorically weak method*.
[6] Entende-se por padrão, uma relação existente nos dados, seja ela exacta, aproximada ou estatística.
[7] Entende-se por relação linear, uma relação que pode ser descrita através de uma função linear.

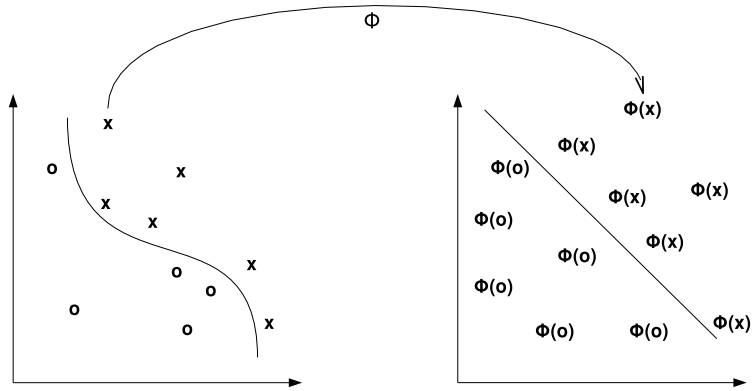

Figura 5.1: Através da função Φ, os exemplos são mapeados para um espaço de características onde as relações não lineares se tornam lineares. A função de núcleo permite encontrar essas relações directamente a partir dos exemplos.

A modularidade dos métodos de núcleo torna o desenho de sistemas de aprendizagem simples e elegante: os módulos são combinados entre si para obter um sistema de aprendizagem. Além disso, permite a reutilização do algoritmo de aprendizagem e da função de núcleo. O mesmo algoritmo pode ser utilizado com qualquer núcleo e, desta forma, em qualquer domínio e a componente do núcleo, que é específica para cada tipo de dados, pode ser combinada com diferentes algoritmos para resolver uma completa gama de tarefas de análise de padrões. Entre essas tarefas, destacam-se a classificação binária e multi-classe, a regressão, a detecção de novidades[8], o agrupamento[9] e a redução de dimensão[10]. Shawe-Taylor and Cristianini [2004] apresentam e discutem diversos (tipos de) algoritmos de aprendizagem que utilizam métodos de núcleo.

A Figura 5.2 mostra as etapas envolvidas na utilização de métodos de núcleo. Este constitui o aprendiz da aproximação de Aprendizagem Automática na classificação de textos (ver Figura 3.1).

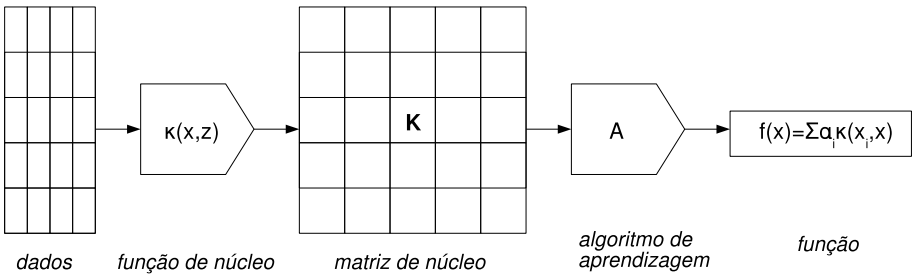

Figura 5.2: Os dados são processados utilizando uma função de núcleo para criar uma matriz que, por sua vez, é processada por um algoritmo de aprendizagem que produz uma função que reconhece padrões. Esta função é utilizada para classificar novos exemplos.

[8]Do inglês, *novelty detection*.
[9]Do inglês, *clustering*.
[10]Do inglês, *dimensionality reduction*.

Do ponto de vista computacional os métodos de núcleo têm duas propriedades importantes. Primeiro, permitem o acesso a espaços de características de muito grandes dimensões e, assim, muito flexíveis, com um baixo custo computacional em termos de tempo e espaço. Depois, e apesar da complexidade das classes de funções resultantes, uma grande parte dos algoritmos resolve problemas de optimização convexa não sofrendo, assim, de mínimos locais.

5.2 Funções de núcleo e suas propriedades

Como já referido, nos métodos de núcleo, os dados são embutidos num espaço de características de grande dimensão, onde é feita uma análise de padrões lineares, que se traduz numa análise não linear no espaço de exemplos.

Este **espaço de características** é um espaço vectorial com produto interno. A existência de produto interno permite aplicar construções geométricas que podem ser formuladas em termos de ângulos, comprimentos e distâncias. Por exemplo, o produto interno canónico $\langle \mathbf{x}, \mathbf{z} \rangle$ entre os vectores $\mathbf{x}$ e $\mathbf{z}$ pode ser interpretado geometricamente como o co-seno do ângulo entre os vectores (desde que estejam normalizados para comprimento unitário), o comprimento (ou norma) do vector $\mathbf{x}$ pode ser calculado como $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ e a distância entre dois vectores calculada como o comprimento do vector diferença.

Por outro lado, um algoritmo de aprendizagem deve ser capaz de **generalizar** para além dos dados observados. No caso da classificação isto significa que o classificador construído deverá ser capaz de indicar se um novo exemplo pertence ou não à classe em estudo e, de uma maneira geral, utiliza, de alguma forma, as etiquetas dos exemplos similares ao exemplo a classificar.

O tratamento dos exemplos de forma geométrica permite aos métodos de núcleo a utilização do produto interno como **medida de semelhança** entre exemplos e admite a utilização de algoritmos de aprendizagem baseados em álgebra linear, geometria analítica e estatística. No entanto, estes métodos não assumem que os exemplos se encontram num espaço vectorial com produto interno. Eles podem constituir qualquer tipo de informação já que a **função de núcleo** garante o mapeamento do espaço de entrada num espaço de características utilizando, tipicamente, uma função não linear Φ. A liberdade de escolha de Φ permite desenhar uma grande variedade de medidas de semelhança e algoritmos de aprendizagem.

5.2.1 Função de núcleo

Um **núcleo** é uma função κ que corresponde a um produto interno num espaço de características $\mathcal{H}$ através de um mapa Φ,

$$\begin{aligned} \Phi \ : \ & \mathcal{X} \to \mathcal{H} \\ & x \mapsto \mathbf{x} := \Phi(x), \end{aligned} \tag{5.1}$$

ou seja,

$$\kappa(x, z) = \langle \Phi(x), \Phi(z) \rangle \tag{5.2}$$

É possível verificar se uma função constitui um núcleo válido, construindo um espaço de características e calculando o produto interno entre duas imagens dos exemplos.

Exemplo 5.1. A função $\kappa(x, z) = \langle x, z \rangle^2$ constitui uma função de núcleo, já que, dado um espaço bidimensional $\mathcal{X} \subseteq \mathbb{R}^2$ e o mapa

$$\Phi : x = (x_1, x_2) \mapsto \Phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2) \in \mathcal{H} = \mathbb{R}^3, \tag{5.3}$$

a composição do mapa com o produto interno no espaço de características pode ser calculada directamente a partir de x e z:

$$\begin{aligned} \langle \Phi(x), \Phi(z) \rangle &= \left\langle (x_1^2, x_2^2, \sqrt{2}x_1x_2), (z_1^2, z_2^2, \sqrt{2}z_1z_2) \right\rangle \\ &= x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 \\ &= (x_1z_1 + x_2z_2)^2 \\ &= \langle x, z \rangle^2 \end{aligned} \tag{5.4}$$

O mesmo núcleo calcula o produto interno correspondente ao espaço de características a 4-dimensões

$$\Phi : x = (x_1, x_2) \mapsto \Phi(x) = (x_1^2, x_2^2, x_1x_2, x_2x_1) \in \mathcal{H} = \mathbb{R}^4, \tag{5.5}$$

o que mostra que o espaço de características não é determinado de forma única pela função de núcleo. □

A subsecção 5.2.3 define formalmente um núcleo (válido) como função definida positiva e apresenta as suas propriedades e as do espaços de características induzidos.

5.2.2 A matriz de Gram

A função de núcleo, quando aplicada aos dados de entrada dá origem a uma matriz designada por **matriz de Gram** ou matriz de núcleo. Esta matriz contém a avaliação do núcleo para todos os pares de um conjunto de exemplos.

Definição 5.2 (Matriz de Gram)**.** *Dada uma função $\kappa : \mathcal{X}^2 \to \mathbb{R}$ e os exemplos $x_1, \ldots, x_m \in \mathcal{X}$, a matriz K $m \times m$ com elementos*

$$K_{ij} = \kappa(x_i, x_j) \tag{5.6}$$

denomina-se ***matriz de Gram*** *ou (****matriz de núcleo****) de κ para $x_1, \ldots, x_m$.*

A matriz de núcleo (com a excepção das etiquetas dos exemplos no caso da aprendizagem supervisionada) contém toda a informação disponível para a execução do passo de aprendizagem. É através dela que o algoritmo obtém informação sobre o espaço de características e os próprios dados de treino.

No entanto, a sua utilização pode provocar alguma perca de informação quando comparada com aquela disponível no conjunto original de dados. Por exemplo, se os dados de entrada pertencerem a um espaço vectorial, a informação sobre a sua orientação com respeito à origem e sobre o alinhamento entre os pontos e os eixos é perdida, já que a matriz de produtos internos é invariante a qualquer rotação do sistema de coordenadas.

5.2.3 O núcleo e o espaço de características induzido

Definição 5.3 (Núcleo (definido positivo))**.** *Seja $\mathcal{X}$ um conjunto não vazio. Uma função simétrica $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ é um núcleo definido positivo em $\mathcal{X}$ se, para todo o $n \in \mathbb{N}$, $x_1, \ldots, x_n \in \mathcal{X}$, e $c_1, \ldots, c_n \in \mathbb{R}$, resultar que*

$$\sum_{i,j \in 1,\ldots,n} c_i c_j \kappa(x_i, x_j) \geq 0. \tag{5.7}$$

Um núcleo (definido positivo) dá origem a uma matriz de Gram definida positiva.

As definições do núcleo definido positivo (5.3) e de matriz definida positiva (equação D.33) diferem no facto de o primeiro permitir a escolha dos pontos nos quais o núcleo é avaliado – para cada escolha, o núcleo induz uma matriz de Gram definida positiva.

A propriedade do núcleo e a da respectiva matriz de Gram serem definidos positivos implica a positividade na diagonal, $\kappa(x, x) \geq 0$ e a simetria da matriz, $K_{ij} = K_{ji}$.

Núcleo reprodutivo

Teorema 5.4 (Caracterização do núcleo)**.** *Uma função contínua ou com um domínio finito*

$$\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}, \tag{5.8}$$

pode ser decomposta num mapa Φ em ambos os seus argumentos para um espaço de Hilbert $\mathcal{H}$ seguida da avaliação do produto interno em $\mathcal{H}$,

$$\kappa(x, z) = \langle \Phi(x), \Phi(z) \rangle, \tag{5.9}$$

se, e só se, constituir uma função definida positiva.

Na demonstração é construído um mapa Φ para um espaço de Hilbert no qual κ constitui um núcleo. Embora os elementos desse espaço de características sejam funções, constituem pontos num espaço vectorial e satisfazem todas as propriedades dos produtos internos. Estes elementos possuem outra propriedade conhecida por **propriedade reprodutiva** do núcleo: para todas as funções

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i \kappa(\cdot, x_i) \tag{5.10}$$

com $m \in \mathbb{N}, \alpha_i \in \mathbb{R}$ e $x_1, \ldots, x_m \in \mathcal{X}$, tem-se

$$\langle \kappa(\cdot, x), f \rangle = f(x) \tag{5.11}$$

Em particular, tem-se que $\langle \kappa(\cdot, x), \kappa(\cdot, x') \rangle = \kappa(x, x')$. Por esta razão, os núcleos (positivos definidos) são também conhecidos como **núcleos reprodutivos**.

A identidade (5.9) forma a base para o **truque do núcleo**: dado um algoritmo formulado em termos de um núcleo definido positivo κ, pode-se construir um algoritmo alternativo substituindo κ por outro núcleo definido positivo $\tilde{\kappa}$. Efectivamente, significa que, se o algoritmo original for considerado um algoritmo que opera nos dados vectoriais $\Phi(x_1), \ldots, \Phi(x_m)$ então o algoritmo obtido ao substituir κ por $\tilde{\kappa}$ é exactamente o mesmo algoritmo mas que opera nos dados $\tilde{\Phi}(x_1), \ldots, \tilde{\Phi}(x_m)$.

Dado um núcleo reprodutivo κ, o espaço de características correspondente, $\mathcal{H}_\kappa$, denomina-se Espaço de Hilbert de Núcleo Reprodutivo (EHNR)[11] e o produto interno denota-se por $\langle \cdot, \cdot \rangle_{\mathcal{H}_\kappa}$.

Qualquer núcleo pode ser utilizado para construir um espaço de Hilbert que satisfaça a propriedade reprodutiva. De igual forma, se uma função simétrica $\kappa(\cdot, \cdot)$ satisfizer a propriedade reprodutiva num espaço de Hilbert $\mathcal{H}$ de funções, então κ é um núcleo definido positivo.

Esta caracterização dos núcleos não requer que os dados pertençam ao espaço vectorial; aplica-se a todos os tipos de dados como cadeias, estruturas discretas, imagens ou séries temporais; desde que a matriz de núcleo correspondente a um conjunto de dados finito

[11] Do inglês, *Kernel Hilbert Space*.

seja definida positiva, o núcleo calcula o produto interno depois de projectar os dados num espaço de características.

Núcleo de Mercer

Em vez do espaço de funções do espaço de Hilbert de núcleo reprodutivo, o teorema de Mercer define o espaço de características em termos de um espaço vectorial explícito utilizando o espaço de funções $L_2(\mathcal{X})$ (espaço de funções quadradas integráveis). Tipicamente, este teorema é utilizado para construir um espaço de características de um núcleo.

Teorema 5.5 (Mercer)**.** *Seja $\mathcal{X}$ um subconjunto compacto de $\mathbb{R}^N$. Suponha-se que κ é uma função contínua simétrica tal que o operador integral*

$$\begin{aligned} T_\kappa &: L_2(\mathcal{X}) \to L_2(\mathcal{X}) \\ (T_\kappa f)(x) &:= \int_{\mathcal{X}} \kappa(x,x') f(x') d\mu(x'), \end{aligned} \tag{5.12}$$

é positivo definido, ou seja, para todo o $f \in L_2(\mathcal{X})$ tem-se

$$\int_{\mathcal{X}^2} \kappa(x,x') f(x) f(x') d\mu(x) d\mu(x') \geq 0. \tag{5.13}$$

Então, é possível expandir $\kappa(x,x')$ numa série convergente uniforme (em $\mathcal{X}^2$) em termos das funções Φ_j que satisfazem $\langle \Phi_j, \Phi_i \rangle = \delta_{ij}$, sendo δ_{ij} o delta de Kronecker, que satisfaz $\delta_{ij} = 1$ se $i = j$ e 0 nos outros casos (ver secção D.12).

$$\kappa(x,x') = \sum_{j=1}^{\infty} \Phi_j(x) \Phi_j(x'). \tag{5.14}$$

Mais ainda, a série $\sum_{i=1}^{\infty} ||\Phi_i||^2_{L_2(\mathcal{X})}$ é convergente.

Para um dado núcleo, é possível construir diferentes espaços de características. De facto, eles podem até diferir na sua dimensão. No entanto, todos são equivalentes desde que as operações realizadas se reduzam a produtos internos: por exemplo, se Φ_1, Φ_2 forem mapas para os espaços de características $\mathcal{H}_1, \mathcal{H}_2$, respectivamente, ambos associados ao mesmo núcleo κ, tem-se que

$$\kappa(x,x') = \langle \Phi_i(x), \Phi_i(x') \rangle_{\mathcal{H}_i}, \text{ para } i = 1,2, \tag{5.15}$$

e, embora normalmente, $\Phi_1(x) \neq \Phi_2(x)$, a igualdade $\langle \Phi_1(x), \Phi_1(x') \rangle = \langle \Phi_2(x), \Phi_2(x') \rangle$ verifica-se sempre.

5.2.4 Medidas básicas no espaço definido por um núcleo

Ao trabalhar num espaço de características definido por um núcleo não é possível representar os pontos explicitamente. Por exemplo, a imagem de um exemplo x é $\Phi(x)$ mas não é possível aceder às componentes deste vector; apenas é possível avaliar o produto interno entre as imagens de x e de outros exemplos. Apesar desta limitação, é possível recolher uma quantidade de informação considerável acerca das imagens dos exemplos no novo espaço.

Utilizando apenas o resultado dos produtos internos contido na matriz de núcleo $\mathbf{K}$ sobre um subconjunto do espaço com produto interno $\mathcal{H}$, é possível obter variada informação sobre o mesmo. Dado um subconjunto finito $S = \{x_1, \ldots, x_m\}$ de um espaço $\mathcal{X}$, um núcleo $\kappa(x, x')$ e um mapa Φ para espaço de características $\mathcal{H}$ que satisfaz $\kappa(x, x') = \langle \Phi(x), \Phi(x') \rangle$, $\Phi(S) = \{\Phi(x_1), \ldots, \Phi(x_m)\}$ constitui a imagem de S sob o mapa Φ.

Norma do vector característica. A norma da imagem de x, $\Phi(x)$, é dada por

$$\|\Phi(x)\|_2 = \sqrt{\|\Phi(x)\|^2} = \sqrt{\langle \Phi(x), \Phi(x) \rangle} = \sqrt{\kappa(x, x)} \tag{5.16}$$

Normalização. A normalização é dada pela transformação

$$\hat{\Phi}(x) = \frac{\Phi(x)}{\|\Phi(x)\|}. \tag{5.17}$$

Para dois pontos, o núcleo transformado $\hat{\kappa}$ é dado por

$$\begin{aligned} \hat{\kappa}(x, z) &= \left\langle \hat{\Phi}(x), \hat{\Phi}(z) \right\rangle = \left\langle \frac{\Phi(x)}{\|\Phi(x)\|} \frac{\Phi(z)}{\|\Phi(z)\|} \right\rangle \\ &= \frac{\langle \Phi(x), \Phi(z) \rangle}{\|\Phi(x)\| \|\Phi(z)\|} = \frac{\kappa(x, z)}{\sqrt{\kappa(x, x)\kappa(z, z)}} \end{aligned} \tag{5.18}$$

Também é possível avaliar a norma da combinação linear das imagens no espaço de características

$$\begin{aligned} \left\| \sum_{i=1}^{l} \alpha_i \Phi(x_i) \right\|^2 &= \left\langle \sum_{i=1}^{l} \alpha_i \Phi(x_i), \sum_{i=1}^{l} \alpha_j \Phi(x_j) \right\rangle \\ &= \sum_{i=1}^{l} \alpha_i \sum_{j=1}^{l} \alpha_j \langle \Phi(x_i), \Phi(x_j) = \sum_{i,j=1}^{l} \alpha_i \alpha_j \kappa(x_i, x_j). \end{aligned} \tag{5.19}$$

Distância entre vectores de características. Um caso especial da norma é a distância entre duas imagens $\Phi(x)$ e $\Phi(z)$, que pode ser calculada como

$$\begin{aligned} \|\Phi(x) - \Phi(z)\|^2 &= \langle \Phi(x) - \Phi(z), \Phi(x) - \Phi(z) \rangle \\ &= \langle \Phi(x), \Phi(x) \rangle - 2\langle \Phi(x), \Phi(z) \rangle + \langle \Phi(z), \Phi(z) \rangle \\ &= \kappa(x, x) - 2\kappa(x, z) + \kappa(z, z). \end{aligned} \tag{5.20}$$

Norma e distância ao centro de massa. O centro de massa do conjunto $\Phi(S)$ é o vector

$$\Phi_S = \frac{1}{m} \sum_{i=1}^{m} \Phi(x_i). \tag{5.21}$$

Como nos casos anteriores, não existe uma representação explícita deste ponto, mas, neste caso, pode até não existir um ponto em $\mathcal{X}$ cuja imagem em Φ seja Φ_S, ou seja, potencialmente este ponto não pertence a $\Phi(S)$. No entanto, é possível calcular a sua norma, utilizando apenas a avaliação do núcleo nos elementos de $\Phi(S)$

$$\begin{aligned} \|\Phi_S\|_2^2 &= \langle \Phi_S, \Phi_S \rangle = \left\langle \frac{1}{m} \sum_{i=1}^{m} \Phi(x_i), \frac{1}{m} \sum_{j=1}^{l} \Phi(x_j) \right\rangle \\ &= \frac{1}{m^2} \sum_{i,j=1}^{m} \langle \Phi(x_i), \Phi(x_j) \rangle = \frac{1}{m^2} \sum_{i,j=1}^{m} \kappa(x_i, x_j), \end{aligned} \tag{5.22}$$

ou seja, o quadrado da norma do centro de massa é igual à média das entradas da matriz de núcleo, o que implica que a soma seja maior ou igual a zero; a igualdade acontece quando o centro de massa estiver na origem do sistema de coordenadas.

De igual forma, é possível calcular o quadrado da distância da imagem de um ponto x ao centro de massa Φ_S

$$\begin{aligned} \|\Phi(x) - \Phi_S\|^2 &= \langle \Phi(x), \Phi(x) \rangle + \langle \Phi_S, \Phi_S \rangle - 2\langle \Phi(x), \Phi_S \rangle \\ &= \kappa(x, x) + \frac{1}{m^2} \sum_{i,j=1}^{m} \kappa(x_i, x_j) - \frac{2}{m} \sum_{i=1}^{m} \kappa(x, x_i). \end{aligned} \tag{5.23}$$

Distância esperada ao centro de massa. Seguindo a mesma aproximação, o valor esperado do quadrado da distância de um ponto do conjunto à sua média é dado por

$$\begin{aligned}\frac{1}{m}\sum_{s=1}^{m}\|\Phi(x_s)-\Phi_S\|^2 &= \frac{1}{m}\sum_{s=1}^{m}\kappa(x_s,x_s)+\frac{1}{m^2}\sum_{i,j=1}^{m}\kappa(x_i,x_j)-\frac{2}{m^2}\sum_{i,s=1}^{m}\kappa(x_s,x_i)\\ &= \frac{1}{m}\sum_{s=1}^{m}\kappa(x_s,x_s)-\frac{1}{m^2}\sum_{i,j=1}^{m}\kappa(x_i,x_j), \qquad (5.24)\end{aligned}$$

ou seja, a média do quadrado da distância dos pontos ao seu centro de massa é a média da diagonal da matriz de núcleo menos a média de todas as suas entradas.

5.2.5 Operações sobre núcleos

A caracterização das funções de núcleo referida na subsecção 5.2.3 não serve apenas para decidir se uma função candidata constitui, ou não, um núcleo definido positivo. Uma das suas principais consequências é poder ser utilizada para justificar um conjunto de regras de manipulação e combinação de núcleos simples para obter outros mais complexos e úteis. Estas operações, quando aplicadas a um núcleo dão origem a um novo núcleo, ou seja, a classe das funções de núcleo é **fechada** para tais operações.

Proposição 5.6 (Propriedades de fecho)**.** *Sejam κ_1 e κ_2 núcleos definidos em $\mathcal{X}\times\mathcal{X}$, com $\mathcal{X}\subseteq\mathbb{R}^n$, $a\in\mathbb{R}^+$, $f(\cdot)$ uma função real em $\mathcal{X}$, $\Phi:\mathcal{X}\to\mathbb{R}^N$, sendo κ_3 um núcleo sobre $\mathbb{R}^N\times\mathbb{R}^N$ e $\mathbf{B}$ uma matriz $n\times n$ simétrica definida positiva. Então, as seguintes funções são núcleos:*

(i). $\kappa(x,x')=\kappa_1(x,x')+\kappa_2(x,x')$,
(ii). $\kappa(x,x')=a\kappa_1(x,x')$,
(iii). $\kappa(x,x')=\kappa_1(x,x')\kappa_2(x,x')$,
(iv). $\kappa(x,x')=f(x)f(x')$,
(v). $\kappa(x,x')=\kappa_3(\Phi(x),\Phi(x'))$,
(vi). $\kappa(x,x')=x^\top Bx'$.

Proposição 5.7. *Seja $\kappa_1(x,x')$ um núcleo definido em $\mathcal{X}\times\mathcal{X}$ e $p(x)$ é um polinómio com coeficientes positivos. Então as funções seguintes são também núcleos:*

(i). $\kappa(x,x')=p(\kappa_1(x,x'))$,
(ii). $\kappa(x,x')=\exp(\kappa_1(x,x'))$,
(iii). $\kappa(x,x')=\exp(-\|x-x'\|^2/(2\sigma^2))$.

Também é possível transformar o espaço de características realizando operações directamente sobre a matriz de núcleo desde que a matriz se mantenha simétrica e definida positiva. Uma descrição das operações possíveis pode ser encontrada em [Shawe-Taylor and Cristianini 2004, Capítulo 3].

5.2.6 Exemplos de núcleos

Apresentam-se, de seguida, alguns exemplos de núcleos genéricos que têm sido utilizados com sucesso. Para tal, assume-se que $\mathcal{X} \subset \mathbb{R}^N$. O **núcleo polinomial homogéneo**

$$\kappa(x, x') = \langle x, x' \rangle^d \tag{5.25}$$

calcula o produto interno dos vectores obtidos pelo mapa que transforma $x \in \mathbb{R}^N$ num vector cujas entradas são todos os produtos ordenados de grau d de elementos de x.

Para além deste núcleo, Boser et al. [1992], Guyon et al. [1993] e Vapnik [1995] sugerem a utilização do **núcleo Gaussiano** de funções de base radial

$$\kappa(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right), \text{ com } \sigma > 0 \tag{5.26}$$

e o **núcleo de sigmóide**

$$\kappa(x, x') = \tanh(\kappa\langle x, x' \rangle + \vartheta), \text{ com } \kappa > 0, \vartheta < 0. \tag{5.27}$$

Outros núcleos incluem o **núcleo polinomial inhomogéneo**

$$\kappa(x, x') = (\langle x, x' \rangle + c)^d, \text{ com } d \in \mathbb{N}, c \geq 0 \tag{5.28}$$

e o **núcleo B_n-spline**

$$\kappa(x, x') = B_{2p+1}(\|x - x'\|) \text{ com } B_n := \bigotimes_{i=1}^{n} I_{\left[-\frac{1}{2}, \frac{1}{2}\right]}, \tag{5.29}$$

onde I_X denota a função indicador no conjunto X e $\otimes$ denota o operador convolução $(f \otimes g)(x) := \int f(x')g(x' - x)dx'$. Este núcleo calcula B-`splines` de ordem $2p + 1$ ($p \in \mathbb{N}$), definido pela convolução $(2p + 1)$-dobras no intervalo unitário $[-1/2, 1/2]$.

Os núcleos de função de base radial[12] são núcleos que podem ser escritos na forma

$$\kappa(x, x') = f(d(x, x')), \tag{5.30}$$

[12] Do inglês, *radial basis function kernels*.

onde d é uma métrica em $\mathcal{X}$ e f é uma função em $\mathbb{R}_0^+$. Os núcleos Gaussiano e *B-spline* mencionados atrás constituem, assim, exemplos deste tipo de núcleo.

5.3 Máquinas de vectores de suporte

Como referido na secção 5.1, os métodos de núcleo utilizam algoritmos genéricos, robustos e eficientes para a análise de padrões. As máquinas de vectores de suporte constituem um desses algoritmos para as tarefas de classificação e regressão. A teoria matemática que está na sua base e que lhe confere robustez designa-se por **teoria da aprendizagem estatística**[13].

Os **classificadores por hiperplanos**[14] são algoritmos de classificação que induzem classificadores lineares num espaço com produto interno e cujo desempenho estatístico pode ser controlado. As máquinas de vectores de suporte estendem estes classificadores realizando operações equivalentes no espaço de entrada através da função de núcleo.

5.3.1 Teoria da aprendizagem estatística

Como referido na secção 3.4.1, um classificador sobre-ajustado possui bons níveis de desempenho quando reclassifica os exemplos de treino mas, quando aplicado a novos exemplos, os seus níveis de desempenho costumam ser muito baixos. Por outro lado, a utilização de um modelo demasiado simples pode não conseguir classificar de forma adequada uma grande parte dos exemplos. Um modelo de complexidade intermédia que consiga classificar correctamente a maioria dos exemplos representa um bom compromisso entre aqueles dois extremos. A Figura 5.3 mostra um exemplo de classificação binária resolvida utilizando três modelos, onde se mostram as fronteiras de decisão.

A teoria de aprendizagem estatística representa estes argumentos intuitivos matematicamente através da análise das propriedades matemáticas dos algoritmos de aprendizagem. Estas propriedades são, normalmente, propriedades da classe de funções que o algoritmo consegue implementar.

Nesta teoria, um problema de classificação binária é formalizado como a tarefa de inferir uma função desconhecida $f : \mathcal{X} \to \{\pm 1\}$ a partir de um conjunto não vazio de exemplos de treino $\{(x_1, y_1) \ldots, (x_m, y_m)\} \in \mathcal{X} \times \{\pm 1\}$[15] assumindo que os dados são

[13] Do inglês, *statistical learning theory*.

[14] Do inglês, *hyperplane classifiers*.

[15] Esta formalização é equivalente àquela apresentada na secção 3.1 considerando apenas uma classe onde os valores Booleanos V e F são substituidos pelos valores +1 e -1, respectivamente. A pertença

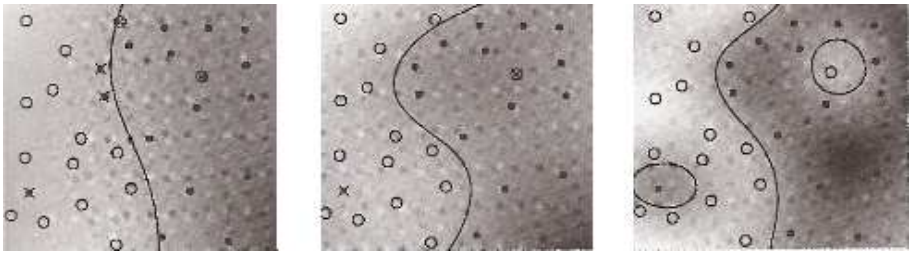

Figura 5.3: Exemplo de classificação binária resolvida utilizando três modelos. Os modelos variam em complexidade desde o mais simples (esquerda) que classifica erradamente um grande número de pontos, até ao mais complexo cuja solução é consistente com todos os pontos de treino (mas pode não funcionar tão bem com novos pontos).

gerados de forma independente por uma distribuição de probabilidade desconhecida (mas fixa) $\mathrm{P}(x, y)$. Os dados gerados desta forma são usualmente referidos como dados **distribuidos de forma idêntica e independente**[16]. O objectivo é, pois, encontrar a função f que classifica exemplos desconhecidos, tal que $f(x) = y$ para os exemplos (x, y) gerados pela distribuição $\mathrm{P}(x, y)$.

A correcção da classificação é medida através de uma função cujo valor é 0 se o exemplo (x, y) for classificado correctamente e 1, caso contrário. Esta função designa-se por **função de perda zero-um**[17] e é definida como

$$c(x, y, f(x)) := \frac{1}{2}|f(x) - y| \tag{5.31}$$

No entanto, a minimização do erro (médio) de treino ou **risco empírico**,

$$R_{emp} = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}|f(x_i - y_i| \tag{5.32}$$

não implica um **risco** (ou erro de teste) mínimo, valor medido sobre os exemplos de teste pertencentes à mesma distribuição $\mathrm{P}(x, y)$,

$$R[f] = \int \frac{1}{2}|f(x) - y| d\mathrm{P}(x, y) \tag{5.33}$$

O risco pode ser definido para qualquer função de perda, desde que o integral exista. Para a função de perda zero-um, o risco é igual à probabilidade de classificação errada.

A teoria de aprendizagem estatística ou teoria VC (Vapnik-Chervonenkis) mostra que é imperativo restringir a escolha da função f a uma classe de funções que tenha uma

ou não à classe é sinalizada com os valores +1 e -1 por questões de conveniência matemática.

[16]Do inglês, *independent and identically distributed*, (idd).

[17]Do inglês, *zero-one loss function*.

capacidade adequada à dimensão do conjunto de treino disponível. Esta teoria fornece limites para o erro de teste, e a sua minimização, que depende do risco empírico e da capacidade da classe da função, constitui o princípio da **minimização do risco estrutural**[18].

Cada função de uma classe de funções consegue separar os exemplos de uma determinada forma, induzindo as suas etiquetas. Como as etiquetas pertencem ao conjunto $\{\pm 1\}$ existem, no máximo 2^m formas distintas de etiquetar os m exemplos; uma classe de funções muito rica poderá apreender as 2^m formas, caso em que se diz que a função consegue **quebrar**[19] os m exemplos.

A **dimensão VC** de uma classe de funções, o conceito de capacidade mais conhecido da teoria da aprendizagem estatística, é definida como o maior m para o qual existe um conjunto de m exemplos que a classe de funções consegue quebrar e ∞ se esse m não existir. A Figura 5.4 mostra um exemplo de dimensão VC.

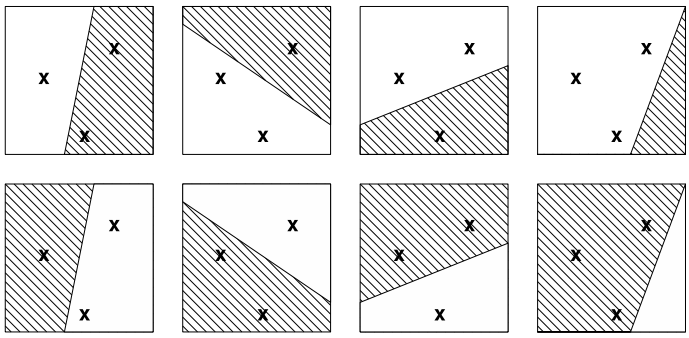

Figura 5.4: Exemplo de uma dimensão VC simples. Existem $2^3 = 8$ formas de associar 3 pontos a duas classes; para pontos representados em $\mathbb{R}^2$, todas as 8 possibilidades podem ser obtidas utilizando hiperplanos separadores, ou seja, esta classe de funções consegue separar os três pontos. Isto não resultaria se existissem 4 pontos, independentemente da sua localização. Assim, a dimensão VC da classe dos hiperplanos separadores em $\mathbb{R}^2$ é 3.

A dimensão VC constitui, de alguma forma, um conceito algo grosseiro já que sumariza a capacidade de uma classe de funções através de um único número. Outras medidas de capacidade mais precisas são a entropia arrefecida VC[20] e a função de crescimento[21].

O **limite VC**[22], um limite para o risco $R[f]$, pode ser calculado a partir da dimensão VC: se $h < m$ for a dimensão VC de uma classe de funções que o algoritmo de aprendizagem pode implementar então, para todas as funções dessa classe, independentemente

[18] Do inglês, *structural risk minimization.*
[19] Do inglês, *shatter.*
[20] Do inglês, *annealed VC entropy.*
[21] Do inglês, *growth function.*
[22] Do inglês, *VC bound.*

da distribuição de geração dos dados P, com uma probabilidade de gerar o conjunto de treino de pelo menos $1 - \delta$, mantém-se o limite

$$R[f] \leq R_{emp}[f] + \phi(h, m, \delta) \tag{5.34}$$

onde ϕ constitui o **termo de confiança**[23] (ou termo de capacidade[24]) e é definido como

$$\phi(h, m, \delta) = \sqrt{\frac{1}{m}\left(h\left(\ln\frac{2m}{h} + 1\right) + \ln\frac{4}{\delta}\right)} \tag{5.35}$$

Para obter previsões não triviais, a classe de funções deve ser restringida de tal forma que a sua capacidade seja suficientemente pequena em relação à quantidade de dados disponível (uma vez que o termo de confiança cresce monotonicamente com h). Ao mesmo tempo, a classe deve ser suficientemente larga para poder gerar funções capazes de modelar as dependências escondidas em $\mathrm{P}(x, y)$. A escolha da classe de funções é, assim, crucial para a tarefa de aprendizagem a partir de exemplos.

5.3.2 Classificadores por hiperplanos

Como descrito na secção anterior, para desenhar algoritmos cuja desempenho estatístico possa ser controlado é necessário encontrar uma classe de funções cuja capacidade possa ser calculada. Vapnik and Lerner [1963] e Vapnik and Chervonenkis [1964] consideraram a classe de hiperplanos num espaço com produto interno $\mathcal{H}$,

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = 0 \text{ onde } \mathbf{w} \in \mathcal{H}, b \in \mathbb{R} \tag{5.36}$$

que corresponde às funções de decisão

$$f(\mathbf{x}) = \mathrm{sgn}(\langle \mathbf{w}, \mathbf{x} \rangle + b) \tag{5.37}$$

e propuseram o algoritmo **Retrato Generalizado**[25], um algoritmo de aprendizagem para problemas separáveis por hiperplanos (também designados por linearmente separáveis). Este algoritmo baseia-se em dois factos:

- entre todos os hiperplanos que separam os dados, existe um único **hiperplano óptimo**, que se distingue pela margem de separação máxima entre dois quaisquer

[23] Do inglês, *confidence term.*
[24] Do inglês, *capacity term.*
[25] Do inglês, *Generalized Portrait.*

pontos e o hiperplano. Este hiperplano constitui a solução da expressão

$$\max_{\mathbf{w}\in\mathcal{H},b\in\mathbb{R}} \min\{\|\mathbf{x}-\mathbf{x}_i\| \mid \mathbf{x}\in\mathcal{H}, \langle\mathbf{w},\mathbf{x}\rangle + b = 0, i = 1,\ldots,m\}. \tag{5.38}$$

- a capacidade da classe de hiperplanos separadores decresce com o aumento da margem.

Estes factos fornecem os argumentos teóricos que suportam um bom desempenho de generalização do hiperplano óptimo. É também computacionalmente atractivo já que o hiperplano pode ser construído resolvendo um problema de programação quadrática (para o qual existem algoritmos eficientes):

$$\min_{\mathbf{w}\in\mathcal{H},b\in\mathbb{R}} \tau(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \tag{5.39}$$

sujeito às **restrições de desigualdade**

$$y_i(\langle\mathbf{w},\mathbf{x}_i\rangle + b) \geq 1 \text{ para todo o } i = 1,\ldots,m. \tag{5.40}$$

As restrições (5.40) asseguram que $f(\mathbf{x}_i)$ será $+1$ para $y_i = +1$ e -1 para $y_i = -1$. Uma descrição detalhada de como a solução constitui o hiperplano de margem máxima encontra-se em [Schölkopf and Smola 2002, Capítulo 7]. Este problema designa-se por **problema de optimização fundamental**[26].

A função τ, designada por **função objectivo**, em conjunto com as restrições (5.40), formam um problema de optimização por restrições. Estes problemas são resolvidos introduzindo os multiplicadores de Lagrange $\alpha_i \geq 0$. A função de Lagrange resultante é

$$L(\mathbf{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\| - \sum_{i=1}^{m}\alpha_i(y_i(\langle\mathbf{x}_i,\mathbf{w}\rangle + b) - 1) \tag{5.41}$$

Esta função L deve ser minimizada em relação às variáveis fundamentais[27] $\mathbf{w}$ e b e maximizada em relação às variáveis secundárias[28] α_i. Por outras palavras, deve ser encontrado um ponto estacionário; neste ponto, as derivadas parciais de L em relação às variáveis fundamentais são nulas, ou seja,

$$\frac{\partial}{\partial b}L(\mathbf{w},b,\boldsymbol{\alpha}) = 0 \text{ e } \frac{\partial}{\partial \mathbf{w}}L(\mathbf{w},b,\boldsymbol{\alpha}) = 0 \tag{5.42}$$

[26] Do inglês, *primal optimization problem.*
[27] Do inglês, *primal variables.*
[28] Do inglês, *dual variables.*

o que resulta nas equações

$$\sum_{i=1}^{m} \alpha_i y_i = 0 \tag{5.43}$$

e

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = 0 \tag{5.44}$$

O vector solução **w** possui, assim, uma expansão em termos de um subconjunto dos exemplos de treino, nomeadamente aqueles que possuem um α_i não nulo. Estes vectores designam-se por **Vectores de Suporte**[29] e, pelas condições Karush-Kuhn-Tucker[30] (condições KKT),

$$\alpha_i[y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1] = 0 \text{ para todo o } i = 1, \ldots, m, \tag{5.45}$$

encontram-se sobre a margem de separação entre dois quaisquer pontos e o hiperplano. Todos os restantes exemplos $(\mathbf{x}_j, y_j)$ são irrelevantes: as suas restrições $y_j(\langle \mathbf{w}, \mathbf{x}_j \rangle + b) \geq 1$ não são necessárias, uma vez que não aparecem na expansão do vector **w**. Por outras palavras, o hiperplano é determinado de forma completa pelos exemplos que lhe estão mais próximos, não dependendo dos restantes. A Figura 5.5 mostra-nos estes conceitos.

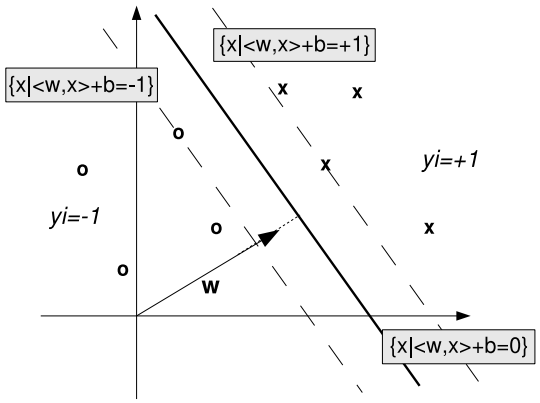

Figura 5.5: Um exemplo de classificação binária. O hiperplano óptimo está representado por uma linha sólida. Se o problema for separável, existe um vector peso **w** e um limiar b para o qual $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0$ $(i = 0, \ldots, m)$. Alterando a escala de tal forma que os pontos mais próximos do hiperplano satisfaçam $|\langle \mathbf{w}, \mathbf{x}_i \rangle + b| = 1$ obtém-se a forma canónica do hiperplano $(\mathbf{w}, b)$, que satisfaz $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$.

[29] Do inglês, *Support Vectors*.

[30] Estas condições complementares da teoria da optimização, descritas em detalhe em [Schölkopf and Smola 2002, Capítulo 6] indicam que, para todas as restrições que não são se traduzem em igualdade, isto é, $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) - 1 > 0$, os α_i correspondentes têm de ser 0. Este é o valor de α_i que maximiza L.

Ao substituir as equações (5.43) e (5.44) na função de Lagrange, eliminam-se as variáveis fundamentais $\mathbf{w}$ e b obtendo o problema que é resolvido usualmente na prática – o **problema de optimização secundário**,

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^m} W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i\alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j\rangle \tag{5.46}$$

sujeito às restrições

$$\alpha_i \geq 0 \text{ para todo o } i = 1,\ldots,m \text{ e } \sum_{i=1}^{m} \alpha_i y_i = 0. \tag{5.47}$$

Utilizando a equação (5.44), a função de decisão pode ser reescrita como

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i\alpha_i\langle \mathbf{x}, \mathbf{x}_i\rangle + b\right) \tag{5.48}$$

onde b é calculado utilizando as condições da equação (5.45).

5.3.3 Classificação com vectores de suporte

Os classificadores descritos na subsecção anterior, utilizam um espaço com produto interno $\mathcal{H}$ para encontrar o hiperplano óptimo que separa os exemplos positivos dos negativos. Para expressar as fórmulas em termos dos exemplos representados no espaço de entrada $\mathcal{X}$ é necessário utilizar uma função de núcleo já que ela permite expressar o produto interno entre os vectores $\mathbf{x}$ e $\mathbf{z}$ (pertencentes ao espaço com produto interno) através do núcleo κ avaliado nos exemplos de entrada x e z

$$\kappa(x, z) = \langle \mathbf{x}, \mathbf{z}\rangle. \tag{5.49}$$

Esta substituição foi utilizada por Boser et al. [1992] para estender o algoritmo do Retrato Generalizado para as Máquinas de Vectores de Suporte não lineares e pode ser aplicada uma vez que todos os vectores característica apenas aparecem em produtos internos (ver equações (5.46) e (5.47)). Obtém-se, assim, a função de decisão

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i\alpha_i\langle \Phi(x), \Phi(x_i)\rangle + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i\alpha_i\kappa(x, x_i) + b\right) \tag{5.50}$$

e o problema de optimização quadrática

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^m} W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} \alpha_i\alpha_j y_i y_j \kappa(x_i, x_j) \tag{5.51}$$

sujeito às restrições

$$\alpha_i \geq 0 \text{ para todo o } i = 1, \ldots, m \text{ e } \sum_{i=1}^{m} \alpha_i y_i = 0. \tag{5.52}$$

Na prática poderá não existir um hiperplano de separação como, por exemplo, quando um nível alto de ruído provoca uma grande sobreposição de classes. Para permitir a possibilidade de que alguns exemplos violem as restrições (5.40), são introduzidas **variáveis permissivas**[31] [Cortes and Vapnik 1995; Scholkopf et al. 2000; Vapnik 1995]

$$\xi_i \geq 0 \text{ para todo o } i = 0, \ldots, m, \tag{5.53}$$

para relaxar as restrições (5.40) obtendo

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \text{ para todo o } i = 1, \ldots, m. \tag{5.54}$$

Através do controlo da capacidade do classificador (via $\|\mathbf{w}\|$) e da soma das variáveis permissivas $\sum_i \xi_i$ é encontrado um classificador com boa generalização. A soma das variáveis permissivas fornece um limite superior para número de erros de treino.

Um classificador deste tipo, considerado um **classificador de margem suave**[32], pode ser obtido minimizando a função objectivo

$$\tau(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \xi_i \tag{5.55}$$

sujeita às restrições (5.53) e (5.54), onde a constante C determina o compromisso entre a maximização da margem e a minimização do erro de treino. Incorporando um núcleo e reescrevendo-o em termos dos multiplicadores de Lagrange obtém-se, de novo, o problema de maximização (5.51) sujeito às restrições

$$0 \leq \alpha_i \leq C \text{ para todo o } i = 1, \ldots, m, \text{ e } \sum_{i=1}^{m} \alpha_i y_i = 0. \tag{5.56}$$

A única diferença relativamente ao caso separável é o limite superior C para os multiplicadores de Lagrange α_i. Desta forma, a influência dos exemplos individuais (que podem ser *outliers*) é limitada. Como no caso anterior, a solução é dada pela equação (5.50). O limiar b pode ser calculado explorando o facto de que, para todos os vectores

[31] Do inglês, *slack variables*.
[32] Do inglês, *soft margin classifier*.

de suporte x_i com $\alpha_i < C$, a variável permissiva ξ_i é zero e, assim

$$\sum_{i=1}^{m} \alpha_j y_j \kappa(x_i, x_j) + b = y_i. \tag{5.57}$$

Outra variante de um classificador de margem suave utiliza a **parametrização-ν**: o parâmetro C é substituído por um parâmetro $\nu \in]0, 1]$ que permite obter os limites inferior e superior para a fracção de exemplos que será vectores de suporte e que terá variáveis permissivas diferentes de zero, respectivamente. Esta variante utiliza a função objectivo dada pela equação (5.54), com o termo de erro $(\frac{1}{\nu m} \sum_i \xi_i) - \rho$ em vez de $C \sum_i \xi_i$ e as restrições de separação que involvem o parâmetro de margem ρ,

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq \rho - \xi_i \text{ para todo o } i = 1, \ldots, m, \tag{5.58}$$

que, por sua vez, é, ele mesmo, uma variável do problema de optimização. O problema secundário consiste na maximização da parte quadrática de (5.51) sujeita às restrições $0 \leq \alpha_i \leq 1/(\nu m)$, $\sum_i \alpha_i y_i = 0$ e à restrição adicional $\sum_i \alpha_i = 1$.

Em [Cristianini and Shawe-Taylor 2000], [Shawe-Taylor and Cristianini 2004, Capítulo 7] e [Schölkopf and Smola 2002, Capítulo 7] as máquinas de vectores de suporte são apresentadas e discutidas de forma detalhada.

5.4 Núcleos para texto

A subsecção 3.4.1 apresenta as técnicas de representação dos documentos mais comuns desenvolvidas pela comunidade de IR. Estas técnicas, que retratam o documento utilizando o modelo do espaço de vectores (MEV), podem ser reinterpretadas naturalmente como métodos de núcleo. Esta nova perspectiva enriquece a compreensão desta representação, além de permitir, de forma natural, novas extensões e melhoramentos.

De seguida apresentam-se o núcleo básico para o MEV e os núcleos para aplicação a dados estruturados que, embora não sejam específicos da língua natural, podem ser aplicados ao texto.

5.4.1 Núcleos para o espaço de vectores

Como referido na subsecção 3.4.1, a versão mais simples do MEV representa um documento utilizando um saco-de-termos. Este é, por sua vez, representado num espaço vectorial através de um vector, onde cada dimensão é associada a um termo do

dicionário[33]

$$\Phi : d \mapsto \Phi(d) = (\text{ft}(t_1, d), \text{ft}(t_2, d), \ldots, \text{ft}(t_N, d)) \in \mathbb{R}^N, \tag{5.59}$$

sendo $\text{ft}(t_i, d)$, a frequência do termo t_i no documento d e N o tamanho do dicionário. Assim, tipicamente, o documento é mapeado para um espaço de muito grande dimensão. No entanto, o vector associado a determinado documento é **esparso**, ou seja, possui a maior parte das suas entradas iguais a 0, uma vez que irá conter muito poucos termos do vasto conjunto de termos possíveis.

Definição 5.8 (Matriz documento-termo)**.** *A matriz documento-termo de um corpus é a matriz cujas linhas são indexadas pelos documentos e cujas colunas são indexadas pelos termos. A entrada (i, j) possui a frequência do termo t_j no documento d_i*

$$\mathbf{D} = \begin{pmatrix} \text{ft}(t_1, d_1) & \cdots & \text{ft}(t_N, d_1) \\ \vdots & \ddots & \vdots \\ \text{ft}(t_1, d_l) & \cdots & \text{ft}(t_N, d_l) \end{pmatrix}. \tag{5.60}$$

A matriz termo-documento é a transposta $\mathbf{D}^\top$ da matriz documento-termo. A matriz termo-termo é dada por $\mathbf{D}^\top\mathbf{D}$ e a matriz documento-documento é $\mathbf{D}\mathbf{D}^\top$.

Note-se que, ao representar o documento d_i através do vector-linha $\Phi(d_i)$, a matriz documento-termo constitui a matriz de dados (os atributos são termos e os exemplos são documentos) e a matriz documento-documento não é mais que a matriz de núcleo.

Definição 5.9 (Núcleo do espaço de vectores)**.** *Utilizando o MEV, é possível criar a matriz de núcleo como*

$$\mathbf{K} = \mathbf{D}\mathbf{D}' \tag{5.61}$$

o que corresponde ao núcleo de espaço de vectores

$$\kappa(d_1, d_2) = \langle \Phi(d_1), \Phi(d_2) \rangle = \sum_{j=1}^{N} \text{ft}(t_j, d_1)\text{ft}(t_j, d_2). \tag{5.62}$$

Uma característica importante do MEV é permitir uma interpretação adicional da dualidade existente entre as representações dos algoritmos de núcleo. Neste caso, a representação secundária corresponde uma visão do problema do ponto de vista do documento (aquela utilizada no problema da classificação), enquanto que a descrição fundamental corresponde à visão do ponto de vista dos termos de indexação. Da

[33] O dicionário pode ser um conjunto pré-definido de termos mas, na prática, apenas é necessário considerar aqueles que aparecem nos documentos a processar.

mesma forma que um documento pode ser visto como uma contagem dos termos nele existentes, um termo pode ser visto como a contagem dos documentos onde aparece.

A representação dos documentos fornecida pelo MEV básico ignora qualquer relação semântica existente entre as palavras. Ao assegurar que documentos que contêm palavras semanticamente equivalentes são mapeados para vectores de características semelhantes, a representação básica será, certamente, melhorada.

Os passos de normalização e redução de dimensão referidos na secção 3.4.1 que transformam o MEV básico podem ser realizados utilizando diferentes funções de núcleo. Estas transformações têm como objectivo capturar o máximo conhecimento do domínio tentando obter, também, algum conteúdo semântico das palavras. Shawe-Taylor and Cristianini [2004] (Capítulo 10) apresentam os núcleos que traduzem algumas das representações mais conhecidas e utilizadas.

Para além dos núcleos que implementam as transformações (lineares) ao MEV básico é, também, possível considerar transformações não lineares utilizando núcleos padrão. Por exemplo, um núcleo polinomial aplicado à representação saco-de-palavras

$$\bar{\kappa}(d_1, d_2) = (\kappa(d_1, d_2) + 1)^d = (\langle \Phi(d_1), \Phi(d_2) \rangle + 1)^d \tag{5.63}$$

utiliza todos os n-tuplos de palavras, com $0 \leq n \leq d$, como atributos. A mesma aproximação pode ser utilizada para qualquer outro núcleo do espaço de vectores.

O MEV convencional foi introduzido por Salton et al. [1975] e tornou-se a representação padrão de documentos em RI, tendo sua utilização como um núcleo sido proposta por Joachims [2002]. Algumas alterações sugeridas, como o MEV generalizado [Wong et al. 1985] e a indexação semântica latente [Deerwester et al. 1990] possuem a respectiva versão de núcleo.

Para refinar a representação do espaço de vectores, Joachims [2002] e Dumais et al. [1998] utilizaram aplicações não lineares, tais como núcleos polinomiais e Gaussianos, em tarefas de classificação de texto utilizando MVS e Leopold and Kinderman [2002] compararam diferentes mapas utilizando diferentes esquemas de ponderação. Em [Cristianini et al. 2002] são discutidas as ideias da indexação semântica no âmbito dos métodos de núcleo.

5.4.2 Núcleos para objectos estruturados

O núcleo mais notório para espaços de representação que não sejam meros tuplos atributo-valor é o **núcleo de convolução**[34]; a ideia chave consiste em definir um

[34]Do inglês, *convolution kernel*.

núcleo para objectos estruturados através de núcleos sobre os seus componentes.

Entende-se por objectos estruturados, objectos formados pela composição de componentes mais simples; frequentemente utilizam-se, recursivamente, objectos mais simples do mesmo tipo. Exemplos deste tipo de dados incluem as cadeias e sequências, árvores e grafos.

Ao propôr o núcleo de convolução, Haussler [1999] utilizou a ideia da semântica de um objecto composto poder ser capturada através de uma relação R entre o objecto e os seus componentes. O núcleo sobre o objecto é assim construído a partir dos núcleos definidos sobre os diferentes componentes.

Sejam $x, x' \in \mathcal{X}$ os objectos e $\vec{x}, \vec{x}' \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_D$ tuplos dos componentes desses objectos. Dada a relação $R : \mathcal{X}_1 \times \cdots \times \mathcal{X}_D \times \mathcal{X}$, é possível definir uma decomposição R^{-1} como $R^{-1}(x) = \vec{x} : R(\vec{x}, x)$. Então o núcleo de convolução é definido como

$$\kappa_{conv}(x, x') = \sum_{\vec{x} \in R^{-1}(x), \vec{x}' \in R^{-1}(x')} \prod_{d=1}^{D} \kappa_d(x_d, x'_d) \tag{5.64}$$

O termo "núcleo de convolução" refere a classe de núcleos que podem ser formulados através da equação (5.64). A sua vantagem consiste em serem muito gerais e poderem ser aplicados a uma diversidade de problemas. No entanto, devido à sua generalidade, requerem uma quantidade significativa de trabalho para os adaptar a um problema específico, tornando a escolha de R em aplicações reais uma tarefa não trivial.

Existem outras definições de núcleos para objectos estruturados; normalmente utilizam uma sintaxe restrita e são específicos para certos domínios. As secções seguintes apresentam exemplos desses núcleos – os núcleos para cadeias, árvores e grafos[35]. Estes núcleos podem ser vistos como especializações do núcleo de convolução.

Núcleos para cadeias e sequências

Como já referido, a função de núcleo utilizada tradicionalmente para classificação de texto consiste no simples produto escalar de dois textos na sua representação saco-de-palavras [Joachims 2002] e, como tal, não considera a estrutura do texto ou das palavras, mas apenas o número de vezes que cada palavra ocorre. Aproximações mais sofisticadas tentam definir uma função de núcleo sobre a sequência de caracteres; a primeira função de núcleo definida sobre cadeias pode ser encontrada em [Watkins 1999] e [Lodhi et al. 2001] e encontra-se também descrita em [Cristianini and Shawe-Taylor 2000, Capítulo 11] e [Schölkopf and Smola 2002, Capítulo 13].

[35]Como se definem alguns conceitos gerais sobre grafos e uma árvore é um tipo especial de grafo, decidiu-se apresentar em primeiro lugar um núcleo sobre grafos e só depois os núcleos sobre árvores.

Este núcleo, designado por núcleo de subsequências de cadeias[36] baseia a semelhança de duas cadeias no número de subsequências comuns. Estas subsequências não necessitam ser contíguas nas cadeias mas, quanto mais saltos[37] existirem na ocorrência, menor é o seu peso na função de núcleo.

Paass et al. [2002] utilizam uma alternativa deste núcleo ao considerar apenas as subsequências contíguas (também designadas por subcadeias) de uma dada cadeia. Uma cadeia é então representada pelo número de vezes que cada subcadeia de até n símbolos ocorre na sequência. Esta representação das cadeias pelas suas subcadeias é conhecida como o *spectrum* da cadeia ou a sua representação em n-gramas. A função de núcleo é apenas o produto escalar desta representação. Resultados empíricos que comparam estes núcleos podem ser encontrados em [Lodhi et al. 2001].

Vishwanathan and Smola [2003] consideram núcleos de cadeias similares ao núcleo de n-gramas e mostram como estes podem ser calculados de forma eficiente utilizando árvores sufixas.

Núcleos para grafos

A aplicação de funções de núcleo a dados estruturados representados por grafos foram introduzidos de forma independente por Gärtner [2002] e Kashima and Inokuchi [2002]. Conceptualmente, estes núcleos baseiam-se numa medida sobre os percursos nos dois grafos que têm algumas ou todas as etiquetas em comum. No primeiro, contam-se os percursos com etiquetas iniciais e finais iguais e no outro são calculadas as probabilidades de percursos aleatórios com sequências de etiquetas iguais.

Gärtner et al. [2003] provam que o cálculo de qualquer função de núcleo que seja capaz de reconhecer completamente a estrutura dos grafos é NP-hard, tornando extremamente improvável encontrar núcleos eficientes e apresentam uma família de funções de núcleo baseada em percursos que inclui os núcleos [Gärtner 2002] e [Kashima and Inokuchi 2002] como casos especiais e que, ainda assim, permitem o seu cálculo em tempo polinomial.

De seguida, expõem-se algumas definições que permitem definir o núcleo desenvolvido em [Gärtner et al. 2003]. Este núcleo contabiliza, nos percursos, sequências de etiquetas iguais. O cálculo, em tempo polinomial, destes percursos (possivelmente infinitos) é possível através da utilização do conceito de grafo de produto directo[38] e do cálculo do limite (quando existe) da série geométrica de uma matriz (a matriz de adjacência).

[36] Do inglês, *string subsequence kernel*.
[37] Do inglês, *gap*.
[38] Do inglês, *direct product graph*.

Definição 5.10 (Grafo)**.** *Um* grafo *G é descrito, normalmente, por um conjunto finito de* vértices *$\mathcal{V}$, um conjunto finito* arcos *$\mathcal{A}$ e uma função Ψ. Para grafos etiquetados existe ainda um conjunto de* etiquetas *$\mathcal{E}$ e uma função de etiquetagem ε, que anexa uma etiqueta a cada arco e vértice.*

Nos grafos dirigidos*, a função Ψ mapeia cada arco com o par constituído pelo seus vértices inicial e terminal $\Psi : \mathcal{A} \to \{(u, v) \in \mathcal{V} \times \mathcal{V}\}$. Num grafo dirigido, os arcos a para os quais $\Psi(a) = (v, v)$ designam-se por laços*[39]*.*

Definição 5.11 (Percurso, caminho e ciclo)**.** *Um* percurso *p é uma sequência de vértices $v_i \in \mathcal{V}$ e arcos $a_i \in \mathcal{A}$ com $p = v_1, a_1, v_2, a_2, \ldots, a_n, v_{n+1}$ e $\Psi(a_i) = (v_i, v_{i+1})$. O* comprimento *do percurso é igual ao número de arcos da sequência, ou seja, n no caso anterior. Um* caminho *é um percurso no qual $v_i \neq v_j \Leftrightarrow i \neq j$. Um* ciclo *é um caminho seguido por um arco a_{n+1} com $\Psi(a_{n+1}) = (v_{n+1}, v_1)$.*

Definição 5.12 (Graus de entrada e saída)**.** *Para um grafo G, o* grau de saída *de um vértice v, $d^+(v)$, é o número de arcos que partem de v, ou seja, $d^+(v) = |\delta^+(v)|$ onde $\delta^+(v) = \{a \in \mathcal{A} | \Psi(a) = (v, u)\}$. De igual forma, o* grau de entrada *de v, $d^-(v)$, é o número de arcos que chegam a v, ou seja, $d^-(v) = |\delta^-(v)|$ onde $\delta^-(v) = \{a \in \mathcal{A} | \Psi(a) = (u, v)\}$. Mais ainda, os graus de entrada e saída máximos de G denotam-se por $\Delta^-(G) = \max\{d^-(v), v \in \mathcal{V}\}$ e $\Delta^+(G) = \max\{d^+(v), v \in \mathcal{V}\}$, respectivamente.*

Definição 5.13 (Matriz de adjacência)**.** *A* matriz de adjacência *E de um grafo, é a matriz cujos elementos $[E]_{ij}$ correspondem o número de arcos do vértice ν_i para o vértice ν_j*[40]*. Se se considerar a nésima potência da matriz de adjacência ($n \in \mathbb{N}, n \geq 0$), a interpretação é similar: cada elemento da matriz $[E^n]_{ij}$ constitui o número de percursos de tamanho n do vértice ν_i para o vértice ν_j.*

O núcleo sobre grafos que se descreve em seguida utiliza o grafo de produto directo. Tecnicamente baseia-se na ideia de contar o número de percursos nesse grafo de produto. Os grafos de produto são uma ferramenta da matemática discreta [Imrich and Klavzar 2000] e grafo de produto directo encontra-se entre os quatro mais importantes. Os restantes três são o Cartesiano, o forte e o lexicográfico[41].

Gärtner et al. [2003] modificam a definição original de grafo de produto directo para permitir a utilização de grafos etiquetados utilizando uma função $igual(e_1, e_2)$ que é "verdade" se as etiquetas e_1 e e_2 se igualarem de alguma forma. No caso mais simples, $igual(e_1, e_2) \Leftrightarrow e_1 = e_2$.

[39] Do inglês, *loops*.

[40] Enquanto ν_1 é utilizado para denotar o mesmo vértice, v_1 é uma variável que pode ter diferentes valores, como por exemplo $\nu_1, \nu_2, \ldots$.

[41] Do inglês, *Cartesian, strong e lexicographic, respectivamente*.

Definição 5.14 (Grafo de produto directo)**.** *O* grafo de produto directo *entre dois grafos* $G_1 = (\mathcal{V}_1, \mathcal{A}_1, \Psi_1)$ *e* $G_2 = (\mathcal{V}_2, \mathcal{A}_2, \Psi_2)$ *denota-se por* $G_1 \times G_2$ *e os grafos* G_1 *e* G_2 *designam-se por seus factores.*

O conjunto de vértices do grafo $G_1 \times G_2$ *é definido como*

$$\mathcal{V}(G_1 \times G_2) = \{(v_1, v_2) \in \mathcal{V}_1 \times \mathcal{V}_2 : igual(\varepsilon(v_1), \varepsilon(v_2))\} \tag{5.65}$$

e o seu conjunto de arcos é definido como

$$\begin{aligned} \mathcal{A}(G_1 \times G_2) = \{(a_1, a_2) \quad &\in \quad \mathcal{A}_1 \times \mathcal{A}_2 : \exists (u_1, u_2), (v_1, v_2) \in \mathcal{V}(G_1 \times G_2) \\ &\wedge \quad \Psi_1(a_1) = (u_1, v_1) \wedge \Psi_2(a_2) = (u_2, v_2) \\ &\wedge \quad igual(\varepsilon(a_1), \varepsilon(a_2))\} \end{aligned} \tag{5.66}$$

Dado um arco $(a_1, a_2) \in \mathcal{A}(G_1 \times G_2)$ *com* $\Psi_1(a_1) = (u_1, v_1)$ *e* $\Psi_2(a_2) = (u_2, v_2)$*, o valor de* $\Psi_{G_1 \times G_2}$ *é*

$$\Psi_{G_1 \times G_2}((a_1, a_2)) = ((u_1, u_2), (v_1, v_2)) \tag{5.67}$$

As etiquetas dos vértices e dos arcos no grafo $G_1 \times G_2$ *correspondem às etiquetas dos factores.*

Núcleo do grafo de produto

Definição 5.15. *Sejam* G_1, G_2 *dois grafos,* $E_\times$ *a matriz de adjacência do seu produto* $E_\times = E(G_1 \times G_2)$ *e* $\mathcal{V}_\times$ *o conjunto de vértices do produto directo* $\mathcal{V}_\times = \mathcal{V}(G_1 \times G_2)$*. Com uma sequência de pesos* $\lambda = \lambda_0, \lambda_1, \ldots$ $(\lambda_i \in \mathbb{R}; \lambda_i \geq 0$ *para todo o* $i \in \mathbb{N})$*, o núcleo do grafo de produto é definido como*

$$\kappa_\times(G_1, G_2) = \sum_{i,j=0}^{|\mathcal{V}_\times|} \left[\sum_{n=0}^{\infty} \lambda_n E_\times^n \right]_{ij} \tag{5.68}$$

se o limite existir.

Gärtner et al. [2003] demonstram que este núcleo constitui uma função definida positiva, e mostram que corresponde ao produto interno num espaço de características composto por todas as sequências de etiquetas contíguas no grafo. Cada valor de característica corresponde ao número de percursos com aquela sequência de etiquetas pesado por $\sqrt{\lambda_n}$ onde n constitui o comprimento da sequência.

Utilizando a série exponencial, $\lambda_i = \beta^i / i!$, ou a série geométrica, $\lambda_i = \gamma^{-i}$, este núcleo possui uma complexidade temporal cúbica. Extensões sugeridas em [Gärtner et al.

2003] incluem um núcleo para contar sequências de etiquetas com saltos e um para lidar com grafos de transição, ou seja, grafos com uma distribuição de probabilidade nos arcos.

Núcleos para árvores

As características utilizadas para descrever as cadeias são as suas subcadeias. De forma semelhante, para dados descritos através de árvores, o espaço de características é indexado por subárvores. A semelhança entre árvores baseia-se nas contabilização de subárvores comuns.

Collins and Duffy [2002] apresentam uma função de núcleo que pode ser aplicada em várias tarefas de processamento de língua natural: utilizando árvores ordenadas etiquetadas, o núcleo captura a informação estrutural sobre as árvores sintácticas considerando todas as subárvores que nela ocorrem.

De seguida, expõem-se algumas definições que permitem definir o núcleo desenvolvido em [Collins and Duffy 2002] que contabiliza os subconjuntos de árvores comuns entre duas árvores. Este núcleo é definido à custa de um mais simples que considera apenas as subárvores da mesma raíz.

Definição 5.16 (Árvore)**.** *Uma* árvore *A é um grafo acíclico dirigido e ligado*[42] *no qual todos os vértices v excepto um, possuem um grau de entrada igual a 1, $d^-(v) = 1$.*

Nas árvores, os vértices designam-se por nós*: o nó com $d^-(v) = 0$ designa-se por* raíz *da árvore, $raiz(A)$, os nós com grau de saída igual a 0, $d^+(v) = 0$, designam-se por* folhas *e aqueles com grau de saída diferente de zero por nós internos. Os nós aos quais um nó v está ligado são conhecidos como seus* filhos*, enquanto v é seu pai. Dois filhos do mesmo pai designam-se por nós irmãos.*

Uma árvore ordenada *é uma árvore na qual existe uma ordem nos filhos de um nó, $filho_1(v), \ldots, filho_{d^+(v)}(v)$. Uma árvore etiquetada é uma árvore onde cada nó tem uma etiqueta associada.*

Duas árvores são idênticas *se existir uma correspondência um-para-um entre os nós que respeitam a relação pai-filho e, caso sejam ordenadas, a ordem dos filhos em cada nó. Duas árvores etiquetadas são idênticas se forem idênticas como árvores e os nós correspondentes tiverem as mesmas etiquetas.*

Uma árvore própria *contém pelo menos um arco. Utiliza-se $\mathcal{P}$ para designar o conjunto de todas as árvores próprias e $\mathcal{P}_\mathcal{E}$ as árvores próprias etiquetadas (com etiquetas do conjunto $\mathcal{E}$).*

[42] Do inglês, *directed connected acyclic graph*.

Definição 5.17 (Subárvores: geral e da mesma raíz)**.** *Uma* subárvore completa $\tau(v)$ *de uma árvore A num nó v é a árvore obtida ao considerar o nó v como raíz e todos os vértices e arcos alcançados a partir de v.*

Uma subárvore da mesma raíz[43] *da árvore A é uma árvore resultante da remoção das subárvores completas de todos os filhos dos nós seleccionados; assim, se um nó v for incluído numa subárvore da mesma raíz, então todos os seus irmãos também o são. A raíz da árvore A é incluída em todas as subárvores da mesma raíz.*

Uma subárvore geral *da árvore A é qualquer subárvore da mesma raíz de uma subárvore completa.*

Definição 5.18 (Núcleo de subárvores da mesma raíz)**.** *O espaço de características associado ao núcleo de subárvore da mesma raíz é indexado por $I = \mathcal{A}$, o conjunto de todas as árvores próprias, com o mapa dado por*

$$\Phi^r_S(A) = \begin{cases} 1 & \text{se } S \text{ é uma subárvore da mesma raíz de } A; \\ 0 & \text{caso contrário.} \end{cases} \tag{5.69}$$

O núcleo associado é definido como

$$\kappa_r(A_1, A_2) = \langle \Phi^r(A_1), \Phi^r(A_2) \rangle = \sum_{S \in \mathcal{A}} \Phi^r_S(A_1) \Phi^r_S(A_2). \tag{5.70}$$

Se qualquer das árvores A_1 ou A_2 constituírem um nó único, temos $\kappa_r(A_1, A_2) = 0$, já que $\Phi^r(A) = \mathbf{0}$ para uma árvore imprópria A; se $d^+(raiz(A_1)) \neq d^+(raiz(A_2))$, então $\kappa_r(A_1, A_2) = 0$ já que uma subárvore da mesma raíz de A_1 não pode ser subárvore da mesma raíz de A_2. Assume-se então que

$$d^+(raiz(A_1)) = d^+(raiz(A_2)). \tag{5.71}$$

Conhecendo os valores do núcleo entre as subárvores completas dos filhos correspondentes de $raiz(A_1)$ e $raiz(A_2)$, ou seja,

$$\kappa_r(\tau(filho_i(raiz(A_1))), \tau(filho_i(raiz(A_2)))), \text{ para } i = 1, \ldots, d^+(raiz(A_1)), \tag{5.72}$$

o valor do núcleo entre as duas árvores A_1 e A_2 é definido (recursivamente) como

$$\begin{aligned} \kappa_r(A_1, A_2) &= \sum_{S \in \mathcal{A}} \Phi^r_S(A_1) \Phi^r_S(A_2) \qquad (5.73) \\ &= \prod_{i=1}^{d^+(raiz(A_1))} \sum_{S_i \in \mathcal{A}_0} \Phi^r_{S_i}(\tau(filho_i(raiz(A_1)))) \Phi^r_{S_i}(\tau(filho_i(raiz(A_2)))), \end{aligned}$$

[43] Do inglês, *co-rooted subtree*.

onde $\mathcal{A}_0$ denota o conjunto de todas as árvores, tanto próprias como impróprias, uma vez que as subárvores próprias da mesma raíz de A são determinadas por qualquer combinação de árvores da mesma raíz de $\tau(filho_i(raiz(A)))$, $i = 1, \ldots, d^+(raiz(A))$. Uma vez que só existe uma subárvore imprópria da mesma raíz, obtém-se a recursividade

$$\kappa_r(A_1, A_2) = \prod_{i=1}^{d^+(raiz(A_1))} (\kappa_r(\tau(filho_i(raiz(A_1))), \tau(filho_i(raiz(A_2)))) + 1). \quad (5.74)$$

Núcleo de subconjuntos de árvores

Definição 5.19 (Núcleo de subconjuntos de árvores)**.** *O espaço de características associado ao núcleo de subconjuntos de árvores é indexado por $I = \mathcal{A}$, o conjunto de todas as árvores próprias, com o mapa*

$$\Phi_S^r(A) = \begin{cases} 1 & \text{se } S \text{ é uma subárvore } A; \\ 0 & \text{caso contrário.} \end{cases} \quad (5.75)$$

O núcleo associado é definido como

$$\kappa(A_1, A_2) = \langle \Phi(A_1), \Phi(A_2) \rangle = \sum_{S \in \mathcal{A}} \Phi_S(A_1) \Phi_S(A_2). \quad (5.76)$$

O cálculo deste núcleo pode ser reduzido ao caso das subárvores da mesma raíz avaliando aquele núcleo para todos os pares de nós nas duas árvores, ou seja,

$$\kappa(A_1, A_2) = \sum_{v_1 \in A_1, v_2 \in A_2} \kappa_r(\tau(v_1), \tau(v_2)). \quad (5.77)$$

Capítulo 6

Núcleo para informação semântica

A representação dos documentos tem uma influência fundamental na forma como o algoritmo de aprendizagem generaliza e, uma vez que o espaço de hipóteses do classificador (como, por exemplo, o tipo de regras geradas e a sua ordenação) é inerente ao algoritmo, a escolha da representação torna-se a ferramenta central de modelação. O texto deve ser transformado numa representação que seja apropriada quer para o algoritmo de aprendizagem, quer para a tarefa de classificação.

Como já referido, a aproximação padrão para a classificação de texto utiliza a técnica clássica de representação de texto [Salton et al. 1975] que mapeia um documento para um vector de características de grande dimensão, onde cada componente do vector representa a presença ou ausência de uma característica. Esta aproximação perde toda a informação sobre a ordem das palavras, retendo apenas a frequência dos termos que aparecem no documento. Esta técnica é usualmente acompanhada pela remoção de palavras não informativas (*stopwords*) e pela substituição das palavras pelo seu radical ou raíz.

Este trabalho propõe uma aproximação radicalmente diferente ao considerar a informação semântica dos documentos. Cada documento é representado na sua forma lógica através de Estruturas de Representação do Discurso. Estas são, por sua vez, representadas de forma adequada para permitir a utilização de métodos de núcleo, nomeadamente um núcleo para grafos. Os documentos são mapeados para um espaço de características constituído pelo conjunto de todos os percursos do grafo (ver secção 5.4.2); dois documentos são mais semelhantes entre si, quanto maior número de percursos tiverem em comum.

6.1 Representação da informação semântica

Tradicionalmente, as aproximações formais para representar a semântica da língua natural centravam-se nas frases individuais, tentando explicar o seu significado em termos de condições de satisfabilidade. No entanto, há muito que é reconhecido que o conteúdo e o contexto estão interligados e que, de facto, um possui um factor determinante na resolução do outro.

Este foco na dependência do contexto, traduziu-se numa mudança importante da concepção "clássica" da semântica formal, que se baseia nos conceitos de referência e verdade, para uma perspectiva na qual o conceito central não é a verdade, mas a informação. Nesta perspectiva, o significado de uma frase não é obtido pelas suas condições de satisfabilidade, mas pelo seu "potencial de alteração da informação", ou seja, a sua capacidade para modificar contextos ou estados de informação iniciais noutros.

A Teoria de Representação do Discurso (TRD) desenvolvida por Kamp and Reyle [1993] é uma destas teorias. Como referido na secção 2.5.3, a sua arquitectura baseia-se no tratamento de pressupostos [Soames 1984], em duas fases distintas. Primeiro constrói uma representação "preliminar" para cada frase considerando-a individualmente. Esta representação, traduzida através de uma **Estrutura de Representação do Discurso** (ERD), inclui todos os pressupostos impostos pela frase.

Durante a segunda fase os pressupostos da representação preliminar são verificados no contexto; sempre que necessário e viável, o contexto é "acomodado", ou seja, alterado ou actualizado. Quando todos os pressupostos tiverem sido satisfeitos, a parte da representação preliminar sem os pressupostos é fundida com o contexto (original ou alterado). O resultado é uma ERD que, além da informação do contexto (possivelmente acomodado), inclui a contribuição dada pela frase.

As soluções desta teoria para a interpretação do contexto e sua actualização baseiam-se numa concepção original da forma lógica e na utilização de Referentes do Discurso (RD) para representar as contribuições semânticas produzidas pelos sintagmas nominais. A forma lógica é representada por um conjunto de ERDs, que podem ser alargadas e fundidas; assim, as que representam frases podem ser combinadas noutras que representam discursos multi-frase. Os RDs são os constituintes das ERDs que servem para representar as entidades, podendo ser descritos como as suas variáveis.

Assim, a TRD utiliza um conjunto de ferramentas básicas (tais como os RDs, as condições sobre os RDs e sua acessibilidade) que, no seu conjunto, permitem obter uma representação semântica que incorpora o contexto. No entanto, a ferramenta utilizada neste trabalho, a aplicação SIN2SEM, apenas gera as ERDs obtidas na primeira fase do tratamento de pressupostos representando, desta forma, a **forma lógica** da frase.

6.1.1 Estrutura de Representação do Discurso

Como referido na secção 2.5.3, na TRD a representação semântica é especificada em termos de uma linguagem sobre ERDs. Uma ERD simples é um par constituído por um conjunto de referentes do discurso U, usualmente referido como o universo da ERD, e um conjunto de condições C. Na sua forma geral é representada como

$$\langle U, C \rangle$$

Intuitivamente o universo reúne as entidades do discurso enquanto as condições expressam as restrições (propriedades, relações) sobre essas entidades. Por exemplo a frase `A mãe observa a filha que brinca com a boneca` corresponde à ERD

$$\langle \{x, y, z\}, \{\texttt{mae}(x), \texttt{observar}(x,y), \texttt{filha}(y), \texttt{brincar}(y,z), \texttt{boneca}(z)\} \rangle$$

ou, através da notação gráfica comumente utilizada,

x y z
mae(x)
observar(x,y)
filha(y)
brincar(y,z)
boneca(z)

Figura 6.1: Representação gráfica da ERD correspondente à frase `A mãe observa a filha que brinca com a boneca`.

Intuitivamente, os sintagmas `a mãe`, `a filha` e `a boneca` contribuem com os referentes x, y e z para o universo da ERD e com as condições atómicas `mae`(x), `filha`(y) e `boneca`(z) para o seu conjunto de condições. Os predicados `observa` e `brinca` contribuem com a condições `observar`(x,y) e `brincar`(y,z).

A semântica associada à sua construção assegura que, num determinado modelo, esta ERD simples é verdade apenas quando existe um mapeamento dos RDs para o universo do modelo que satisfaça todas as condições do conjunto de condições. Desta forma, os referentes do discurso no topo da representação gráfica de uma ERD são dotados de quantificação existencial e o conjunto de condições é interpretado de forma conjuntiva.

6.2 A forma lógica e os métodos de núcleo

Na aproximação padrão de classificação de documentos a unidade básica de informação é a palavra [Joachims 1998]; neste trabalho utiliza-se a Estrutura de Representação do

Discurso para representar a forma lógica da frase e assim incorporar a informação semântica dos documentos.

No entanto, para utilizar um método de núcleo é necessário definir um núcleo específico ou adaptar a representação do problema a um núcleo existente. Devido à possibilidade de representar uma ERD através de um grafo dirigido e de já existirem funções de núcleo sobre este tipo estruturado de dados optou-se pela segunda hipótese.

6.2.1 A ERD como um grafo

Uma ERD pode ser transformada num grafo dirigido mapeando as unidades básicas da ERD – referentes e condições sobre referentes, nas unidades básicas do grafo – vértices e arcos, e descrevendo a forma como se interligam. Assim:

- um referente é um vértice etiquetado com o nome do referente e
- uma condição é um vértice etiquetado com o nome da condição e um conjunto de arcos que representam as restrições impostas aos referentes. Os arcos partem do nome da condição para os referentes restringidos e são etiquetados com um inteiro que indica a posição do referente na condição.

Utilizando este mapa, a ERD da Figura 6.1 é representada através do grafo (dirigido) ilustrado na Figura 6.2.

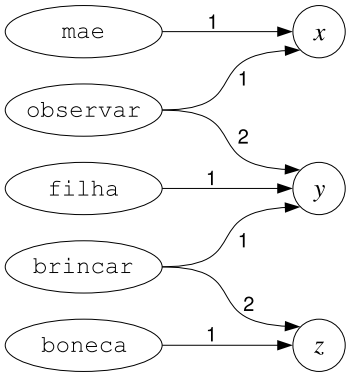

Figura 6.2: Grafo que ilustra a ERD da frase `A mãe observa a filha que brinca com a boneca`.

Esta transformação directa das ERDs em grafos não permite, no entanto, a utilização do núcleo de grafo de produto (Definição 5.15) já que a matriz de adjacência do grafo de produto directo seria nula. Esta afirmação é uma consequência da análise da definição de grafo de produto directo por um lado e das características dos grafos representantes

das ERDs, por outro. Pela definição de grafo de produto directo (Definição 5.14):

- o conjunto de vértices é constituído apenas pelos vértices comuns. Uma vez que a ferramenta SIN2SEM analisa cada frase individualmente gerando referentes distintos para cada uma sem fazer, posteriormente, qualquer tipo de unificação, o conjunto de vértices comuns é composto por aqueles que representam nomes de condições comuns aos dois grafos, e

- o conjunto de arcos é constituído pelos arcos comuns entre os vértices do grafo de produto directo. Como nesta representação apenas existem arcos que ligam os vértices dos nomes das condições aos vértices dos referentes restringidos (que são seguramente distintos de ERD para ERD), o grafo do produto é constituído apenas pelos vértices disconexos, o que determina a característica enunciada desta matriz de adjacência.

6.2.2 Substituição de referentes

Ao analisar as ERDs construidas pela TRD, constatou-se que a substituição adequada dos referentes restringidos por dois tipos de condições específicas, aquelas relacionadas com nomes próprios e as condições atómicas, conduzia a um grafo adequado à utilização do núcleo do grafo de produto.

Nomes próprios

Um nome próprio associado a uma entidade é representado através de uma condição do tipo `nome(`x`,y)`. Neste caso é possível obter uma ERD equivalente ao retirar a condição associada ao nome próprio da ERD, substituindo, nas restantes condições, o referente x por `y`. Formalmente, e utilizando a representação geral de uma ERD, esta alteração corresponde a

$$\langle U, C\rangle \longrightarrow \langle U', C'\rangle \tag{6.1}$$

com

$$U' = U - \{x : \exists y\, \mathrm{nome}(x,y) \in C\} \tag{6.2}$$

$$\begin{aligned} C' = {} & C - \{\mathrm{nome}(x,y) : \mathrm{nome}(x,y) \in C\} - \\ & -\{c : c \in C \wedge (\exists \mathrm{nome}(x,y) \in C : x \in args(c))\} \cup \\ & \cup\{d : (\exists c \in C, \exists \mathrm{nome}(x,y) \in C : x \in args(c) \wedge \\ & \quad \wedge pred(d) = pred(c) \wedge args(d) = subst(args(c), x, y))\} \end{aligned} \tag{6.3}$$

onde $pred(c)$ devolve o nome da condição c, $args(c)$ devolve a lista de referentes restringidos por c e $subst(l, a, b)$ é a função que substitui em l as ocorrências de a por b.

Se esta substituição for realizada em todas as ERDs $\langle U_i, C_i \rangle$ que contêm a condição `nome(`x`,y)`, com $x \in \cup_i U_i$, obtém-se a unificação, a nível do corpus, de todos os referentes que dizem respeito ao mesmo nome `y`. A Figura 6.3 mostra um exemplo que involve duas ERDs.

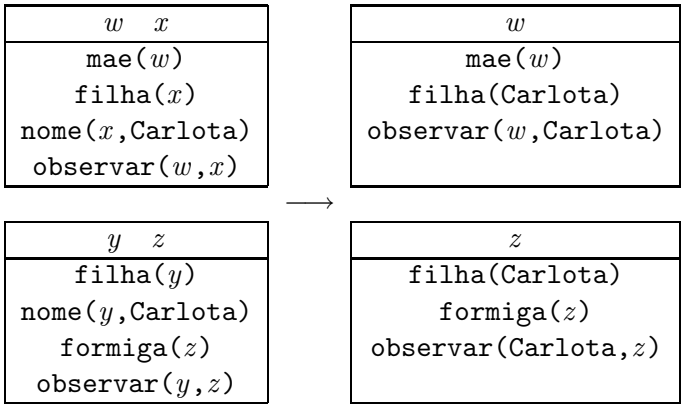

Figura 6.3: Representação gráfica da ERDs correspondente às frases `A mãe observa a filha Carlota` e `A filha Carlota observa a formiga` antes e depois da substituição dos referentes envolvidos em condições do tipo `nome(`x`,y)`.

Condições atómicas

Uma condição atómica é uma condição do tipo *cond*`(`x`)`. Neste caso, retira-se esta condição da ERD e substitui-se, nas restantes condições, o referente x por *cond*. No entanto, como é possível existir mais que uma condição atómica sobre o mesmo referente, a substituição do referente é feita por um conjunto. Formalmente, e utilizando a representação geral de uma ERD, esta alteração corresponde a

$$\langle U, C \rangle \longrightarrow \langle U', C' \rangle \tag{6.4}$$

com

$$U' = U - \{x : (\exists c \in C : args(c) = [x])\} \tag{6.5}$$

$$\begin{aligned} C' = {} & C - \{c : c \in C \wedge \#args(c) = 1\} - \\ & - \{d : d \in C \wedge (\exists x \in U, \exists c \in C : args(c) = [x] \wedge x \in args(d))\} \cup \\ & \cup \{e : (\exists d \in C, \exists x \in U, \exists c \in C : args(c) = [x] \wedge x \in args(d) \wedge \\ & \quad \wedge pred(e) = pred(d) \wedge args(e) = subst(args(d), x, y) \wedge pred(c) \in y))\} \end{aligned} \tag{6.6}$$

onde $pred(c)$ devolve o nome da condição c, $args(c)$ e $\#args(c)$ devolvem, respectivamente, a lista e o número de referentes restringidos por c e $subst(l, a, b)$ é a função que substitui em l as ocorrências de a por b.

Esta alteração apenas pode ser realizada a nível da ERD, ou seja, a nível da frase. A Figura 6.4 mostra o resultado desta substituição aplicada à ERD da Figura 6.1.

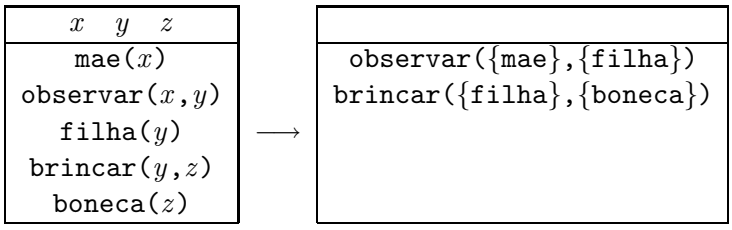

Figura 6.4: Representação gráfica da ERD correspondente à frase `A mãe observa a filha que brinca com a boneca` antes e depois da substituição dos referentes envolvidos em condições atómicas do tipo `cond(`x`)`.

Nomes próprios e condições atómicas

Como o mesmo referente x pode estar restringido por condições do tipo `nome(`x`,y)` e *cond*`(`x`)`, sempre que é feita a substituição de um referente cria-se um conjunto.

Com estas substituições, um referente deixa de ser apenas uma simples variável, podendo passar a ser um conjunto. Este conjunto contém os identificadores de condições unárias e/ou um nome próprio associados ao referente. Por exemplo, a forma lógica da frase `A mãe observa a filha Carlota que brinca com a boneca` é representada pela ERD ilustrada na Figura 6.5.

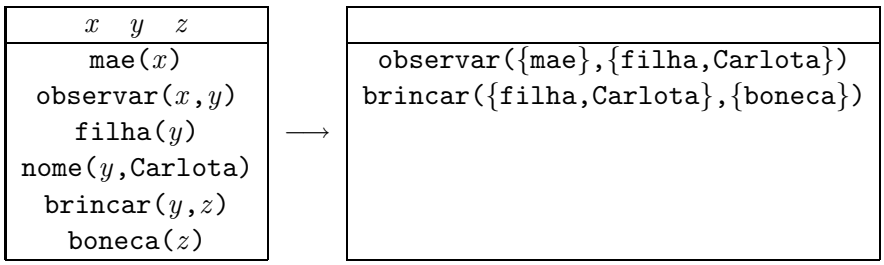

Figura 6.5: Representação gráfica da ERD correspondente à frase `A mãe observa a filha Carlota que brinca com a boneca` antes e depois da substituição dos referentes restringidos por nomes próprios e condições atómicas.

6.2.3 A Representação de uma ERD

Com as transformações descritas sobre as ERDs, a sua representação através de um grafo dirigido já permite a utilização do núcleo do grafo de produto (secção 5.4.2):

- uma condição é um vértice etiquetado com o nome da condição e um conjunto de arcos. Os arcos partem do vértice etiquetado com o nome da condição para os vértices que representam os referentes restringidos. Os arcos são etiquetados com um inteiro que indica a posição do referente na condição;
- um referente de uma condição é:
  - um vértice etiquetado pelo referente e com um grau de saída igual a zero, ou
  - um subgrafo composto por um vértice com uma etiqueta própria, igual para todos os subgrafos que representam um referente, e um conjunto de arcos de saída que ligam a cada um dos elementos do conjunto de substituição. Estes arcos são todos etiquetados com uma (a mesma) etiqueta própria;
- cada elemento do conjunto de substituição é um vértice etiquetado com o identificador da condição unária ou do nome próprio que restringem o referente representado pelo conjunto de substituição. Este vértice tem grau de saída igual a zero.

A Figura 6.6 ilustra o grafo correspondente à ERD da Figura 6.4.

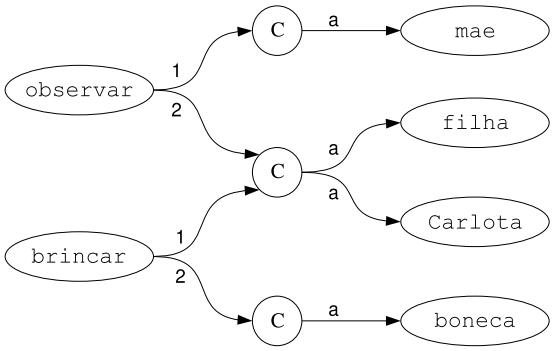

Figura 6.6: Grafo que ilustra a ERD com substituição de referentes da frase `A mãe observa a filha que brinca com a boneca`.

6.2.4 A semelhança entre ERDs

O núcleo do grafo de produto (secção 5.4.2) mede a semelhança entre ERDs através da contabilização de percursos no grafo de produto directo dos grafos factores; quanto

mais percursos existirem no grafo de produto directo, mais semelhante é a informação semântica contida nas frases.

A Figura 6.7 ilustra duas ERDs (à esquerda) e o grafo produto directo correspondente (à direita); este grafo é construído utilizando a Definição 5.14.

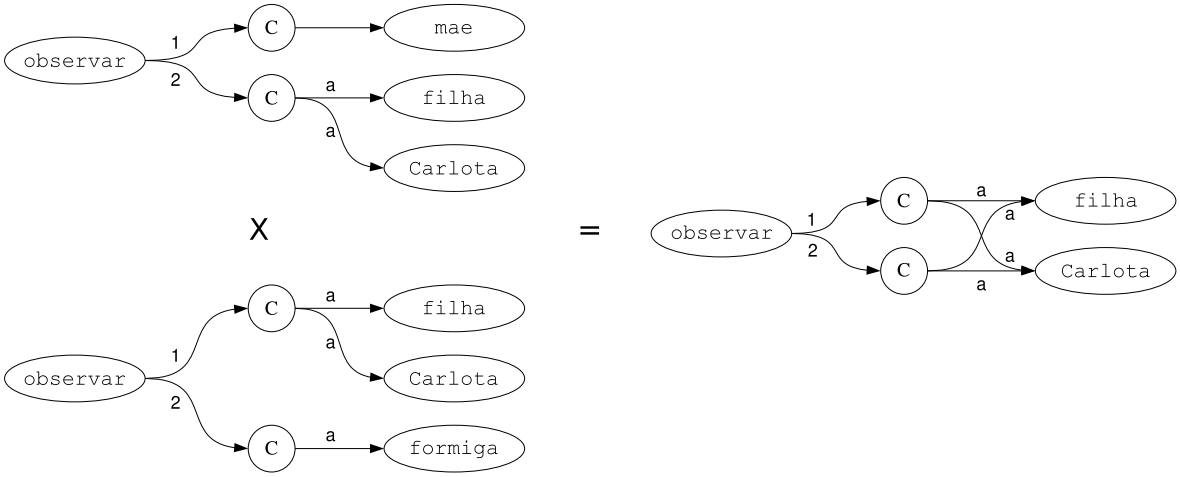

Figura 6.7: Grafos que ilustram as ERDs das frases `A mãe observa a filha Carlota` e `A filha Carlota observa a formiga` e o grafo produto directo correspondente.

A contabilização de percursos (de vários comprimentos) é realizada através de potências da matriz de adjacência $[E]_{ij}$ do grafo de produto, construída de acordo com a Definição 5.13. A Figura 6.8 representa a matriz de adjacência originada pelo grafo de produto ilustrado na Figura 6.7.

	observar	C'	C''	filha	Carlota
observar	0	1	1	0	0
C'	0	0	0	1	1
C''	0	0	0	1	1
filha	0	0	0	0	0
Carlota	0	0	0	0	0

Figura 6.8: Matriz de adjacência do grafo de produto representado na Figura 6.7.

A segunda potência desta matriz, $[E]^2_{ij}$ possui todas as componentes iguais a zero, excepto as correspondentes aos pares `observar-filha` e `observar-Carlota` cujo valor é dois; as restantes potências desta matriz, $[E]^n_{ij}$ $(n > 2)$ são matrizes nulas.

6.2.5 Representação da forma lógica de um documento

Utilizando a TRD, a forma lógica de um documento é obtida pelo conjunto de ERDs que o descrevem. Assim, definida a representação adequada da ERD para o núcleo de grafos, torna-se simples a representação do documento – é o grafo constituído pelo conjunto das suas ERDs.

No entanto, com as substituições descritas pode haver replicação de subgrafos em ERDs distintas. É então necessário retirar essas replicações alterando os arcos provenientes das várias ERDs para o mesmo subgrafo. A partir dos conjuntos iniciais de vértices $\mathcal{V}$ e arcos $\mathcal{A}$, o novo conjunto de vértices $\mathcal{V}'$ é calculado como

$$\mathcal{V}' = (\mathcal{V} - R_C - R_S) \cup A_C \cup A_S \tag{6.7}$$

com

$$\begin{aligned} R_C &= \{v : v \in \mathcal{V} \wedge (\exists u \in \mathcal{V} : v \neq u \wedge \varepsilon(v) = \varepsilon(u) = \texttt{'C'} \wedge & (6.8)\\ &\quad \wedge\, etqseg(v) = etqseg(u))\} \\ R_S &= \{v : v \in \mathcal{V} \wedge (\exists u \in R_C, \exists a \in \mathcal{A} : \Psi(a) = (u, v))\} & (6.9)\\ A_C &= \{v : v \in R_C \wedge (\neg\exists x \in A_C : x \neq v \wedge etqseg(x) = etqseg(v))\} & (6.10)\\ A_S &= \{v : v \in R_S \wedge (\exists x \in A_C, \exists a \in \mathcal{A} : \Psi(a) = (x, v))\} & (6.11) \end{aligned}$$

onde

$$\begin{aligned} etqseg(v) &= \{l : \exists x \in seg(v) \wedge \varepsilon(x) = l\} \text{ e} & (6.12)\\ seg(v) &= \{u : u \in \mathcal{V} \wedge (\exists a \in \mathcal{A} : \Psi(a) = (v, u))\} & (6.13) \end{aligned}$$

O conjunto R_C contém todos os vértices com etiqueta `'C'` que sejam vértices de entrada de subgrafos duplicados e o conjunto R_S contém os seus sucessores; o conjunto A_C contém um vértice com etiqueta `'C'` para cada grupo de subgrafos duplicados em R_C e o conjunto A_S contém os seus vértices sucessores.

O novo conjunto de arcos $\mathcal{A}'$ é

$$\mathcal{A}' = (\mathcal{A} - R_{AC}) \cup A_{AC} \tag{6.14}$$

com

$$\begin{aligned} R_{AC} &= \{a : a \in \mathcal{A} \wedge (\exists u \in R_C, \exists v \in \mathcal{V} : \Psi(a) = (u, v) \vee \Psi(a) = (v, u))\} & (6.15)\\ A_{AC} &= \{a : a \in \mathcal{A} \wedge (\exists u \in A_C, \exists v \in \mathcal{V} : \Psi(a) = (u, v) \vee \Psi(a) = (v, u))\} & (6.16) \end{aligned}$$

A Figura 6.9, ilustra a representação em grafo das frases `A mãe observa a filha Carlota` e `A filha Carlota observa a formiga` no mesmo documento[1].

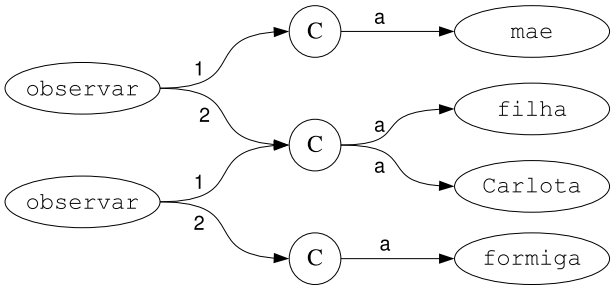

Figura 6.9: Grafo que ilustra a representação das frases `A mãe observa a filha Carlota` e `A filha Carlota observa a formiga` no mesmo documento.

Estes grafos que representam a forma lógica das frases obtida através de ERDs simples com as substituições de referentes propostas têm duas propriedades: não possuem ciclos e qualquer percurso tem no máximo comprimento dois.

[1]Estas frases encontram-se representadas individualmente na Figura 6.7, como os grafos factores do grafo de produto directo.

Capítulo 7

Avaliação da utilização de informação linguística em CT

Com o objectivo de estudar a influência de diferentes níveis de informação linguística no desempenho de classificadores de texto, este trabalho apresenta diversas representações do documento que consideram a sua informação morfológica, sintáctica e semântica.

As diversas representações são obtidas de forma encadeada utilizando as ferramentas de extracção de informação linguística descritas na secção 2.5. A Figura 7.1 ilustra o processo de obtenção dessa informação.

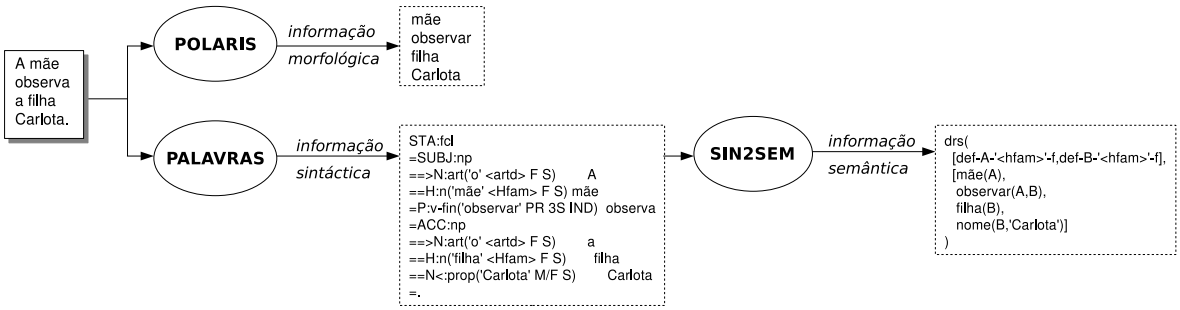

Figura 7.1: Ferramentas de extracção de informação linguística: exemplo de utilização.

Os corpora utilizados contêm texto escrito na língua Portuguesa e permitem, através da sua representação, induzir classificadores de forma automática utilizando máquinas de vectores de suporte.

Este Capítulo começa por descrever as diferentes representações do documento para cada nível de informação linguística e a caracterização dos corpora utilizados. Os classificadores de texto induzidos são avaliados na secção 7.3 e os resultados são discutidos

na secção 7.4. No final, esta avaliação para a língua Portuguesa é confrontada com avaliações realizadas noutros trabalhos.

7.1 Representação do documento

Como já referido, a grande maioria dos trabalhos de CT são aplicados a textos escritos na língua Inglesa e, embora existam alguns que utilizam o nível sub-palavra ou multi-palavra, o termo de indexação mais utilizado é, sem dúvida, a palavra (ver secção 4.1).

De seguida, apresentam-se diversas representações do documento: em algumas a unidade básica é a palavra, mas noutras a unidade básica é estruturada e representa a frase. Para além dos níveis morfológico, sintáctico e semântico, foi ainda utilizada a informação morfo-sintáctica das palavras como método de selecção de termos numa representação saco-de-palavras.

7.1.1 Informação morfológica

Primeiro considerou-se a informação morfológica com a tradicional **representação saco-de-palavras**.

Esta aproximação corresponde à representação tipicamente utilizada pelas técnicas usuais de RI e, para além de verificar a importância da lematização no processo de classificação, serve de dupla base de comparação. Por um lado permite verificar se os resultados obtidos para a língua Inglesa são válidos para a língua Portuguesa; por outro, serve para certificar a utilidade da informação sintáctica e semântica na classificação de texto.

A Figura 7.2 mostra um documento e a respectiva representação saco-de-palavras utilizando a frequência do documento[1].

Consideraram-se técnicas de normalização, selecção e ponderação de termos. Quanto à normalização fizeram-se experiências utilizando a palavra original e a sua raíz (obtida por lematização). As palavras foram, depois, seleccionadas utilizando três classes de experiências: a eliminação de *stopwords*, a função de graduação e o valor de limiar. Na ponderação dos termos seleccionados para indexação foram utilizadas combinações das três componentes usuais: documento, colecção e normalização.

[1]Tipicamente utiliza-se a notação esparsa representando, para cada documento, a frequência das palavras que nele aparecem; todas as restantes palavras que constituem termos de indexação e não aparecem no documento têm um valor igual a zero.

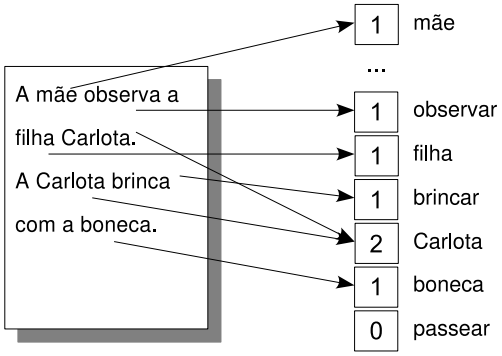

Figura 7.2: O documento original e a respectiva representação saco-de-palavras.

Como função de graduação (ver secção C.1) testou-se:

- a frequência, $\texttt{freq(t}_k\texttt{)}$;
- e a informação mútua, $\texttt{imut(t}_k\texttt{,c}_i\texttt{)}$.

Enquanto na primeira o valor da função é independente da etiqueta da classe, na segunda isso não acontece. Assim, para `imut`, o valor de cada termo é o valor máximo obtido entre todas as classes.

Como valor de limiar escolheu-se o número de palavras resultante da aplicação de um limite inferior (`l`) à frequência de cada palavra no corpus, removendo aquelas que ocorrem no máximo $l - 1$ vezes; por exemplo, quando o limite é 100, são escolhidas como termo de indexação todas as palavras que aparecem pelo menos 100 vezes no total dos documentos. Os valores de limiar variaram consoante o corpus em estudo.

Na ponderação dos termos (ver secção C.3) escolheu-se:

- a frequência do termo (`t`) para a componente documento;
- sem componente (`x`) e frequência inversa do documento (`f`) para a componente colecção;
- normalização do co-seno (`c`) para a componente normalização.

Estas opções podem ser representadas graficamente num espaço tridimensional com eixos para normalização, selecção e ponderação; as técnicas de selecção e ponderação podem, por sua vez, ser representadas noutros espaços a três dimensões (ver Figura 7.3).

7.1.2 Informação morfo-sintáctica

Para avaliar o poder discriminante das diferentes classes (ou conjunto de classes) gramaticais no processo de classificação de texto, utilizou-se a informação morfo-

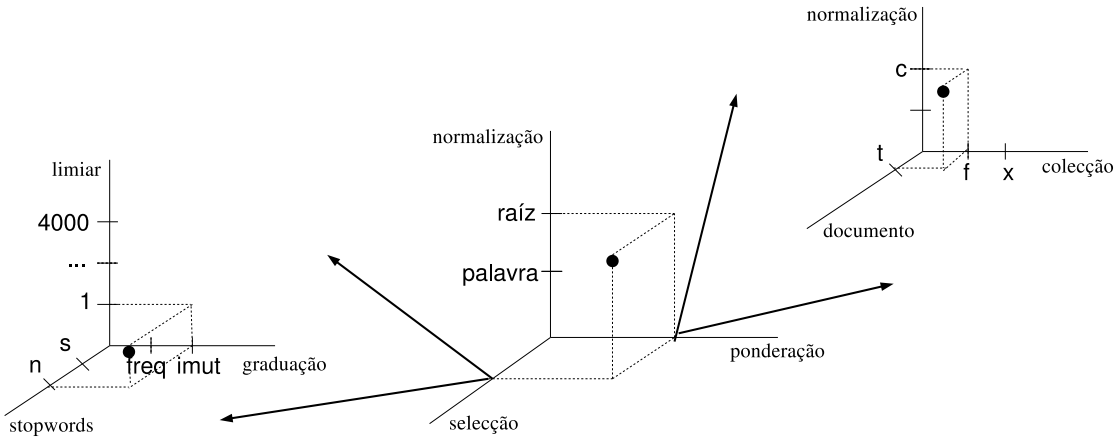

Figura 7.3: Através de espaços a três dimensões é possível visualizar graficamente o conjunto de experiências realizadas com informação morfológica. O ponto marcado corresponde à experiência: eliminação de *stopwords* com a raíz da palavra como termo de indexação, aplicação da função de graduação `imut` e valor de limiar igual a 1 e `tfidf` como medida de ponderação.

sintáctica como técnica de selecção dos termos de indexação do documento numa **representação saco-de-palavras**.

Experimentaram-se subconjuntos das classes gramaticais mais informativas: nome (`n`), nome próprio (`prop`), adjectivo (`adj`) e verbo (`v`).

7.1.3 Informação sintáctica

Enquanto existem muitos trabalhos de CT (incluindo alguns para a língua Portuguesa) que consideram informação morfológica, são relativamente poucos os que utilizam informação sintáctica.

Como referido na secção 2.4.2 a estrutura sintáctica da frase é usualmente representada através de uma árvore – a árvore sintáctica.

Representação árvore-sintáctica

A árvore sintáctica é uma árvore ordenada, e como tal, um documento pode ser representado através de um conjunto de árvores ordenadas. Para considerar a ordem das frases esse conjunto pode, por sua vez, ser representado através de uma árvore ordenada. A esta nova representação do documento deu-se o nome de **representação árvore-sintáctica**.

Nesta representação cada documento é reproduzido utilizando de uma árvore cujos

filhos da raíz são as árvores sintácticas de cada uma das suas frases. No entanto, a árvore construída não é composta pela árvore sintáctica completa, mas contém apenas os nós das seguintes classes gramaticais[2]: o nome (`n`), o nome próprio (`prop`), o adjectivo (`adj`), o verbo (`v`), o pronome (`pron`) e o advérbio (`adv`). As folhas da árvore são as palavras do documento, que podem ser as originais ou substituidas pela sua raíz. Neste trabalho utilizou-se a raíz.

A Figura 7.4 mostra um documento e a respectiva representação árvore-sintáctica.

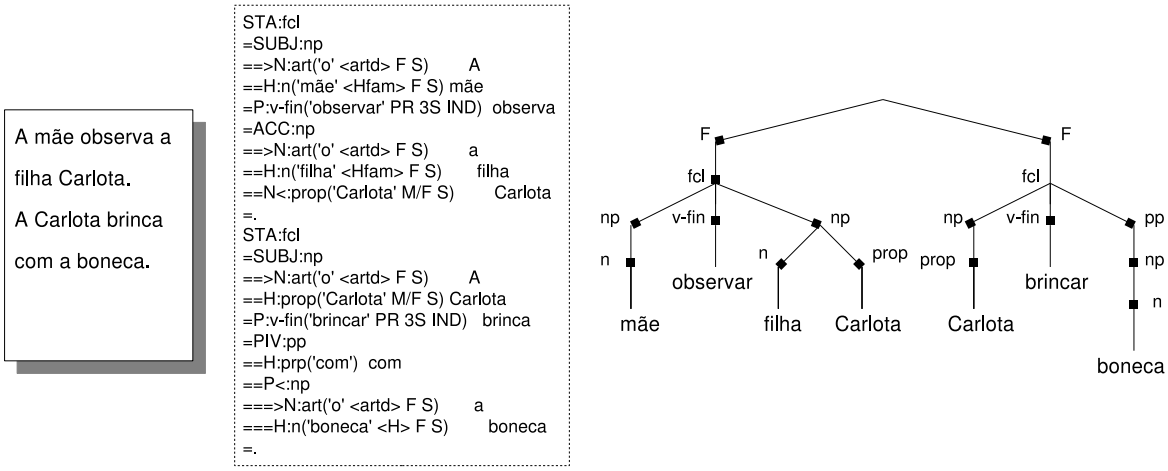

Figura 7.4: O documento original, as árvores sintácticas geradas pela ferramenta PALAVRAS e a respectiva representação árvore-sintáctica.

Outras representações

Com informação retirada das árvores sintácticas das frases utilizaram-se ainda outras duas representações do documento: a representação sequência-de-palavras e a representação saco-de-palavras. Estas representações servem para verificar qual o poder discriminante da representação estruturada com informação sintáctica, ou seja, se a representação árvore-sintáctica é benéfica relativamente a outras representações que não consideram a estrutura sintáctica da frase.

Na **representação sequência-de-palavras** o documento é reproduzido utilizando uma lista ordenada de palavras. Foi obtida a partir da representação árvore-sintáctica utilizando uma árvore ordenada onde os filhos da raíz são as palavras do documento.

A Figura 7.5 mostra um documento e a respectiva representação sequência-de-palavras.

[2]Não foram consideradas todas as classes porque essa representação tornaria o processo de comparação de documentos muito lento já que o comprimento do documento seria muito mais longo.

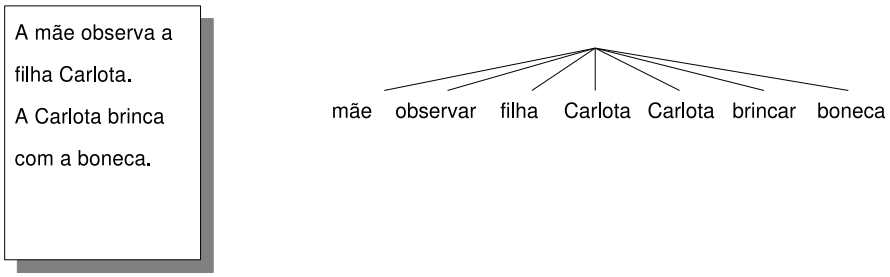

Figura 7.5: O documento original e a respectiva representação sequência-de-palavras.

Esta representação foi utilizada como nível intermédio entre as representações árvore-sintáctica e saco-de-palavras. Enquanto a primeira considera a estrutura dentro da frase, a representação saco-de-palavras não considera nenhuma estrutura do documento; a representação sequência-de-palavras não considera a estrutura da frase, mas permite estudar a influência da ordem das palavras na classificação de textos, já que considera a sequência pela qual as palavras aparecem no documento.

A representação saco-de-palavras utiliza as palavras retiradas da representação árvore-sintáctica.

7.1.4 Informação semântica

A informação semântica é representada utilizando a forma lógica (ver secção 2.4.3). Esta, por sua vez, pode ser produzida directamente a partir da estrutura sintáctica da frase. Na Teoria da Representação do Discurso (ver secção 2.5.3), a forma lógica de uma frase é representada através de uma Estrutura de Representação do Discurso (ERD) que pode ser obtida a partir da árvore sintáctica correspondente.

Representação estrutura-do-discurso

Como referido na secção 6.2.5, o grafo que representa a informação semântica do documento não possui ciclos. Esta propriedade torna possível a sua transformação numa estrutura menos complexa e que permite a utilização de funções de núcleo mais eficientes computacionalmente – uma estrutura em árvore.

Esta transformação compreende dois pontos:

- a criação de um novo vértice (com etiqueta própria igual para todos os documentos) que constitui a raíz da árvore; dele partem arcos dirigidos que se ligam aos vértices que identificam as condições das ERD;

- a replicação de subgrafos para cada condição quando duas ou mais condições se ligam ao mesmo vértice (vértice com grau de entrada maior ou igual a dois).

Com esta transformação obtém-se uma árvore cuja raíz representa o documento, os nós de profundidade um representam as condições das ERDs e os filhos de cada um desses nós, os referentes restringidos pela condição. Para os referentes que foram substituidos por nomes próprios e/ou condições atómicas existe ainda mais um nível que representa o conjunto dessas substituições.

Como é necessário indicar a posição de cada referente nas condições a árvore que representa o documento deve ser ordenada. No entanto, uma vez que as condições das ERDs constituem um conjunto, torna-se necessário ordená-las alfabeticamente. O mesmo tipo de ordenação deve ser realizado sobre os vértices que representam cada um dos elementos do conjunto de substituição.

A formalização desta operação é mais simples se considerarmos a representação do documento como a união das ERDs correspondentes a cada uma das suas frases – um grafo disconexo G com um conjunto de vértices $\mathcal{V}$ e um conjunto de arcos $\mathcal{A}$. Assim, a árvore obtida é um grafo acíclico dirigido sem etiquetas nos arcos G' com um conjunto de vértices $\mathcal{V}'$ e um conjunto de arcos $\mathcal{A}'$ onde

$$\mathcal{V}' = \mathcal{V} \cup \{raiz\} \tag{7.1}$$
$$\mathcal{A}' = \mathcal{A} \cup \{a : a \in \mathcal{A} \wedge (\exists v \in \mathcal{V} : \Psi(a) = (raiz, v) \wedge d_G^-(v) = 0)\} \tag{7.2}$$

Sendo

$$C = \{raiz\} \cup \{v : v \in \mathcal{V} \wedge \varepsilon(v) = \texttt{'C'}\} \tag{7.3}$$
$$D = \mathcal{V}' - C \tag{7.4}$$

a ordenação da árvore é respeitada ao garantir as seguintes condições:

$$\forall v \in C,\ \forall 1 \leq i < j \leq d^+(v), \tag{7.5}$$
$$\varepsilon(filho_i(v)) \leq \varepsilon(filho_j(v))$$
$$\forall v \in D,\ \forall 1 \leq i \leq d^+(v),\ \exists^1 u \in \mathcal{V}, \tag{7.6}$$
$$u = filho_i(v) \wedge \exists a \in \mathcal{A}, \Psi(a) = (v, u) \wedge \varepsilon(a) = i$$

Enquanto a primeira condição garante a ordenação alfabética das condições das ERDs e dos conjuntos de substituição de cada referente, a segunda garante o posicionamento correcto dos argumentos das condições ao utilizar a etiqueta do arco no grafo G.

A esta representação do documento em árvore que incorpora a sua forma lógica e que permite avaliar o impacto da informação semântica no processo de classificação deu-se o nome de **representação estrutura-do-discurso**. A Figura 7.6 ilustra a representação estrutura-do-discurso correspondente ao grafo ilustrado na Figura 6.9.

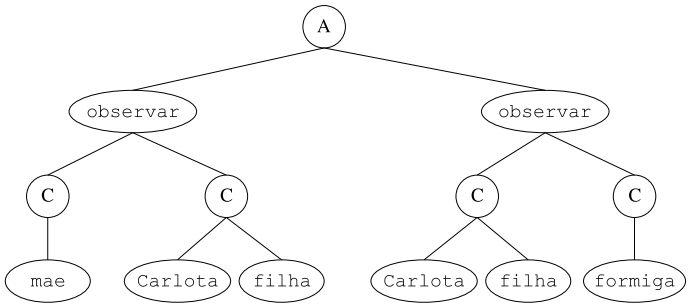

Figura 7.6: A representação estrutura-do-discurso com as frases `A mãe observa a filha Carlota` e `A filha Carlota observa a formiga` no mesmo documento.

Na secção 6.2.2 propõem-se dois tipos de substituição de referentes; cada um origina representações estrutura-do-discurso distintas. A Figura 7.7 mostra um exemplo com o texto original, as ERDs geradas pela aplicação SIN2SEM e a respectiva representação estrutura-do-discurso quando aplicadas as substituições de nomes próprios e condições atómicas.

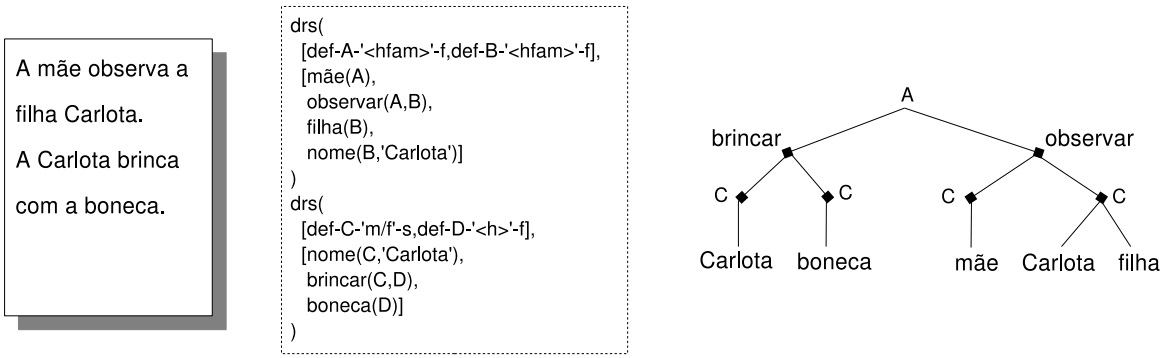

Figura 7.7: O documento original, as ERDs geradas e a respectiva representação estrutura-do-discurso. Esta representação do documento utiliza a substituição de nomes próprios e condições atómicas.

Outras representações

Com a informação retirada das árvores representantes da forma lógica dos documentos utilizou-se ainda uma representação saco-de-palavras para estudar a influência da

estrutura da representação no processo de classificação. Para a construir, extraíram-se as palavras que aparecerem na representação estrutura-do-discurso.

7.2 Corpora

Para testar o poder discriminativo de cada representação utilizaram-se três corpora escritos na língua Portuguesa: um corpus de documentos jurídicos com decisões da Procuradoria Geral da República (corpus `PGR`) e dois corpora de notícias da imprensa escrita: o jornal brasileiro Folha de S. Paulo (`Folha`) e o jornal diário Público (`Publico`).

7.2.1 PGR

Os documentos do corpus `PGR` representam as decisões da Procuradoria Geral da República desde 1940. O corpus original é composto por 8151 documentos que foram classificados manualmente por peritos num conjunto de classes (classificação multi-etiqueta) pertencentes a uma taxonomia de conceitos jurídicos com cerca de 6000 termos. Após uma avaliação preliminar, constatou-se que apenas cerca de 3000 conceitos foram utilizados e que destes, apenas 801 contêm dez ou mais documentos.

Do corpus completo, apenas 6773 documentos contêm pelo menos uma palavra para todas as experiências realizadas. Utilizou-se este conjunto de documentos e retiraram-se as cadeias de caracteres com algarismos num total de 34339 cadeias distintas. Contabilizaram-se então 68877 palavras distintas e médias por documento de 1608 palavras (símbolos) e 366 palavras distintas (tipos). A Figura 7.8 mostra histograma do número de documentos que pertencem às 50 classes mais comuns (aquelas com mais de 75 documentos).

Este corpus foi utilizado para verificar a existência de diferenças no desempenho de classificadores aplicados a corpora de carácter geral e de domínios específicos. O estudo efectuado recaiu sobre as 10 classes mais comuns e realizaram-se as experiências com a representação saco-de-palavras para os níveis morfológico e morfo-sintáctico (secções 7.1.1 e 7.1.2). A Tabela 7.1 contém a descrição e número de documentos para essas classes.

Para os restantes níveis não se utilizou este corpus porque o seu texto apresenta construções sintácticas e semânticas de tal forma complexas que tornaram inviável a sua representação de forma automática. Optou-se então por continuar o estudo apenas com os documentos jornalísticos já que estes possuem construções muito mais simples.

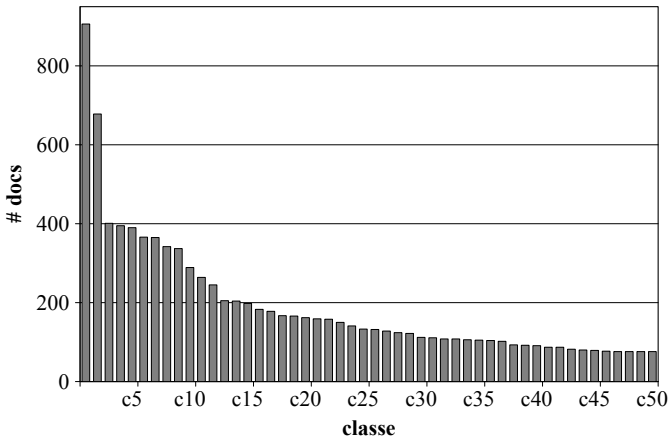

Figura 7.8: Corpus PGR: histograma do número de documentos para as 50 classes mais comuns.

classe	nome	# docs
c_1	aposentação	342
c_2	competência	336
c_3	deficiente das forças armadas	678
c_4	estado da Índia	395
c_5	exemplar conduta moral e cívica	289
c_6	funcionário público	365
c_7	louvor	366
c_8	militar	388
c_9	pensão por serviços excepcionais	906
c_{10}	prisioneiro de guerra	401

Tabela 7.1: Corpus PGR: as 10 classes mais comuns e o seu número de documentos.

7.2.2 Folha de S. Paulo

Este corpus constitui um subconjunto do corpus NILC[3] (Núcleo Inter-Institucional de Linguística Computacional) e contém 855 documentos do ano de 1994 [Silva et al. 2004]. Existem 5 classes (`desporto`, `imoveis`, `informatica`, `politica` e `turismo`), cada uma com 171 documentos; como cada documento pertence a uma única classe, estamos em presença de um problema multi-classe. O corpus, depois de retiradas as cadeias de caracteres com algarismos (num total de 1367), contém 19522 palavras distintas e uma média por documento de 215 símbolos e 124 tipos. Utilizou-se este corpus nas experiências com informação morfológica e morfo-sintáctica (secções 7.3.1 e 7.3.2).

Para os outros níveis de informação (secções 7.3.3 e 7.3.4) utilizou-se um subconjunto, `Folha2`, composto por 844 documentos. Esta decisão deve-se ao facto do analisador sintáctico não ter conseguido fazer uma análise correcta de alguma(s) frase(s) dos restantes 11 documentos. A distribuição pelas classes continuou praticamente equitativa (168 para a classe `politica` e 169 para as restantes classes), existindo no total 19417 palavras distintas e uma média por documento de 135 símbolos e 101 tipos.

7.2.3 Público

O corpus `Publico` contém as notícias diárias de 9 secções (problema multi-classe) do jornal Público dos anos de 1994 e 1995 num total de 101646 documentos. Depois de retiradas as cadeias de caracteres com algarismos (19403 cadeias distintas), o corpus contém 282657 palavras distintas e uma média por documento de 512 símbolos e 254 tipos. Este corpus foi utilizado nas experiências com informação morfológica e morfo-sintáctica.

Uma vez que a ferramenta SIN2SEM (utilizada para obter a representação semântica) não consegue processar algumas árvores sintácticas produzidas pela ferramanta PALAVRAS,sendo necessário verificar o resultado da análise sintáctica para cada documento, restringiu-se o estudo apenas a um mês para os restantes níveis de informação. Ao utilizar as notícias do mês de Outubro de 2005, obteve-se um corpus com 4290 documentos, o `Publico9510`. Nele existem 70743 palavras distintas e médias por documento de 299 símbolos e 213 tipos.

A Tabela 7.2 indica, para os corpora `Publico` e `Publico9510`, o nome das secções, o número de documentos em cada uma e a correspondente percentagem relativa ao corpus.

[3]Disponível em `http://www.nicl.icmc.usp.br/nicl/`.

		Publico		Publico9510	
classe	secção	# docs	%	# docs	%
c_1	ciências, tecnologia e educação	6285	6.2	287	6.7
c_2	cultura	15706	15.5	624	14.5
c_3	desporto	10109	9.9	441	10.3
c_4	diversos	8370	8.2	346	8.1
c_5	economia	13483	13.3	449	10.5
c_6	local	17447	17.2	913	21.3
c_7	mundo	9561	9.4	399	9.3
c_8	nacional	9348	9.2	441	10.3
c_9	sociedade	11340	11.2	390	9.1

Tabela 7.2: Corpora `Publico` e `Publico9510`: número de documentos por classe e a respectiva proporção relativa ao corpus completo.

7.3 Resultados

Seguindo a mesma estrutura da secção 7.1, esta secção apresenta os resultados obtidos nos diversos níveis de informação linguística, começando pelos níveis mais simples e terminando naquele que constitui uma aproximação inovadora na classificação de textos – o nível de informação semântica.

Utilizaram-se aproximações e ferramentas distintas consoante o nível de informação:

- Nos níveis morfológico e morfo-sintáctico utilizou-se:
  - a ferramenta WEKA (secção 3.5.1) com a respectiva implementação das máquinas de vectores de suporte, para permitir a comparação com outros algoritmos de classificação, nomeadamente o algoritmo naïve de Bayes e árvores de decisão. Nesta ferramenta utilizaram-se todos os parâmetros por omissão.
  - a técnica de validação-cruzada com 10-pastas para os corpora `PGR` e `Folha` (ver secção 3.3.1) e a aproximação treino-e-teste para o corpus `Publico` (33% de exemplos utilizados para teste).
- Nos níveis sintáctico e semântico utilizou-se:
  - uma implementação de máquinas de vectores de suporte que permite a utilização de dados estruturados – a ferramenta SVM^{light}-TK (secção 3.5.3). Escolheu-se, para ambas os corpora, o núcleo linear para a representação saco-de-palavras e o núcleo de subconjuntos de árvores para as restantes representações com $L = 0.001$ e $c = 10$ depois de se realizarem os seguintes testes:
    - para o núcleo para a representação saco-de-palavras experimentou-se o núcleo linear e o polinomial (grau 2, 3, 4 e 5);

  - para o núcleo das restantes representações testou-se núcleo de subárvores e núcleo de subconjuntos de árvores (ver secção 3.5.3);
  - para factor de declínio[4], L, do núcleo estruturado, testaram-se valores de $L \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.3\}$, e
  - para o compromisso entre o erro de treino e a margem, C, testaram-se valores de $C \in \{0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50\}$.
- a aproximação treino-e-teste para os corpora `Folha2` e `Publico9510` (33% de exemplos utilizados para teste).

Para avaliar o desempenho dos classificadores estimou-se as micro- ($^{\mu}$) e macro-médias (^{M}) da precisão (π), cobertura (ρ) e da medida combinada f_1 (ver secção 3.4.3). Os testes de significância consideram um nível de confiança de 95%.

Cada subsecção apresenta as experiências realizadas e os resultados obtidos em diferentes representações para cada corpus estudado, terminando com a escolha de uma delas como melhor representante do respectivo nível de informação linguística.

7.3.1 Informação morfológica

Utilizando a informação morfológica com a representação saco-de-palavras testaram-se todas as combinações de normalização, função de graduação e ponderação referidas na secção 7.1.1 (a lematização foi utilizada juntamente com a eliminação de *stopwords*).

O número de termos (#) e as médias de símbolos (μ_s) e tipos (μ_t) por documento para os documentos originais (`orig`), e os respectivos valores ao eliminar as *stopwords* (`stop`) e aplicar a lematização (`lema`) encontram-se apresentados na Tabela 7.3.

	PGR			Folha			Publico		
	#	μ_s	μ_t	#	μ_s	μ_t	#	μ_s	μ_t
`orig`	68877	1608	366	19522	215	124	282657	512	254
`stop`	68679	963	331	19352	134	100	282450	304	214
`lema`	42399	921	258	13317	128	91	205155	288	189

Tabela 7.3: Número de palavras (#) e médias por documento de símbolos (μ_s) e tipos (μ_t) antes e depois da eliminação de *stopwords* e lematização (corpora `PGR`, `Folha` e `Publico`).

Os valores de limiar (`l`) testados dependeram do corpus; a Tabela 7.4 indica esses valores e o correspondente número de palavras (antes da eliminação de *stopwords* e lematização[5]).

[4]Do inglês, *decay factor*.

[5]Para as experiências `stop` e `lema` utilizou-se um número de termos igual ao obtido para a

PGR	l_{50}	l_{100}	l_{200}	l_{300}	l_{400}	l_{500}	l_{600}	l_{700}	l_{800}	l_{900}
	9477	6435	4236	3226	2577	2198	1897	1678	1514	1369
Folha	l_5	l_{10}	l_{20}	l_{30}	l_{40}	l_{50}	l_{60}	l_{70}	l_{80}	l_{90}
	4420	2315	1153	745	529	397	334	265	222	199
Publico	l_{50}	l_{100}	l_{200}	l_{300}	l_{400}	l_{500}	l_{1000}	l_{2000}	l_{3000}	l_{4000}
	36398	24616	16060	12241	10010	8547	4920	2777	1893	1419

Tabela 7.4: Número de termos obtidos para diferentes valores de limiar (corpora PGR, Folha e Publico).

Para o corpus Folha foram realizadas experiências para todas as combinações possíveis das opções descritas num total de 132; para o corpus PGR não se realizaram as experiências com as palavras originais totalizando 88 experiências; para o corpus Publico, devido à sua dimensão, além de não ter realizado as experiências com as palavras originais não se realizou a experiência stop-l_1. A Tabela 7.5 apresenta os valores mínimo, máximo, média e desvio padrão das micro- e macro-médias.

		PGR			Folha			Publico		
		π	ρ	f_1	π	ρ	f_1	π	ρ	f_1
micro-média	min	.497	.742	.606	.863	.863	.863	.787	.787	.787
	max	.878	.789	.816	.982	.982	.982	.843	.843	.843
	média	.837	.768	.800	.947	.947	.947	.824	.824	.824
	desvio	.043	.013	.023	.026	.026	.026	.012	.012	.012
macro-média	min	.491	.687	.559	.865	.863	.864	.779	.770	.774
	max	.799	.741	.758	.983	.982	.982	.842	.831	.836
	média	.768	.716	.734	.947	.947	.947	.821	.810	.815
	desvio	.034	.015	.022	.026	.026	.026	.013	.013	.013

Tabela 7.5: Valores mínimo, máximo, médio e desvio padrão das micro- e macro-médias (corpora PGR, Folha e Publico).

A melhor representação do documento foi escolhida em duas fases:

1. analisou-se o conjunto das melhores experiências para cada medida. Este conjunto é formado pelas experiências com resultados equivalentes[6] ao valor máximo obtido.

2. Entre aquelas que possuíam várias medidas de desempenho nesse conjunto, escolheu-se a representação com lematização com maior valor de limiar (menos termos).

As subsecções seguintes descrevem os resultados obtidos para cada corpus.

experiência orig.

[6]Equivalentes do ponto de vista de não existirem diferenças significativas.

Corpus PGR. Este corpus apresenta grandes diferenças no conjunto das melhores experiências consoante a medida em estudo. Estas diferenças são tanto em número como em variedade, não existindo uma única experiência no conjunto das melhores para todas as medidas analisadas: os conjuntos da precisão e da cobertura são disjuntos.

A Tabela 7.6 contabiliza, para cada medida de desempenho, o número de experiências pertencentes ao conjunto das melhores, discriminando, também, esse valor para cada opção de normalização e funções de graduação e ponderação (por exemplo, f_1^M tem 16 melhores: 7 para a experiência `stop` e 9 para a `lema`; 14 para a experiência `freq` e 2 para `imut`; 13 para `txc` e 3 para `tfc`).

	π^μ	ρ^μ	f_1^μ	π^M	ρ^M	f_1^M
total	5	22	34	14	21	16
`stop`	2	10	14	4	7	7
`lema`	3	12	20	10	14	9
`freq`	2	21	19	2	20	14
`imut`	3	1	15	12	1	2
`txc`	1	16	26	10	16	13
`tfc`	4	6	8	4	5	3
$\mathtt{l}_1$	0	2	0	0	2	0
$\mathtt{l}_{50}$	0	2	0	0	2	0
$\mathtt{l}_{100}$	0	2	0	0	2	0
$\mathtt{l}_{200}$	0	4	1	0	3	0
$\mathtt{l}_{300}$	0	4	2	1	4	3
$\mathtt{l}_{400}$	0	2	3	1	2	2
$\mathtt{l}_{500}$	0	2	6	1	2	3
$\mathtt{l}_{600}$	0	2	6	1	2	3
$\mathtt{l}_{700}$	0	1	5	3	1	2
$\mathtt{l}_{800}$	0	1	5	3	1	2
$\mathtt{l}_{900}$	5	0	6	4	0	1

Tabela 7.6: Corpus `PGR`: número de experiências que não apresentam diferenças significativas com o melhor resultado.

Embora os resultados variem para cada medida de desempenho, a partir da Tabela é possível concluir, que as medidas `freq` e `txc` são as mais estáveis. As experiências com todas as palavras (valor de limiar $\mathtt{l}_1$) apenas possuem valores no conjunto das melhores para a cobertura (experiências `lema-freq-txc` e `lema-imut-txc`).

Analisando os conjuntos de melhores foi também possível concluir que, no que respeita à cobertura e f_1 (micro- e macro-médias), estavam incluidas as experiências `stop-freq-txc` com valores de limiar entre $\mathtt{l}_{600}$ e $\mathtt{l}_{800}$ e `lema-freq-txc` com valores de limiar entre $\mathtt{l}_{400}$ e $\mathtt{l}_{800}$. Para estas medidas, o valor de limiar mais alto existente no conjunto é $\mathtt{l}_{800}$. A Tabela 7.7 apresenta os resultados destas experiências[7].

[7]Os valores a negrito correspondem a valores sem diferenças significativas quando comparados com

		π^{μ}	ρ^{μ}	f_1^{μ}	π^{M}	ρ^{M}	f_1^{M}
stop	$\mathtt{l}_{600}$	.831	**.782**	**.806**	.771	**.733**	**.750**
	$\mathtt{l}_{700}$	.833	**.780**	.806	.772	**.731**	.749
	$\mathtt{l}_{800}$	.837	**.781**	**.808**	.773	**.731**	**.749**
lema	$\mathtt{l}_{400}$	.830	**.785**	**.807**	.771	**.735**	**.751**
	$\mathtt{l}_{500}$	.834	**.785**	**.808**	.773	**.734**	**.751**
	$\mathtt{l}_{600}$	.844	**.786**	**.814**	.781	**.736**	**.755**
	$\mathtt{l}_{700}$	.846	**.789**	**.816**	.786	**.739**	**.758**
	$\mathtt{l}_{800}$	.846	**.782**	**.813**	.782	**.732**	**.753**

Tabela 7.7: Corpus PGR: valores das medidas de desempenho para as melhores experiências (no que respeita à cobertura e f_1).

Os melhores valores para a precisão ($\pi^{\mu} \approx .870$, $\pi^{M} \approx .799$) foram obtidos para as experiências com valor de limiar $\mathtt{l}_{900}$; no entanto, estas experiências possuem valores mais baixos de cobertura e f_1 (por exemplo, $\rho^{M} \approx 0.70$ e $f_1^{M} \approx 0.73$).

Considerando os valores da Tabela 7.7 a experiência escolhida como representante é lema-freq-txc-$\mathtt{l}_{800}$.

Corpus Folha. Ao contrário do corpus PGR, este apresenta o mesmo conjunto de melhores experiências (num total de 19) para todas as medidas de desempenho. Todas estas experiências têm a medida de graduação imut e são:

- todas as experiências com valor de limiar $\mathtt{l}_5$ e $\mathtt{l}_{10}$;
- as experiências stop ou lema com valor de limiar $\mathtt{l}_{20}$;
- as experiências lema com txc e valor de limiar $\mathtt{l}_{30}$ e
- as experiências lema com tfc e valores de limiar $\mathtt{l}_{30}$ e $\mathtt{l}_{40}$.

A Tabela 7.8 apresenta os valores obtidos para algumas destas experiências. Do conjunto das melhores escolheu-se como representante lema-imut-tfc-$\mathtt{l}_{40}$.

	π^{μ}	ρ^{μ}	f_1^{μ}	π^{M}	ρ^{M}	f_1^{M}
orig-txc-$\mathtt{l}_{10}$	.975	.975	.975	.976	.975	.975
stop-tfc-$\mathtt{l}_{20}$	.974	.974	.974	.974	.974	.974
lema-txc-$\mathtt{l}_{30}$	.974	.974	.974	.974	.974	.974
lema-tfc-$\mathtt{l}_{40}$	.975	.975	.975	.976	.975	.975

Tabela 7.8: Corpus Folha: valores das medidas de desempenho para algumas das melhores experiências.

o melhor valor obtido.

Corpus Publico. Este corpus também possui o mesmo conjunto de melhores para todas as medidas de desempenho[8] (cinco experiências no total[9]):

- as experiências `lema` e valor de limiar $\mathtt{l}_1$;
- as experiências `stop` com `freq` e valor de limiar $\mathtt{l}_{50}$ e
- a experiência `stop` com `imut`, `tfc` e valor de limiar $\mathtt{l}_{50}$;

A Tabela 7.9 apresenta esses valores. Considerando-os, existem duas experiências que satisfazem o critério para a escolha da representante: escolheu-se a experiência `lema-freq-tfc-`$\mathtt{l}_1$ por utilizar a mesma medida de ponderação do corpus `Folha`.

	π^μ	ρ^μ	f_1^μ	π^M	ρ^M	f_1^M
`lema-freq-txc-`$\mathtt{l}_1$	**.840**	**.840**	**.840**	**.839**	**.828**	**.833**
`lema-freq-tfc-`$\mathtt{l}_1$	**.843**	**.843**	**.843**	**.842**	**.831**	**.836**
`stop-freq-txc-`$\mathtt{l}_{50}$	**.839**	**.839**	**.839**	**.837**	.826	**.831**
`stop-freq-tfc-`$\mathtt{l}_{50}$	**.840**	**.840**	**.840**	**.838**	**.828**	**.832**
`stop-imut-tfc-`$\mathtt{l}_{50}$	**.840**	**.840**	**.840**	**.839**	**.828**	**.833**

Tabela 7.9: Corpus `Publico`: valores das medidas de desempenho para as melhores experiências.

Outras experiências

Sobre os corpora `PGR` e `Folha` foram realizadas algumas experiências iniciais que utilizaram outras medidas de graduação e ponderação. No entanto, estas originaram classificadores cujo desempenho foi muito inferior ao aqui apresentado. Realizaram-se ainda experiências que visaram comparar a eficiência e o desempenho das máquinas de vectores de suporte com outros algoritmos de aprendizagem convencionais, nomeadamente o algoritmo naïve de Bayes e árvores de decisão. Os resultados podem ser consultados em [Gonçalves and Quaresma 2005a] e [Gonçalves and Quaresma 2003], respectivamente.

Como referido no início da secção (ver página 112), os níveis sintáctico e semântico utilizam outra implementação de máquinas de vectores de suporte, subconjuntos dos corpora `Folha` e `Publico` e uma avaliação do classificador distinta. Realizaram-se, então, para os corpora `Folha2` e `Publico` as experiências correspondentes à representante da informação morfológica utilizando o mesmo cenário. A Tabela 7.10 apresenta esses resultados.

[8] Excepto para a ρ^M em que a experiência `stop-freq-txc-`$\mathtt{l}_{50}$ não pertence ao conjunto.
[9] Para o valor de limiar $\mathtt{l}_1$ as experiências `freq` e `imut` são iguais.

	π^{μ}	ρ^{μ}	f_1^{μ}	π^{M}	ρ^{M}	f_1^{M}
Folha2	.976	.976	.976	.977	.976	.976
Publico9510	.855	.855	.855	.854	.840	.844

Tabela 7.10: Valores das medidas de desempenho para as experiências com informação morfológica (corpora Folha2 e Publico9510).

7.3.2 Informação morfo-sintáctica

Como referido na secção 7.1.2, utilizou-se a informação morfo-sintáctica para verificar o poder discriminativo das classes gramaticais no processo de classificação. Consideraram-se as classes gramaticais mais informativas – nome (n), nome próprio (prop), adjectivo (adj) e verbo (v), utilizando uma representação saco-de-palavras.

A Tabela 7.11 apresenta o número de termos (#) e as médias de símbolos (μ_s) e tipos (μ_t) por documento para cada uma das classes gramaticais consideradas.

	PGR			Folha			Publico		
	#	μ_s	μ_t	#	μ_s	μ_t	#	μ_s	μ_t
n	20384	423	110	5428	52	37	12315	111	79
prop	14560	90	27	5517	26	18	9627	48	30
adj	7270	115	41	1816	11	9	5744	31	26
v	8899	173	72	2488	23	19	11733	68	58

Tabela 7.11: Número de palavras (#) e médias por documento de símbolos (μ_s) e tipos (μ_t) para cada classe gramatical (corpora PGR, Folha e Publico).

Como ponto de partida para estas experiências, utilizou-se a experiência representante da informação morfológica; utilizando a mesma normalização, função de graduação e ponderação da experiência base, experimentaram-se as 15 combinações possíveis de classes gramaticais.

Corpus PGR. Os resultados obtidos para cada conjunto de classes gramaticais, utilizando como base a experiência lema-freq-txc-l$_{800}$ encontram-se na Tabela 7.12[10].

A partir desta Tabela é possível afirmar que apenas a combinação dos nomes com outras duas ou três classes gramaticais (n+prop+adj, n+adj+v, n+prop+v e n+prop+adj+v) consegue produzir resultados que, juntamente com a experiência base, pertencem ao conjunto dos melhores (para ρ, f_1 e π^{M}).

Realizou-se, então, mais um conjunto de experiências reduzindo o número de termos:

[10]Os valores a negrito correspondem valores sem diferenças significativas quando comparados com o melhor valor representado.

	π^{μ}	ρ^{μ}	f_1^{μ}	π^{M}	ρ^{M}	f_1^{M}
lema-freq-txc-l_{800}	.846	**.782**	**.813**	**.782**	**.732**	**.753**
adj	.855	.678	.757	.684	.605	.627
n	.875	.762	**.814**	**.793**	.709	.738
prop	.852	.579	.689	.666	.538	.579
v	.874	.706	.781	.757	.643	.679
n+adj	.861	.761	**.808**	.779	.706	.734
n+prop	.870	.771	**.817**	**.793**	.717	**.745**
n+v	.860	.770	**.812**	**.785**	.718	**.745**
prop+adj	**.888**	.691	.778	.707	.621	.647
adj+v	.874	.722	.791	.765	.660	.695
prop+v	.877	.719	.790	.768	.659	.693
n+prop+adj	.861	**.776**	**.816**	**.788**	**.723**	**.748**
n+adj+v	.855	**.775**	**.813**	**.785**	**.722**	**.748**
n+prop+v	.856	**.782**	**.817**	**.788**	**.731**	**.754**
prop+adj+v	.862	.739	.796	.763	.678	.708
n+prop+adj+v	.852	**.783**	**.816**	**.785**	**.732**	**.753**

Tabela 7.12: Corpus PGR: valores das medidas de desempenho para as experiências com conjuntos de classes gramaticais.

- Para os conjuntos n+prop+adj, n+adj+v, n+prop+v e n+prop+adj+v experimentaram-se os valores de limiar l_{1000}, l_{1200} e l_{1400} (correspondentes a 1259, 1074 e 959 termos, respectivamente), e

- para os valores de limiar l_{1500}, l_{1600} e l_{1700} (correspondentes a 869, 830 e 793 termos, respectivamente) utilizou-se apenas n+prop+adj+v por ser a única com todas as medidas de desempenho no conjunto dos melhores para l_{1400}.

A Tabela 7.13 apresenta os resultados dessas experiências. Por questões de legibilidade, incluem-se as melhores experiências da informação morfo-sintáctica com l_{800}. Considerando estes valores, podemos escolher a experiência n+prop+adj+v-l_{1400} para representante da informação morfo-sintáctica já que possui todas as medidas no conjunto das melhores.

Corpus Folha. Os resultados obtidos para cada conjunto de classes gramaticais (utilizando como base lema-imut-tfc-l_{40}) encontram-se discriminados na Tabela 7.14.

Para este corpus, é possível afirmar que apenas a combinação dos nomes com nomes próprios com e sem adjectivos e a combinação das quatro classes gramaticais (n+prop, n+prop+adj e n+prop+adj+v) consegue valores equivalentes à experiência base.

Assim sendo, realizou-se mais um conjunto dessas experiências para maiores valores de limiar. Experimentou-se l_{50}, l_{60} e l_{70}. A Tabela 7 apresenta esses valores incluindo, os

		π^{μ}	ρ^{μ}	f_1^{μ}	π^{M}	ρ^{M}	f_1^{M}
$\mathtt{l}_{800}$	n+prop+adj	.861	**.776**	**.816**	.788	**.723**	**.748**
	n+adj+v	.855	**.775**	**.813**	.785	**.722**	**.748**
	n+prop+v	.856	**.782**	**.817**	.788	**.731**	**.754**
	n+prop+adj+v	.852	**.783**	**.816**	.785	**.732**	**.753**
$\mathtt{l}_{1000}$	n+prop+adj	.867	**.775**	**.819**	**.795**	.718	**.746**
	n+adj+v	.860	.770	**.812**	.785	.716	**.744**
	n+prop+v	.861	**.772**	**.814**	.784	.718	**.744**
	n+prop+adj+v	.858	**.776**	**.815**	.786	**.724**	**.749**
$\mathtt{l}_{1200}$	n+prop+adj	**.873**	.769	**.818**	**.794**	.715	**.742**
	n+adj+v	.864	.765	**.812**	.787	.711	.740
	n+prop+v	.866	.769	**.815**	.787	.715	**.742**
	n+prop+adj+v	.867	**.775**	**.818**	**.794**	**.723**	**.750**
$\mathtt{l}_{1400}$	n+prop+adj	**.875**	.768	**.818**	**.793**	.713	.740
	n+adj+v	**.873**	.767	**.817**	**.800**	.713	**.746**
	n+prop+v	**.872**	.767	**.816**	.795	.713	**.742**
	n+prop+adj+v	**.874**	**.774**	**.821**	**.801**	**.721**	**.751**
$\mathtt{l}_{1500}$	n+prop+adj+v	**.874**	**.772**	**.820**	**.797**	.718	**.746**
$\mathtt{l}_{1600}$	n+prop+adj+v	**.879**	.766	**.819**	**.803**	.711	**.742**
$\mathtt{l}_{1700}$	n+prop+adj+v	**.878**	.767	**.819**	**.803**	.712	**.743**

Tabela 7.13: Corpus PGR: valores das medidas de desempenho das melhores experiências com conjuntos de classes gramaticais para diferentes valores de limiar.

	π^{μ}	ρ^{μ}	f_1^{μ}	π^{M}	ρ^{M}	f_1^{M}
lema-imut-tfc-$\mathtt{l}_{40}$	**.975**	**.975**	**.975**	**.976**	**.975**	**.975**
adj	.718	.718	.718	.738	.718	.720
n	.951	.951	.951	.951	.951	.951
prop	.891	.891	.891	.900	.891	.893
v	.698	.698	.698	.702	.698	.699
n+adj	.950	.950	.950	.950	.950	.950
n+prop	**.966**	**.966**	**.966**	**.966**	**.966**	**.966**
n+v	.950	.950	.950	.950	.950	.950
prop+adj	.923	.923	.923	.924	.923	.923
adj+v	.809	.809	.809	.811	.809	.810
prop+v	.892	.892	.892	.894	.892	.893
n+prop+adj	**.967**	**.967**	**.967**	**.967**	**.967**	**.967**
n+adj+v	.946	.946	.946	.946	.946	.946
n+prop+v	.961	.961	.961	.961	.961	.961
prop+adj+v	.919	.919	.919	.920	.919	.919
n+prop+adj+v	**.966**	**.966**	**.966**	**.966**	**.966**	**.966**

Tabela 7.14: Corpus Folha: valores das medidas de desempenho para as experiências com conjuntos de classes gramaticais.

resultados das melhores experiências com informação morfo-sintáctica. Como representante das experiências morfo-sintácticas escolheu-se a experiência `n+prop+adj+v-l`$_{50}$ por ser aquela com maiores valores dentro das experiências equivalentes.

		π^{μ}	ρ^{μ}	f_1^{μ}	π^{M}	ρ^{M}	f_1^{M}
l_{40}	n+prop	**.966**	**.966**	**.966**	**.966**	**.966**	**.966**
	n+prop+adj	**.967**	**.967**	**.967**	**.967**	**.967**	**.967**
	n+prop+adj+v	**.966**	**.966**	**.966**	**.966**	**.966**	**.966**
l_{50}	n+prop	**.958**	**.958**	**.958**	**.958**	**.958**	**.958**
	n+prop+adj	**.961**	**.961**	**.961**	**.962**	**.961**	**.961**
	n+prop+adj+v	**.967**	**.967**	**.967**	**.968**	**.967**	**.967**
l_{60}	n+prop	**.959**	**.959**	**.959**	**.959**	**.959**	**.959**
	n+prop+adj	**.958**	**.958**	**.958**	**.958**	**.958**	**.958**
	n+prop+adj+v	.949	.949	.949	.949	.949	.949
l_{70}	n+prop	.939	.939	.939	.940	.939	.939
	n+prop+adj	.950	.950	.950	.950	.950	.950
	n+prop+adj+v	.946	.946	.946	.946	.946	.946

Tabela 7.15: Corpus `Folha`: valores das medidas de desempenho das melhores experiências com conjuntos de classes gramaticais para diferentes valores de limiar.

Corpus Publico. Os resultados obtidos para cada conjunto de classes gramaticais, utilizando como base `lema-freq-tfc-l`$_1$ encontram-se discriminados na Tabela 7.16. Para este corpus é possível afirmar que apenas a combinação das quatro classes gramaticais (`n+prop+adj+v`) consegue valores equivalentes à experiência base.

Mais uma vez, realizou-se um conjunto de experiências com valores de limiar mais altos. Experimentou-se l_{50} e l_{100}, mas nestas experiências obtiveram-se valores inferiores aos da melhor experiência morfo-sintáctica. Assim, escolheu-se como representante a experiência `n+prop+adj+v-l`$_1$.

7.3.3 Informação sintáctica

Como referido na secção 7.1.3 a informação sintáctica da frase foi utilizada, através da correspondente árvore sintáctica, em três representações distintas do documento: árvore-sintáctica (`arv`), sequência-de-palavras (`seq`) e saco-de-palavras (`saco`).

Em todas as representações utilizou-se um subconjunto da árvore sintáctica completa – os nós cujas palavras pertenciam às classes gramaticais nome (`n`), nome próprio (`prop`), adjectivo (`adj`), verbo (`v`), pronome (`pron`) e advérbio (`adv`). As palavras incluidas na representação foram substituidas pela sua raíz.

Consideraram-se duas experiências distintas. A primeira inclui todas as árvores sin-

	π^{μ}	ρ^{μ}	f_1^{μ}	π^{M}	ρ^{M}	f_1^{M}
lema-freq-tfc-l$_1$	**.843**	**.843**	**.843**	**.842**	**.831**	**.836**
adj	.683	.683	.683	.671	.663	.666
n	.808	.808	.808	.806	.704	.706
prop	.784	.784	.784	.778	.766	.771
v	.635	.635	.635	.623	.614	.618
n+adj	.807	.807	.807	.803	.792	.796
n+prop	.829	.829	.829	.824	.814	.818
n+v	.803	.803	.803	.821	.789	.802
prop+adj	.804	.804	.804	.800	.788	.793
adj+v	.732	.732	.732	.726	.715	.720
prop+v	.805	.805	.805	.802	.789	.795
n+prop+adj	.833	.833	.833	.830	.819	.824
n+adj+v	.813	.813	.813	.809	.800	.804
n+prop+v	.835	.835	.835	.833	.822	.827
prop+adj+v	.816	.816	.816	.811	.801	.805
n+prop+adj+v	**.839**	**.839**	**.839**	**.838**	**.827**	**.831**

Tabela 7.16: Corpus Publico: valores das medidas de desempenho para as experiências com conjuntos de classes gramaticais.

tácticas do documento (tot) e a outra contém as árvores sintácticas das orações que se consideraram mais informativas (fcl) – as orações finitas (fcl) que contêm o sujeito (SUBJ), predicado (P) e complemento directo (ACC). Experimentou-se ainda incluir apenas as n primeiras orações de cada documento, com $n \in \{1, 3, 5, 10\}$ para a experiência fcl.

No total, realizaram-se 6 experiências para cada representação. Para cada corpus, escolheu-se para representante do nível de informação, a representação árvore-sintáctica (arv) com melhores desempenhos. As subsecções seguintes apresentam esses resultados.

Corpus Folha2. A Tabela 7.17 apresenta os valores das medidas de desempenho das experiências com informação sintáctica para o corpus Folha2. É possível observar que, embora não se tenham obtido melhores desempenhos com a representação árvore-sintáctica relativamente às outras representações, obtiveram-se valores equivalentes ao utilizar todas as árvores sintácticas dos documentos. Escolhemos então essa experiência como representante do corpus Folha2 com informação sintáctica.

Corpus Publico9510. A Tabela 7.18 apresenta os valores das medidas de desempenho das experiências com informação sintáctica para o corpus Publico9510. A experiência saco-tot (com todas as árvores sintácticas) apresenta os melhores resultados com diferenças significativas. Relativamente às restantes, os valores a negrito correspondem a valores sem diferenças significativas quando comparados com o segundo

		π^{μ}	ρ^{μ}	f_1^{μ}	π^{M}	ρ^{M}	f_1^{M}
arv	tot	**.948**	**.948**	**.948**	**.949**	**.948**	**.947**
	fcl	.912	.912	.912	.912	.912	.911
	$\texttt{fcl}_1$	.787	.787	.787	.789	.788	.786
	$\texttt{fcl}_3$	.871	.871	.871	.873	.872	.871
	$\texttt{fcl}_5$	.900	.900	.900	.902	.900	.900
	$\texttt{fcl}_{10}$	.916	.916	.916	.916	.916	.916
seq	tot	**.964**	**.964**	**.964**	**.967**	**.964**	**.964**
	fcl	**.956**	**.956**	**.956**	**.957**	**.956**	**.956**
	$\texttt{fcl}_1$	.811	.811	.811	.811	.812	.811
	$\texttt{fcl}_3$	.908	.908	.908	.909	.908	.908
	$\texttt{fcl}_5$	**.952**	**.952**	**.952**	**.953**	**.952**	**.952**
	$\texttt{fcl}_{10}$	**.952**	**.952**	**.952**	**.955**	**.952**	**.951**
saco	tot	**.964**	**.964**	**.964**	**.965**	**.964**	**.964**
	fcl	**.956**	**.956**	**.956**	**.958**	**.956**	**.956**
	$\texttt{fcl}_1$	.896	.896	.896	.896	.896	.896
	$\texttt{fcl}_3$	**.936**	**.936**	**.936**	**.938**	**.936**	**.936**
	$\texttt{fcl}_5$	**.944**	**.944**	**.944**	**.945**	**.944**	**.944**
	$\texttt{fcl}_{10}$	**.948**	**.948**	**.948**	**.950**	**.948**	**.948**

Tabela 7.17: Corpus `Folha2`: Valores das medidas de desempenho para diversas representações utilizando informação sintáctica.

melhor valor obtido. Neste corpus voltou-se a escolher `arv-tot` como representante, embora esta experiência não possua valores equivalentes ao segundo melhor para as medidas ρ^M e f_1^M.

7.3.4 Informação semântica

Como referido em 7.1.4 a informação semântica da frase foi utilizada, através das correspondentes estruturas de representação do discurso em duas representações distintas do documento: estrutura-do-discurso (`disc`) e saco-de-palavras (`saco`).

Na representação estrutura-do-discurso consideraram-se duas experiências distintas consoante as substituições realizadas sobre os referentes (ver secção 6.2.2):

- `corpus`: aplicou-se a substituição de nomes próprios;
- `corpus+frase`: aplicou-se a substituição de nomes próprios e condições atómicas.

Para cada uma, experimentou-se, também, utilizar apenas as n (com $n \in \{1, 3, 5, 10\}$) primeiras estruturas de representação do discurso de cada documento.

Para construir a representação saco-de-palavras utilizaram-se as folhas (palavras) de ambas as transformações. Para o corpus `Folha2` obtiveram-se 13508 palavras distintas

		π^{μ}	ρ^{μ}	f_1^{μ}	π^{M}	ρ^{M}	f_1^{M}
arv	tot	**.812**	**.812**	**.812**	**.810**	**.788**	**.792**
	fcl	.789	.789	.789	.789	.765	.770
	$\texttt{fcl}_1$	.675	.674	.674	.544	.526	.530
	$\texttt{fcl}_3$	.699	.699	.699	.683	.667	.670
	$\texttt{fcl}_5$	.734	.734	.734	.720	.702	.706
	$\texttt{fcl}_{10}$	.782	.782	.782	.776	.757	.760
seq	tot	**.829**	**.829**	**.829**	**.821**	**.811**	**.814**
	fcl	**.819**	**.819**	**.819**	**.814**	**.801**	**.804**
	$\texttt{fcl}_1$	.672	.671	.671	.549	.534	.539
	$\texttt{fcl}_3$	.708	.708	.708	.698	.681	.686
	$\texttt{fcl}_5$	.761	.761	.761	.750	.738	.741
	$\texttt{fcl}_{10}$	.791	.791	.791	.779	.771	.772
saco	tot	*.857*	*.857*	*.857*	*.858*	*.842*	*.847*
	fcl	**.824**	**.824**	**.824**	**.823**	**.809**	**.811**
	fcl	.824	.824	.824	.823	.809	.811
	$\texttt{fcl}_1$	.613	.613	.613	.605	.588	.594
	$\texttt{fcl}_3$	.734	.734	.734	.725	.707	.711
	$\texttt{fcl}_5$	.790	.790	.790	.782	.765	.768
	$\texttt{fcl}_{10}$	**.815**	**.815**	**.815**	**.803**	**.793**	**.793**

Tabela 7.18: Corpus `Publico9510`: Valores das medidas de desempenho para diversas representações utilizando informação sintáctica.

e para o corpus `Publico9510`, 46186 palavras distintas.

No total, realizaram-se 12 experiências. Para cada corpus, escolheu-se como representante, a representação estrutura-do-discurso (`disc`) com melhores desempenhos. As subsecções seguintes apresentam esses resultados.

Corpus Folha2. A Tabela 7.19 apresenta os valores das medidas de desempenho das experiências com informação semântica para o corpus `Folha2`.Escolheu-se como representante, a experiência `corpus+frase-disc`.

Corpus Publico9510. A Tabela 7.20 apresenta os valores das medidas de desempenho das experiências com informação semântica para o corpus `Publico9510`. A experiência `corpus+frase-disc` foi, de igual forma, a escolhida como representante deste conjunto de experiências.

		π^{μ}	ρ^{μ}	f_1^{μ}	π^{M}	ρ^{M}	f_1^{M}
corpus	disc	.787	.787	.787	.842	.787	.796
	disc_1	.582	.582	.582	.677	.583	.586
	disc_3	.691	.691	.691	.757	.691	.702
	disc_5	.727	.727	.727	.777	.727	.734
	disc_{10}	.755	.755	.755	.817	.755	.766
	saco	**.964**	**.964**	**.964**	**.965**	**.964**	**.964**
corpus+ frase	disc	**.952**	**.952**	**.952**	**.956**	**.952**	**.952**
	disc_1	.795	.795	.795	.798	.795	.794
	disc_3	.916	.916	.916	.921	.916	.916
	disc_5	.920	.920	.920	.926	.920	.920
	disc_{10}	**.948**	**.948**	**.948**	**.951**	**.948**	**.948**
	saco	**.952**	**.952**	**.952**	**.954**	**.952**	**.952**

Tabela 7.19: Corpus Folha2: Valores das medidas de desempenho para diversas representações utilizando informação semântica.

		π^{μ}	ρ^{μ}	f_1^{μ}	π^{M}	ρ^{M}	f_1^{M}
corpus	disc	.655	.655	.655	.732	.599	.623
	disc_1	.364	.364	.364	.533	.278	.288
	disc_3	.484	.484	.484	.604	.418	.451
	disc_5	.545	.545	.545	.660	.481	.510
	disc_{10}	.593	.593	.593	.692	.538	.567
	saco	**.821**	**.821**	**.821**	**.816**	**.808**	**.810**
corpus+ frase	disc	**.833**	**.833**	**.833**	**.831**	**.817**	**.820**
	disc_1	.471	.471	.471	.484	.437	.445
	disc_3	.679	.679	.679	.671	.645	.651
	disc_5	.740	.740	.740	.735	.710	.717
	disc_{10}	.787	.787	.787	.780	.772	.773
	saco	**.814**	**.814**	**.814**	**.822**	.788	.788

Tabela 7.20: Corpus Publico9510: Valores das medidas de desempenho para diversas representações utilizando informação semântica.

7.4 Avaliação comparativa

Esta secção faz uma avaliação comparativa sobre utilização de informação linguística na classificação de textos na língua Portuguesa. Discutem-se as diferenças obtidas:

- entre representações que utilizam o mesmo nível de informação linguística,
- entre diferentes níveis de informação linguística, e ainda
- entre diferentes classes da mesmo corpus.

7.4.1 Diferenças entre representações do mesmo nível

As diferenças entre representações com o mesmo nível de informação linguística podem ser avaliadas através de algumas Tabelas apresentadas ao longo da secção anterior.

Informação morfológica

Para os corpora `PGR` e `Folha` a redução do número de atributos foi benéfica: atingiram-se melhores desempenhos quando aplicada uma redução de termos com agressividades de 45.5 ($\mathtt{l}_{800}$) e 36.9 ($\mathtt{l}_{40}$), respectivamente. Para o corpus `Publico` foi possível reduzir o número de atributos com uma agressividade de 7.9 ($\mathtt{l}_{50}$) sem deteriorar os resultados.

Quanto à normalização existem, para os três corpora, experiências que utilizam a raíz e a palavra original no grupo das melhores (ver Tabelas 7.7, 7.8 e 7.9). No entanto, se considerarmos os valores de limiar mais altos, aparecem diferenças: no corpus `Folha` deve ser utilizada a raíz, no `Publico` as palavras originais e no `PGR` é indiferente.

Se considerarmos as funções de graduação e de ponderação, a frequência das palavras combinada com a mesma medida normalizada constitui a melhor opção para o corpus `PGR` enquanto que a informação mútua combinada com a medida `tfidf` é-o para o corpus `Folha`. Para o corpus `Publico` estas combinações já não são tão claras: enquanto a informação mútua deve ser utilizada com a medida `tfidf`, a utilização da frequência das palavras torna a escolha da função de ponderação indiferente. Esta diferença poderá estar relacionada com tamanho do corpus em estudo: para corpus com mais documentos, as estatísticas são mais fiáveis tornando os resultados menos variáveis a diferentes representações.

Sobre todo este conjunto de experiências é possível concluir que a melhor representação com informação morfológica depende do corpus em estudo e, como tal, deve ser realizada uma análise inicial para encontrar qual a combinação de normalização, função de graduação, valor de limiar e função de ponderação que mais se adequa.

Informação morfo-sintáctica

Os conjuntos de classes gramaticais mais informativos para a classificação também parecem variar com o corpus (Tabelas 7.12, 7.14 e 7.16):

- para o corpus `PGR` são os nomes com mais duas ou três classes gramaticais (`n+adj+v`, `n+prop+adj`, `n+prop+v` e `n+prop+adj+v`);
- para a `Folha` são os nomes e nomes próprios com ou sem adjectivos ou a combinação das quatro classes gramaticais (`n+prop`, `n+prop+adj` e `n+prop+adj+v`);
- para o `Publico` apenas o conjunto das quatro classes gramaticais (`n+prop+adj+v`) obteve valores equivalentes à representação morfológica.

Desta forma é possível concluir que os nomes são uma classe gramatical fundamental para a classificação, mas sozinhos não possuem poder discriminativo suficiente sobre as classes [Gonçalves and Quaresma 2005b; Gonçalves et al. 2006]. Mais, os nomes próprios são igualmente importantes pelo menos para os corpora jornalísticos.

Quando aplicados maiores valores de limiar, apenas o conjunto das quatro classes gramaticais `n+prop+adj+v` originou desempenhos equivalentes para todos os corpora. Mais ainda, a utilização destas classes gramaticais juntamente com valores de limiar mais altos que os utilizados na informação morfológica permitiu, até, obter uma melhoria da micro-precisão para o corpus `PGR` (ver Tabelas 7.12 e 7.13) sem deteriorar as restantes medidas.

Informação sintáctica

As três representações estudadas com informação sintáctica (`arv`, `seq` e `saco`) comportaram-se de forma diferente consoante o corpus (ver Tabelas 7.17 e 7.18): enquanto se obtiveram resultados equivalentes com os documentos completos para as três representações no corpus `Folha2`, a representação saco-de-palavras (`saco`) obteve resultados superiores às restantes no corpus `Publico9510`. Este resultado pode dever-se, uma vez mais, ao tamanho do corpus: sendo o `Publico9510` maior, as estatísticas das palavras tornam-se mais fiáveis, suplantando outro tipo de informação.

Por outro lado, a utilização da representação árvore-sintáctica apenas obteve bons resultados utilizando toda a informação dos documentos (experiência `tot`), enquanto a representação sequência-de-palavras obteve resultados equivalentes utilizando as orações finitas com sujeito, predicado e complemento directo (experiência `fcl`). Mais ainda, para o corpus `Folha2`, obtiveram-se resultados equivalentes utilizando apenas as 5 e 10 primeiras orações desse tipo.

Das três, a representação saco-de-palavras pode ser considerada a mais estável. Com ela obtiveram-se resultados equivalentes ao melhor com as três primeiras orações finitas para o corpus `Folha2` e as dez primeiras orações finitas para o corpus `Publico9510` (micro-precisão, cobertura e micro-f_1).

Sobre este conjunto de experiências é possível concluir que a informação sintáctica pode ser utilizada na classificação de documentos, mas a representação árvore-sintáctica não é a mais adequada. A representação sequência-de-palavras embora ordenada, é mais simples conseguindo discriminar as classes com menos informação. Na mesma linha, se utilizarmos a informação sintáctica numa representação saco-de-palavras são necessárias menos frases para poder discriminar de forma equivalente as classes.

Informação semântica

Para ambos os corpora obtiveram-se resultados equivalentes nas representações saco-de-palavras (`saco`) e estrutura-do-discurso (`disc`) com a substituição `corpus+frase` (ver Tabelas 7.19 e 7.20). Mais ainda, no corpus `Publico9510`, obtiveram-se valores de macro-cobertura e macro-f_1 superiores na representação estruturada, e no corpus `Folha2` obtiveram-se resultados equivalentes utilizando a informação das 10 primeiras frases ($\texttt{disc}_{10}$).

Por outro lado, a substituição apenas dos nomes próprios (nível `corpus`) não pareceu produzir bons resultados; apenas na representação saco-de-palavras para o corpus `Folha2` se obtiveram desempenhos equivalentes aos melhores.

Sobre este conjunto de experiências é possível concluir que a representação estrutura-do-discurso com a substituição `corpus+frase` constitui uma representação válida para problemas de classificação de texto.

7.4.2 Diferenças entre níveis de informação linguística

A Tabela 7.21 resume os valores das medidas de desempenho das experiências representantes de cada nível de informação linguística[11].

Com informação morfo-sintáctica apenas se conseguiram resultados equivalentes aos da informação morfológica nos corpora `Folha` e `Publico`; no corpus `Folha` foi possível reduzir o número de atributos (de 529 para 397 atributos) mas no `Publico` não. No entanto, para o corpus `PGR`, foi possível melhorar a precisão do classificador sem deteriorar a sua cobertura; além disso, este valor foi obtido com um valor de limiar mais alto que permitiu reduzir o número de atributos de 1514 para 959.

[11] Os valores a negrito correspondem a valores sem diferenças significativas para cada corpus.

			π^{μ}	ρ^{μ}	f_1^{μ}	π^{M}	ρ^{M}	f_1^{M}
PGR	Morfológica	lema-freq-txc-l$_{800}$	.846	**.782**	**.813**	.782	**.732**	**.753**
	Morfo-sint.	n+prop+adj+v-l$_{1400}$	**.874**	**.774**	**.821**	**.801**	**.721**	**.751**
Folha	Morfológica	lema-imut-tfc-l$_{40}$	**.975**	**.975**	**.975**	**.976**	**.975**	**.975**
	Morfo-sint.	n+prop+adj+v-l$_{50}$	**.967**	**.967**	**.967**	**.968**	**.967**	**.967**
Publico	Morfológica	lema-freq-tfc-l$_{1}$	**.843**	**.843**	**.843**	**.842**	**.831**	**.836**
	Morfo-sint.	n+prop+adj+v-l$_{1}$	**.839**	**.839**	**.839**	**.838**	**.827**	**.831**
Folha2	Morfológica	lema-imut-tfc-l$_{40}$	**.976**	**.976**	**.976**	**.977**	**.976**	**.976**
	Sintáctica	arv-tot	.948	.948	.948	.949	.948	.947
	Semântica	corpus+frase-disc	**.952**	**.952**	**.952**	**.956**	**.952**	**.952**
Publico9510	Morfológica	lema-freq-tfc-l$_{1}$	**.855**	**.855**	**.855**	**.854**	**.840**	**.844**
	Sintáctica	arv-tot	.812	.812	.812	.810	.788	.792
	Semântica	corpus+frase-disc	**.833**	**.833**	**.833**	**.831**	**.817**	**.820**

Tabela 7.21: Valores das medidas de desempenho para as experiências escolhidas em cada nível de informação linguística.

Esta diferença pode dever-se à especificidade da linguagem dos documentos jurídicos e/ou ao tipo de classificação destes documentos (classificação multi-etiqueta numa hierarquia de classes). Além disso, a etiquetagem dos documentos foi realizada ao longo de vários anos (existem documentos datados de 1940 até 2003) por várias pessoas o que provoca, com certeza, algum ruído no processo; a informação morfo-sintáctica parece minorar este problema.

Desta forma é possível concluir que a informação morfo-sintáctica das palavras constitui uma forma válida de selecção de atributos, nomeadamente eliminando não apenas as *stopwords*, mas todas as palavras que não são nomes, nomes próprios, adjectivos ou verbos.

Por outro lado, para os corpora `Folha2` e `Publico9510`, os níveis morfológico e semântico apresentaram melhores resultados que a representação do nível sintáctico. Pode afirmar-se, então, que a informação semântica possui um poder discriminativo sobre as classes equivalente à informação morfológica, o que já não acontece com a informação sintáctica. Mais ainda, a utilização da informação semântica realiza uma forma válida de selecção de atributos: enquanto os corpora originais `Folha2` e `Publico9510` possuem 19417 e 70743 palavras distintas (página 111), a sua representação com informação semântica possui, respectivamente, 13508 e 46186 palavras distintas (página 123).

7.4.3 Diferenças entre classes do mesmo corpus

As subsecções anteriores analisam as micro- e macro-médias entre diferentes representações que utilizam o mesmo nível de informação linguística e as melhores representações escolhidas para cada nível. Nesta subsecção analisam-se os valores individuais da precisão, cobertura e f_1 para cada um dos corpora.

Os valores da Tabela 7.22 correspondem à informação morfo-sintáctica para corpora `PGR`, `Folha` e `Publico` e os da Tabela 7.23 à representação `corpus+frase-disc` da informação semântica para corpora `Folha2` e `Publico9510`. Estes valores dizem respeito às experiências referidas na Tabela 7.21.

Escolheram-se estas representações porque:

- a primeira permitiu obter resultados equivalentes aos da informação morfológica com valores de limiar iguais ou superiores e
- a segunda representa o documento através do seu significado e não através das palavras utilizadas para transmitir esse significado.

	PGR			Folha			Publico		
	π	ρ	f_1	π	ρ	f_1	π	ρ	f_1
c_1	.635	.591	.612	.988	.971	.979	.838	.807	.822
c_2	.388	.231	.290	.929	.924	.927	.826	.884	.854
c_3	.986	.969	.978	.965	.959	.962	.964	.966	.965
c_4	.987	.982	.985	.971	.971	.971	.672	.593	.630
c_5	.980	.834	.901	.943	.971	.957	.930	.933	.932
c_6	.449	.299	.359				.826	.856	.841
c_7	.843	.880	.861				.908	.910	.909
c_8	.588	.567	.577				.839	.772	.804
c_9	.971	.970	.971				.691	.700	.695
c_{10}	.998	.995	.996						

Tabela 7.22: Corpora `PGR`, `Folha` e `Publico`: valores das medidas de desempenho individuais para o nível de informação morfo-sintáctico.

	Folha2			Publico9510		
	π	ρ	f_1	π	ρ	f_1
c_1	1.000	.940	.969	.835	.845	.840
c_2	1.000	.860	.925	.810	.871	.839
c_3	.925	.980	.951	.933	.955	.944
c_4	.980	1.000	.990	.695	.559	.620
c_5	.875	.980	.925	.914	.955	.934
c_6				.815	.872	.842
c_7				.914	.891	.902
c_8				.781	.864	.820
c_9				.759	.538	.630

Tabela 7.23: Corpora `Folha2` e `Publico9510`: valores das medidas de desempenho individuais para o nível de informação semântica.

O corpus `Folha` apresenta valores muito uniformes entre as várias classes e todos eles se situam acima de 0.92. Os valores mais baixos para as três medidas de desempenho dizem respeito à classe `imoveis`.

O corpus Folha2 também apresenta valores altos para todas as classes (com informação semântica): até se atingiram valores de precisão máxima ($\pi = 1$) para as classes desporto e imoveis e de cobertura máxima ($\rho = 1$) para a classe politica. No entanto, estes valores foram conseguidos à custa de valores mais baixos da outra medida de desempenho.

Por outro lado, no corpus Publico os resultados já não são tão uniformes: os classificadores para as classes desporto, economia e mundo obtiveram os melhores desempenhos com valores de precisão, cobertura e f_1 acima de 0.90, enquanto os classificadores para as classes diversos e sociedade obtiveram valores entre 0.59 e 0.70. Para as restantes classes (ciencias-tecnologia-educacao, cultura, local e nacional) obtiveram-se classificadores com f_1 entre 0.80 e 0.85.

Se pensarmos nos conceitos que cada uma destas classes reflecte, de facto desporto ou economia são conceitos mais objectivos e bem definidos que diversos ou sociedade; estas diferenças traduzem-se em desempenhos superiores dos classificadores para as primeiras classes. O corpus Publico9510 demonstra, como seria de esperar, o mesmo comportamento.

Quanto ao corpus PGR, as diferenças de desempenho dos classificadores para cada classe são ainda maiores: as classes competencia e funcionario publico têm valores de precisão inferiores a 0.45 e de cobertura inferiores a 0.30; as classes aposentacao e militar têm valores entre 0.56 e 0.64 e as restantes classes têm valores de f_1 superiores a 0.90. Mais uma vez, os valores de desempenho reflectem o tipo de conceito que a classe representa: a classe competencia representa um conceito abstracto e tem um valor de $f_1 = .290$ enquanto, prisioneiro de guerra constitui um conceito muito mais objectivo e tem um valor de $f_1 = .996$.

Dos três corpora originais, o Folha é, sem dúvida, aquele possui os conceitos mais concretos e bem definidos. Talvez seja essa a razão pela qual os seus classificadores possuam, na generalidade, desempenhos superiores. No corpus Publico, embora as classes representem conceitos concretos, estes não são tão estanques: existem documentos que podem ser classificados tanto na classe sociedade como na classe nacional, por exemplo, o que torna o processo de classificação automática mais difícil; isto sem falar na classe diversos que não possui um conceito associado. Por outro lado, os conceitos do corpus PGR, possuem graus de generalidade e de abstracção não uniformes, talvez daí a grande dificuldade de alguns classificadores.

7.5 Trabalho relacionado

Os trabalhos referidos na secção 4.1 utilizam corpora escritos na língua Inglesa. A sua maioria utiliza o corpus Reuters e alguns as categorias do Yahoo. Embora nalguns sejam exploradas estruturas sintácticas das frases ou o significado das palavras, originando termos de indexação mais complexos que a representação saco-de-palavas, o documento é representado através de estatísticas feitas sobre esses termos.

Por exemplo, o estudo de Basili et al. [2004] utiliza como termos de indexação dos documentos, além das palavras originais, alguma informação sintáctica (através das etiquetas morfo-sintácticas e da detecção de nominais complexos e nomes próprios) e também semântica (através do significado de palavras), representando o documento através de um saco desses termos. Os resultados obtidos por estas representações permitiram concluir que quer os sintagmas quer o significado das palavras podem ser substituidos pelas palavras simples já que as categorias diferem nas palavras utilizadas e não nos seus significados [Basili et al. 2004].

Estas conclusões, embora obtidas para a língua Inglesa, estão de acordo com os resultados conseguidos neste trabalho. No entanto, este trabalho vai mais longe ao representar o documento de forma estruturada e ao utilizar, não o significado das palavras, mas o significado (parcial) das frases.

Capítulo 8

Conclusões e Trabalho Futuro

Ao longo desta dissertação abordaram-se questões relacionadas com a informação linguística e com a classificação automática de textos, apresentaram-se as técnicas de classificação de texto mais utilizadas e, em particular os métodos de núcleo, propôs-se uma nova representação dos documentos que utiliza informação linguística e utilizaram-se núcleos estruturados para encontrar as semelhanças entre documentos e, finalmente, avaliaram-se os resultados obtidos com a utilização dos vários níveis de informação linguística.

Com este trabalho realizou-se:

- uma análise exaustiva das diversas técnicas de extracção de informação morfológica dos documentos, técnicas usualmente utilizadas em Recuperação de Informação. Para a análise utilizaram-se corpora escritas na língua Portuguesa;

- um exame de vários algoritmos para classificação de textos. Adoptou-se um, a máquina de vectores de suporte, que permite a utilização de representações estruturadas dos documentos;

- a proposta de uma nova representação do documento. Esta representação contém a forma lógica do documento, encerrando a sua informação semântica que, através um núcleo estruturado, é utilizada pela máquina de vectores de suporte;

- uma avaliação sobre a contribuição das estruturas linguísticas no processo de classificação automática de documentos para a língua Portuguesa.

A próxima secção apresenta as suas principais conclusões, enquanto a seguinte aponta caminhos para trabalho futuro.

8.1 Conclusões

Neste trabalho, orientado para a língua Portuguesa, estudou-se a influência do processamento da língua natural na tarefa da classificação automática de textos. Por um lado, utilizaram-se ferramentas de processamento de informação linguística para obter a representação sintáctica e semântica dos documentos; por outro, escolheu-se a máquina de vectores de suporte como algoritmo de aprendizagem supervisionada.

Joachims [2002] formalizou a sua utilização no âmbito da classificação de textos e este algoritmo tem sido considerado como aquele que produz classificadores com os melhores desempenhos. Além disso, quando associado ao núcleo apropriado permite utilizar representações do documento quer no espaço de vectores (a representação tradicional) quer em espaços estruturados. A sua aplicação a textos escritos na língua Portuguesa confirma esta realidade.

A arquitectura modular desenvolvida permite seleccionar o nível de informação linguística dos documentos a utilizar na tarefa de classificação: informação morfológica, sintáctica ou semântica.

Utilizando a informação morfológica sobre três corpora (`PGR`, `Folha` e `Publico`) com diferentes características, examinaram-se as seguintes opções (usuais na área de recuperação de informação) na obtenção de uma representação estatística dos documentos:

- duas formas de normalização das palavras: aplicar ou não a lematização;
- duas funções de graduação das palavras: frequência (`freq`) e a informação mútua (`imut`);
- vários valores de limiar para selecção dos termos;
- duas funções de ponderação: frequência do termo (`txc`) e frequência do termo com frequência inversa do documento (`tfc`);

A análise dos resultados (ver Tabelas 7.7, 7.8 e 7.9) mostra que as diferentes opções podem aperfeiçoar o desempenho se forem combinadas de maneira apropriada. A melhor combinação parece depender do corpus. No entanto, a redução do número de termos (até certo ponto) mostra ser benéfica, e a lematização (pelo menos quando não é aplicado nenhum valor de limiar) também.

Por outro lado, as funções de graduação e de ponderação parecem depender da natureza do corpus: no que retrata uma área de conhecimento específica é a combinação frequência das palavras com essa medida normalizada (`freq-txc`) e nos de âmbito geral os melhores desempenhos surgem com a combinação informação mútua e a frequência do documento inversa normalizada (`imut-tfc`). No entanto, no corpus `Publico` a

função de graduação frequência dos documentos produziu resultados igualmente bons (devido à sua dimensão, é possível que a existência de estatísticas mais fiáveis torne os resultados menos variáveis a diferentes representações).

Para a língua Portuguesa, a utilização da informação morfo-sintáctica das palavras como função de selecção de termos produz um desempenho equivalente às técnicas usuais de recuperação de informação (ver Tabela 7.21). Mais ainda, ao seleccionar as palavras pertencentes às classes gramaticais nome, nome próprio, adjectivo e verbo é possível reduzir o número de termos sem deteriorar o desempenho (para o corpus jurídico obtiveram-se até melhores precisões).

Para os níveis de informação sintáctica e semântica utilizaram-se subconjuntos de dois dos corpora anteriores, `Folha2` e `Publico`.

Com as árvores sintácticas das frases construíram-se três representações distintas. Na primeira, representou-se a sequência das árvores sintácticas (parciais) das frases do documento (`arv`); na segunda, planificou-se a árvore representando, desta forma, a sequência das palavras do documento (`seq`); na terceira retirou-se a informação relativa à ordem das palavras obtendo uma representação saco-de-palavras (`saco`). Enquanto a última utiliza um núcleo do espaço de vectores as outras duas utilizam um núcleo para dados estruturados.

A análise dos resultados mostra que, embora as três representações utilizem as mesmas palavras do documento, a representação não estruturada é superior, ou porque o seu desempenho é maior ou porque um desempenho equivalente é alcançado com as primeiras n árvores sintácticas do documento (ver Tabelas 7.17 e 7.18).

Para construir uma representação que, por um lado contivesse a informação semântica e, por outro, fosse utilizável pelo algoritmo de aprendizagem, propôs-se uma nova representação estruturada do documento, a representação estrutura-do-discurso (`disc`). Esta representação transforma as entidades de representação do discurso numa estrutura em árvore que considera restrições sobre as entidades referidas no documento (referentes). Examinaram-se duas substituições distintas sobre os referentes: apenas de nomes próprios – substituição a nível do `corpus`, e dos nomes próprios e condições atómicas – substituição a nível do `corpus+frase`. Para cada substituição examinou-se, também, a representação saco-de-palavras (`saco`) resultante da eliminação da estrutura da representação.

A análise dos resultados indica que a informação semântica, apoiada na representação estruturada do documento com a substituição adequada (`corpus+frase`) possui o mesmo poder discriminativo sobre as classes do que a representação não estruturada (ver Tabelas 7.19 e 7.20). Para o corpus `Publico2`, obtiveram-se, até, valores superiores em algumas medidas de desempenho.

Ao examinar os resultados das melhores experiências para os níveis morfológico, sintáctico e semântico (ver Tabela 7.21) conclui-se que, para a língua Portuguesa, o primeiro e o último são equivalentes e são, por sua vez, superiores ao nível sintáctico. A inferioridade da representação sintáctica está de acordo com as expectativas iniciais – a estrutura sintáctica da frase não é reveladora da categoria semântica do documento; será, porventura, reveladora do autor do mesmo ou da especificidade da linguagem utilizada (documentos de carácter geral como os jornalísticos ou de áreas de conhecimento muito específicos como os jurídicos).

Por outro lado, a equivalência de resultados obtidos com as representações morfológica e semântica demonstra que ambas as representações do documento, uma baseada em estatísticas sobre as suas palavras e outra baseada no seu significado, são válidas.

Se considerarmos o número de palavras distintas que as representações morfológica e semântica contêm conclui-se que a utilização da forma lógica dos documentos realiza uma forma válida de selecção de atributos: conseguiram-se reduções na ordem dos 30%.

Concluindo, tal como Lewis and Jones [1997] afirmaram para a tarefa de Recuperação de Informação, a utilização de informação linguística "sofisticada" (semântica) não trouxe ganhos significativos versus a utilização de abordagens estatísticas para a selecção e redução de atributos. No entanto, ambos podem ser interpretados como métodos distintos de redução da dimensão na representação dos documentos.

Os resultados obtidos pela utilização da informação semântica são, ainda assim, reveladores da aptidão da representação proposta para conter a forma lógica do documento e, embora não se tenham conseguido obter desempenhos superiores, os resultados obtidos são animadores já que a informação semântica utilizada apenas descreve parcialmente o significado do documento. Acredita-se que, ao aperfeiçoar a representação da forma lógica com as possibilidades descritas na secção seguinte, os desempenhos da representação semântica suplantem os da representação morfológica.

8.2 Trabalho futuro

Existem muitas linhas em aberto para trabalho futuro. Por um lado, os resultados devem ser validados aplicando esta metodologia a outros corpora/domínios; por outro, considera-se que, ao aperfeiçoar as ferramentas de processamento da língua, os resultados serão superiores.

Para validar os resultados é necessário utilizar outros corpora. O primeiro passo, e mais simples, será alargar o estudo sobre a influência da informação sintáctica e semântica ao corpus `Publico`; desta forma obteremos um resultado para corpora de maior dimensão. Depois, é necessário aplicar esta metodologia a outros corpora, quer de âmbito geral

como o corpus `Publico`, quer de âmbito mais específico como o `PGR`.

Será também frutuoso perceber se as diferenças de resultados obtidos entre os corpora `Folha` e `Publico` se devem ao tamanho do corpus ou ao facto de o primeiro utilizar Português do Brasil e o último Português Europeu. Para tal é necessário empregar esta metodologia a corpora de outras área de conhecimento para ambas as línguas.

Finalmente, esta abordagem deverá ser explorada em corpora de outras línguas, nomeadamente a língua Inglesa, quer por ser aquela onde existe maior variedade, quer por existir um maior número de trabalhos sobre classificação automática de documentos e, como tal, permitir estudos comparativos mais reveladores. Para este estudo é necessário desenvolver novas aplicações (ou utilizar outras já existentes) correspondentes ao PALAVRAS e SIN2SEM: um analisador sintáctico e um gerador da forma lógica correspondente que utilize Estruturas de Representação de Discurso.

Por outro lado, são conhecidas algumas limitações das ferramentas de processamento de língua natural utilizadas que, ao serem eliminadas, irão tornar a representação semântica do documento mais próxima do seu verdadeiro significado.

Embora a ferramenta PALAVRAS produza sempre um resultado, mesmo em presença de frases incorrectas ou incompletas, existem algumas situações onde a árvore sintáctica gerada é incorrecta; uma dessas situações acontece ao processar frases interrogativas. Como a árvore sintáctica gerada é processada, de seguida, pela ferramenta SIN2SEM o erro propaga-se na geração da ERD respectiva.

A própria ferramenta SIN2SEM pode ser aperfeiçoada. As suas limitações incluem:

- não realizar a subcategorização dos verbos, ou seja, qual o tipo de semântica que cada verbo requer; por exemplo, ao verbo `nascer` podem estar associadas as propriedades `onde` e `quando`, que ao serem utilizadas garantem uma aproximação ao verdadeiro significado dos documentos;
- não tratar correctamente os sintagmas nominais com mais que uma entidade; por exemplo, o resultado do processamento da frase `O João e a Maria jogam à bola`, é:

  ```
  drs(
      [def-A-'<hum>'-f,def-B-'<hum>'-m,def-C-'<food-c-h>'-'<tool>'],
      [nome(A,'Maria'),nome(B,'João'),jogar(A,C),bola(C)]
  )
  ```

  que, como se vê, não representa o sentido da frase.

Mais ainda, e como referido na secção 6.1, esta ferramenta apenas implementa a primeira fase do tratamento de pressupostos da Teoria de Representação do Discurso,

construindo de forma independente uma representação "preliminar" para cada frase. A implementação da segunda fase do tratamento de pressupostos, onde a representação preliminar é verificada no contexto, irá aproximar significativamente a representação do documento ao seu verdadeiro significado.

Por outro lado, a incorporação do resultado de outras análises de informação linguística irá reduzir o número de condições e referentes da representação semântica e, desta forma, traduzir de forma mais próxima o verdadeiro sentido do documento. Por exemplo:

- com a aplicação um algoritmo de resolução de anáforas, o número de referentes a nível do documento irá diminuir;
- com a identificação de Entidades Mencionadas, o número de referentes a nível do documento e do corpus será também reduzida. Para o sistema conseguir inferir que dois referentes distintos, por exemplo `Lisboa` e `capital de Portugal`, correspondem à mesma Entidade, será necessário conjugar a informação contida no documento com uma base de conhecimento sobre Entidades Mencionadas equivalentes;
- com a inclusão de informação sobre sinónimos, o número de condições da representação semântica será, por sua vez, menor.

Anexos

Anexo A

Morfemas e Palavras

Neste anexo apresenta-se a classificação de morfemas gramaticais e as características inerentes a cada uma das classes gramaticais das palavras.

A.1 Morfemas gramaticais

Um morfema gramatical pode ser classificado como:

- uma **desinência**[1]. Indica o género e o número dos nomes, adjectivos e certos pronomes e o número e pessoa dos verbos; unem-se à parte final do radical;

- um **afixo**[2]. É um elemento que modifica de forma precisa o sentido do radical a que se agrega. Chamam-se **prefixos** aos afixos que precedem o radical e **sufixos** aos os que o sucedem;

- uma **vogal temática**. É uma vogal acrescida ao radical dos verbos e que caracteriza a sua conjugação. O radical acrescido da vogal temática, isto é, pronto para receber uma desinência ou sufixo, denomina-se **tema**.

[1] Também conhecido como morfema flexional.
[2] Também conhecido como morfema derivacional.

A.2 Classes gramaticais das palavras

Os nomes

Os nomes são palavras que designam pessoas, animais, plantas, lugares, objectos, acções, estados, qualidades ou sentimentos. É uma classe aberta, já que está sempre a receber novas palavras, à medida que a língua evolui.

Subclasses

De acordo com os seres ou coisas que referem, os nomes agrupam-se em subclasses podendo ser incluídos em mais do que uma:

- um nome é **concreto** quando designa um ser propriamente dito; é **abstracto** quando designa uma noção, acção, estado, qualidade ou sentimento;
- um nome é **comum** quando nomeia pessoas, animais, plantas ou coisas sem os individualizar; é **próprio** quando se aplica a determinado indivíduo da espécie.
- um nome comum é **colectivo** quando designa, no singular, um conjunto de seres ou coisas da mesma espécie.

Flexões

Os nomes podem variar:

- em **número**. Quanto a esta flexão, uma palavra pode estar no **singular**, quando designa um ser único ou um conjunto de seres considerados como um todo, ou no **plural**, quando designa mais que um ser, ou mais que um desses conjuntos orgânicos;
- em **género**. Na língua portuguesa existem dois géneros: o **masculino** e o **feminino**. O masculino é o termo não marcado; o feminino, o termo marcado.
- em **grau**. O nome, além da sua significação **normal**, pode apresentar-se nos graus **aumentativo** (com significação exagerada) ou **diminutivo** (com significação atenuada). Os graus aumentativo e diminutivo não traduzem, necessariamente, a noção de grande ou pequeno, de maior ou menor. Muitas vezes utiliza-se o aumentativo com valor depreciativo; o diminutivo, em determinados contextos, também é depreciativo, noutros tem valor afectivo.

Os adjectivos

O adjectivo é, essencialmente, um modificador do nome. Modifica-lhe o sentido limitando-lhe ou acrescentando-lhe uma qualidade, uma característica ou propriedade.

Flexões

Tal como os nomes, os adjectivos variam em género, número e grau. Concordam em género e número com o nome a que estão ligados. Quando o mesmo adjectivo qualifica nomes de géneros diferentes, utiliza-se o masculino do adjectivo; quando qualifica vários nomes no singular o adjectivo vai para o plural.

O sentido expresso pelo adjectivo pode variar em **grau** de intensidade (sem referência a outros elementos) ou por comparação com outros:

- no grau **normal**, o adjectivo atribui uma característica ao nome;
- no grau **comparativo**, o adjectivo atribui uma característica com a qual se estabelece uma comparação entre os nomes. Pode ser de superioridade, de inferioridade ou de igualdade;
- no grau **superlativo absoluto** a característica atribuída pelo adjectivo estabelece um valor sem recorrer a uma comparação. Pode ser analítico ou sintético;
- no grau **superlativo relativo** a característica atribuída compara um elemento com toda uma classe. Pode ser de superioridade ou de inferioridade.

Os artigos

Dá-se o nome de artigo às palavras que se antepõem aos substantivos para indicar:

- que se trata de um ser já conhecido, seja por ter sido mencionado antes, seja por ser objecto de um conhecimento ou experiência;
- que se trata de um simples representante de uma dada espécie ao qual não se fez menção anterior.

No primeiro caso dizemos que o artigo é **definido**, no segundo **indefinido**.

Existem formas simples e formas combinadas dos artigos, quer definidos quer indefinidos. As formas são combinadas quando o artigo vem precedido de uma preposição.

Flexões

Os artigos concordam em género e em número com o nome, sendo, por vezes, através deles que sabemos se o nome é masculino ou feminino, se é singular ou plural.

Os pronomes

O pronome substitui habitualmente o grupo nominal, evitando repetições desnecessárias. Pode também, nalguns casos, substituir o adjectivo ou mesmo uma frase inteira.

Subclasses

Existem seis subclasses de pronomes: pessoais, possessivos, demonstrativos, relativos, interrogativos e indefinidos.

Enquanto os pronomes **pessoais** denotam as pessoas gramaticais, os pronomes possessivos e demonstrativos indicam algo determinado por elas. Os pronomes **possessivos** acrescentam à noção de pessoa gramatical uma ideia de posse; os **demonstrativos** situam (no espaço ou tempo) a pessoa ou a coisa designada relativamente à pessoa gramatical. Os pronomes demonstrativos podem ter, também, uma função anafórica quando se empregam para lembrar ao leitor o que já foi mencionado ou o que se vai mencionar.

Os pronomes **relativos** são assim chamados porque se referem, regra geral, a um termo anterior – o antecedente. O antecedente pode ser um substantivo, um pronome, um adjectivo, um advérbio ou uma oração.

Chamam-se **interrogativos** os pronomes empregues para formular uma pergunta directa ou indirecta e **indefinidos** os pronomes que se aplicam à 3ª pessoa gramatical, quando considerada de um modo vago e indeterminado. Aqueles estão estritamente ligados a estes: nuns e noutros a significação é indeterminada, embora, no caso dos interrogativos, a resposta, em geral, venha esclarecer o que foi perguntado.

Flexões

Os pronomes pessoais apresentam variação em género, número e pessoa, mas nos pronomes possessivos, para cada pessoa as formas variam de acordo com o género e número da coisa possuída e com o número de pessoas representadas no possuidor.

Os pronomes demonstrativos, relativos e indefinidos podem ser invariáveis ou variáveis de acordo com o género e número. Os relativos podem, por outro lado, ser de forma

simples ou composta.

Existem pronomes interrogativos invariáveis, outros flexionam-se em número e outros ainda em género e número.

A variação em **pessoa** tem, através das três pessoas gramaticais, a capacidade de indicar: quem fala – **1ª pessoa**, com quem se fala – **2ª pessoa** ou de quem se fala – **3ª pessoa**.

Numerais

Os numerais são uma classe especial de palavras empregues para indicarmos uma quantidade exacta de pessoas ou coisas, ou para assinalar o lugar que elas ocupam numa série.

Subclasses

Os numerais podem ser cardinais, ordinais, multiplicativos ou fraccionários. Os **cardinais** são os números básicos e os **ordinais** indicam a ordem de sucessão dos seres ou objectos numa dada série. Os numerais ordinais equivalem a adjectivos que, no entanto, se substantivam facilmente.

Enquanto os numerais **multiplicativos** indicam o aumento proporcional da quantidade, os **fraccionários** exprimem a sua diminuição. Podem equivaler a adjectivos e, com mais frequência, a substantivos por virem geralmente antecedidos de artigo.

Flexões

Existem numerais cardinais que variam em género, outros em número e outros ainda são invariáveis; os numerais ordinais variam em género e número.

Os numerais multiplicativos são invariáveis quando equivalem a substantivos mas, quando empregues como adjectivos, flexionam-se em género e número. Já os fraccionários concordam com os cardinais que indicam o número das partes.

Os verbos

O verbo é a classe de palavras que nos transmite as noções de processo, acção, estado ou modificação.

Subclasses

Os verbos podem ser classificados quando às variações permitidas e quanto à sua função na oração.

Classificação segundo a variação. Ao classificar um verbo segundo as variações permitidas, um verbo pode ser regular, irregular, defectivo ou abundante. É **regular** se, ao ser conjugado, mantém sempre a mesma raíz[3]. Se, por outro lado, a raíz sofrer alteração na conjugação verbal, estamos perante um verbo **irregular**.

Os verbos **defectivos** são aqueles que não têm certas formas. Entre eles, o impessoais só aparecem na 3ª pessoa do singular (verbos que exprimem fenómenos da natureza) e os unipessoais (normalmente) só se empregam na 3ª pessoa do singular e plural (verbos que indicam vozes de animais).

Os verbos **abundantes** apresentam duas ou mais formas equivalentes. Em regra, essa abundância ocorre no particípio passado.

Classificação segundo a função. Um verbo, classificado segundo a sua função, pode ser principal[4] ou auxiliar[5]. O verbo **principal** é aquele que traz uma ideia nova ao sujeito. Constitui o núcleo do grupo verbal e determina a existência ou não do sujeito e dos complementos (ver Anexo B). Estes verbos apresentam a seguinte classificação:

- verbo intransitivo. A acção está integralmente contida na forma verbal. O predicado não tem complementos.
- verbo transitivo directo. A acção transmite-se a outros elementos directamente, ou seja, sem o auxílio de preposição. O predicado completa-se com o complemento directo.
- verbo transitivo indirecto. A acção transmite-se a outros elementos da oração indirectamente, isto é, por meio de preposição. A ideia completa-se por um complemento indirecto, um complemento preposicional ou um complemento adverbial.
- verbo transitivo directo e indirecto. A acção transmite-se a outros elementos da oração, directa e indirectamente. O predicado completa-se com o complemento directo mais os complementos indirecto, preposicional ou adverbial.

[3] A raíz do verbo é obtida retirando a sua terminação que consiste na vogal temática seguida do sufixo `-r`.

[4] Também designado por significativo.

[5] Também designado por copulativo ou verbo de ligação.

- verbo impessoal. Ao contrário dos verbos atrás referidos, o verbo impessoal não tem sujeito.

O verbo **auxiliar** serve para estabelecer a união entre duas palavras ou expressões de carácter nominal. Não traz propriamente uma ideia nova ao sujeito; funciona apenas como elo entre este e o seu predicativo.

Flexões

As variações do verbo em número e pessoa concordam com o sujeito (ver Anexo B). O verbo permite ainda variações em:

- **Modo**. Chamam-se modos às diferentes formas que o verbo toma para indicar a atitude como o sujeito encara a realização do processo expresso pelo verbo. Na língua Portuguesa existem três modos: o indicativo, o conjuntivo e o imperativo. Enquanto o modo **indicativo** expressa um facto, uma certeza, o **conjuntivo** expressa uma hipótese, uma dúvida ou um desejo e o **imperativo**, uma ordem, um pedido ou um conselho.

  Existem ainda três formas nominais do verbo: o **infinitivo**, o **gerúndio** e o **particípio passado**. As formas nominais são formas impessoais, não flexionáveis em pessoa e número, que podem desempenhar funções idênticas às do nome (ou do adjectivo).

- **Tempo**. O tempo verbal é a variação que indica o momento em que se dá o facto enunciado pelo verbo. Os três tempos naturais são o **presente**, o **pretérito**[6] e o **futuro**, que designam, respectivamente, um facto ocorrido no momento em que se fala, antes do momento em que se fala e após o momento em que se fala.

  Os tempos futuro nos modos indicativo e conjuntivo e pretérito no modo indicativo podem ser simples ou compostos. Os compostos constroem-se com diferentes formas do verbo auxiliar `ter` seguido do particípio passado do verbo principal.

  Nos modos indicativo e conjuntivo, o pretérito pode ser imperfeito, perfeito ou mais-que-perfeito; no indicativo, o futuro pode ainda ser do presente ou do pretérito[7].

  O modo imperativo apresenta-se apenas no tempo verbal presente.

- **Voz**. O facto expresso pelo verbo pode, ainda, ser representado através de três formas distintas: como praticado pelo sujeito (voz **activa**), como sofrido pelo sujeito (voz **passiva**) ou como praticado e sofrido pelo sujeito (voz **reflexiva**).

[6] Também designado por passado.
[7] Também designado por condicional.

Conjugação

Tendo por base a terminação do infinitivo consideram-se, na língua Portuguesa, três conjugações[8] de acordo com vogal temática do verbo: a vogal temática `-a-` corresponde à **1ª conjugação**, a vogal temática `-e-` corresponde à **2ª conjugação** e a vogal temática `-i-` à **3ª conjugação**.

Os advérbios

O advérbio é, fundamentalmente, um modificador do verbo.

Subclasses

Os advérbios, consoante a circunstância ou outra ideia acessória que expressam, recebem a denominação de advérbio de negação, de lugar, de modo ou de tempo, de afirmação ou de dúvida, de exclusão ou designação, de intensidade, de ordem ou interrogativo (por se empregarem nas interrogações directas e indirectas) de causa, lugar, modo ou tempo.

Denomina-se **locução adverbial** o conjunto de duas ou mais palavras que funcionam como advérbio. Em regra, a locução forma-se pela associação de uma preposição com um substantivo, adjectivo ou advérbio.

Flexões

Os advérbios são sempre invariáveis em género e número mas alguns, à semelhança do adjectivo, admitem variação em grau.

As conjunções

As conjunções são vocábulos gramaticais que servem para relacionar duas orações ou dois termos semelhantes da mesma oração. As **locuções conjuncionais** têm a mesma função, mas são constituídas por uma sequência de duas ou mais palavras.

[8] A conjugação é o agrupamento de todas as flexões do verbo, segundo uma ordem determinada.

Subclasses

As conjunções e locuções conjuncionais distribuem-se em duas subclasses: as **coordenativas** relacionam termos ou orações de idêntica função gramatical e as **subordinativas** ligam duas orações, uma das quais determina ou completa o sentido da outra.

As conjunções coordenativas dividem-se em aditivas (ou copulativas), adversativas, alternativas (ou disjuntivas), conclusivas ou explicativas. As subordinativas classificam-se em causais, concessivas, condicionais, finais, temporais, comparativas, consecutivas e integrantes (ou completivas).

As preposições

As preposições são uma classe de palavras invariáveis cuja função consiste em estabelecer relações entre elementos da frase.

As **locuções prepositivas** têm a mesma função das preposições e são formadas por um grupo fixo de palavras sendo, pelo menos a última, uma preposição.

As interjeições

As interjeições transmitem espontaneamente uma emoção. Classificam-se segundo o sentimento que denotam: de alegria, de animação, de aplauso, de desejo, de dor, de espanto ou surpresa, de impaciência, de invocação, de silêncio, de suspensão ou de terror.

As **locuções interjectivas** são interjeições formadas por grupos de duas ou mais palavras.

Anexo B

Períodos e orações

Neste anexo apresentam-se os tipos de frase, a classificação de períodos compostos e as funções e constituintes sintácticos da oração.

B.1 Tipos de frase

A intenção comunicativa do emissor de uma frase determina o tipo de frase utilizada:

- quando a frase informa, refere, descreve ou caracteriza, estamos em presença de uma frase **declarativa**;
- quando formula uma pergunta ou apresenta uma dúvida a frase é **interrogativa**;
- quando expressa sentimentos e emoções a frase é **exclamativa**;
- quando transmite um conselho, uma ordem, um pedido ou chamada de atenção, a frase é **imperativa**.

B.2 Classificação de períodos compostos

Um período composto é constituído por duas ou mais orações. Essas orações podem estar ligadas entre si por coordenação, subordinação ou ambas.

Coordenação

Quando as orações de um período são autónomas, tendo cada uma um sentido próprio e não existindo entre elas nenhuma relação de dependência, diz-se que as orações são **coordenadas** e o período por elas formado diz-se composto por coordenação.

As orações coordenadas podem estar simplesmente justapostas, isto é, colocadas uma ao lado da outra sem qualquer conectivo ou ligadas por uma conjunção coordenativa. No primeiro caso, dizemos que a oração é **assindética**; no segundo, a oração é **sindética**.

As orações coordenadas sindéticas são classificadas de acordo com a conjunção coordenativa que as une: aditiva, adversativa, alternativa, conclusiva ou explicativa).

Subordinação

Ao contrário do que acontece com o período composto por coordenação, no período composto por subordinação, existem orações **subordinadas**, que estão dependentes de outra – a **principal**[1].

As orações subordinadas classificam-se em substantivas, adjectivas ou adverbiais de acordo com as funções desempenhadas no período serem comparáveis às exercidas por substantivos, adjectivos ou advérbios.

Consoante a relação que estabelecem com a oração subordinante, as orações subordinadas classificam-se como: causais, temporais, condicionais, finais, comparativas, concessivas, consecutivas ou integrantes.

B.3 Funções sintácticas dos elementos da oração

Uma oração contém obrigatoriamente um sujeito e um predicado. O **sujeito** é o ser sobre o qual se faz uma declaração; o **predicado** é tudo aquilo que se diz acerca do sujeito.

Por exemplo, na oração *Este aluno obteve ontem uma boa nota*, o sujeito é *este aluno* e o predicado *obteve ontem uma boa nota*.

O predicado, para além do verbo é constituído por outros elementos funcionais: complementos, modificadores e predicativo do sujeito. Numa oração podem ainda existir elementos com outras funções sintácticas como o vocativo.

[1] Também designada como subordinante.

Sujeito

O sujeito pode ser representado por um nome ou pronome, um numeral, uma palavra ou expressão substantivada (utilizada como nome) ou uma oração.

O sujeito de uma oração pode ser simples, composto, oculto ou indeterminado. É **simples** quando é constituído por um só núcleo; é **composto** quando tem mais que um núcleo. Quando o sujeito não está materialmente expresso na oração, mas pode ser identificado, é **oculto**, mas quando o verbo não se refere a uma pessoa determinada (por se desconhecer quem executa a acção ou por não haver interesse no seu conhecimento), dizemos que o sujeito é **indeterminado**.

Existem ainda orações em que o sujeito é **inexistente**. Acontece quando no processo verbal não é atribuído a nenhum ser e, neste caso, diz-se que o verbo é impessoal.

Predicado

O predicado é uma função sintáctica desempenhada pelo verbo e por todos os seus termos integrantes.

O predicado pode ser nominal, verbal ou verbo-nominal. É **nominal** quando o predicado é formado por um verbo de ligação e predicativo. O predicativo pode ser representado por um substantivo ou expressão substantivada, um adjectivo ou locução adjectiva, um pronome, um numeral ou por uma oração substantiva predicativa.

Quando o núcleo do predicado é um verbo principal, estamos em presença de um predicado **verbal**, mas se o núcleo for constituído por um verbo principal e predicativo, o predicado denomina-se **verbo-nominal**.

Termos integrantes do predicado

Os termos integrantes do predicado podem ser complementos, modificadores ou o predicativo do sujeito. Enquanto os complementos são elementos obrigatórios quando exigidos pelo verbo, os modificadores são facultativos. Os complementos podem ser: directo, indirecto, preposicional ou adverbial e os modificadores podem ser preposicional ou adverbial:

- **Complemento directo**. É o complemento de um verbo transitivo directo. Normalmente vem ligado ao verbo sem preposição e indica o ser para o qual se dirige a acção verbal.
- **Complemento indirecto**. É o complemento de um verbo transitivo indirecto e

liga-se ao verbo por meio de uma preposição. É um nome ou pronome que indica sobre quem se exerce indirectamente a acção expressa pelo verbo.

- **Complemento preposicional**. É um elemento introduzido por uma preposição, quando esta é exigida pelo verbo.
- **Complemento adverbial**. É um elemento constituído por um advérbio ou locução adverbial, quando este é exigido pelo verbo.
- **Modificador preposicional**. É um elemento introduzido por uma preposição. Ao contrário do complemento preposicional, o modificador não é exigido pelo verbo ou qualquer outro constituinte da frase, pelo que desempenha uma função circunstancial.
- **Modificador adverbial**. É um elemento constituído ou introduzido por um advérbio ou locução adverbial. Como o modificador preposicional, desempenha, também, uma função circunstancial.
- **Predicativo do sujeito**. É uma palavra (ou expressão) que se junta aos verbos auxiliares (`ser`, `estar`, `parecer`, `ficar`, etc.) para lhes completar o sentido e que identifica ou caracteriza o sujeito.

Vocativo

É um elemento de entoação exclamativa que serve para invocar, chamar ou nomear, com ênfase maior ou menor, uma pessoa ou coisa personificada.

B.4 Constituintes sintácticos da oração

Um **sintagma** é um constituinte sintáctico formado por um grupo de palavras que, por sua vez, possui outros sub-constituintes e onde pelo menos um se designa por núcleo. O(s) **núcleo**(s) é (são) a(s) palavra(s) que determina(m) as propriedades sintácticas e de combinação do sintagma a que ele pertence.

Todas as orações são decomponíveis em sintagmas e estes podem depender, encaixar ou ligar-se com outros mediante relações sintácticas de coordenação ou subordinação, relações morfo-sintácticas de concordância ou ainda relações semânticas.

A composição interior de um sintagma varia; existem sintagmas com uma única palavra que funciona como núcleo até aqueles em que vários sintagmas se encontram dependentes de um central. Um sintagma pode ser ou não obrigatório consoante é requerido ou não pelo predicado.

Consoante o tipo de núcleo, o sintagma pode ser nominal, verbal, adjectivo, adverbial ou preposicional. Algumas gramáticas consideram ainda o sintagma determinante:

- No **sintagma nominal** o núcleo é um nome (comum ou próprio) ou um pronome. Na língua Portuguesa pode funcionar como sujeito, predicativo do sujeito, complemento directo (de coisa) ou complemento/modificador adverbial. Se o núcleo for um pronome pode funcionar como complemento directo (de pessoa) ou complemento indirecto.

  Podem ocorrer muitos sintagmas nominais numa oração, mas apenas um deles terá a função de sujeito; os restantes sintagmas nominais encaixam-se no predicado.

- No **sintagma verbal**, caso exista um verbo principal, este será o seu núcleo. O sintagma verbal funciona sempre como predicado.

- O **sintagma preposicional**, cujo núcleo é uma preposição, pode funcionar como complemento directo ou indirecto, complemento/modificador adverbial, predicativo do sujeito ou complemento/modificador preposicional.

- O **sintagma adverbial**, cujo núcleo é um advérbio, pode funcionar como complemento/modificador adverbial ou como grau de um adjectivo.

- O adjectivo é o núcleo do **sintagma adjectivo** e este pode funcionar como predicativo do sujeito.

Anexo C

Redução de dimensão

Neste anexo descrevem-se as técnicas mais usuais de redução de dimensão. Embora a apresentação assuma uma representação saco-de-palavras, a sua maioria pode ser aplicada a outras representações de forma simples.

A redução de termos pode ser realizada através de um processo de selecção ou de extracção de novos termos. Uma terceira técnica, a poderação de características, pode ser considerada uma forma "leve" de selecção.

C.1 Selecção de termos

As técnicas de selecção de termos tentam escolher, a partir do conjunto original de termos $\mathcal{T}$, o conjunto de termos $\mathcal{T}'$ que, quando utilizados, produzam um desempenho máximo. Yang and Pedersen [1997] mostraram que esta selecção pode até produzir um acréscimo moderado ($\leq$ 5%) no desempenho, dependendo do classificador, da agressividade da redução ($\frac{|T|}{|T'|}$) e da técnica de selecção utilizada.

A **eliminação de *stopwords***, uma técnica aplicada exclusivamente a texto, pode ser considerada uma selecção de termos já que, do conjunto original de palavras, escolhe todas aquelas que não pertençam à lista de *stopwords*. Estas, são palavras (artigos, pronomes, advérbios e preposições) não relevantes uma vez que não possuem poder discriminante sobre as classes devido ao seu papel na frase estar relacionado com a sua organização estrutural.

O método da **frequência dos documentos**[1], $\#T_r(t_k)$, retém os termos t_k (depois de eliminadas as *stopwords*) que ocorrem um maior número de vezes no conjunto dos documentos. Se considerarmos o conjunto de treino como espaço de entrada, esta me-

[1]Do inglês, *document frequency*.

dida pode ser escrita em termos probabilísticos como $P(t_k)$. Yang and Pedersen [1997] mostram que, independentemente do classificador adoptado e da colecção utilizada, é possível obter uma redução da dimensão de um factor de 10 sem perda de desempenho. Variações deste método incluem remover os termos que ocorrem no máximo em m documentos de treino (por exemplo, [Dumais et al. 1998; Maron 1961]) ou aqueles que ocorrem no máximo x vezes no conjunto de treino (por exemplo, [Baker and McCallum 1998; Joachims 1998]).

Têm também sido aplicadas outras funções mais sofisticadas, baseadas em **teoria da informação**, tais como: o factor de associação DIA [Fuhr et al. 1991], o chi-quadrado [Caropeso et al. 2001; Galavotti et al. 2000; Schütze et al. 1995; Sebastiani et al. 2000; Yang and Liu 1999; Yang and Pedersen 1997], o coeficiente NGL [Ng et al. 1997; Ruiz and Srinivasan 1999], a dependência [Dumais et al. 1998; Lam et al. 1997; Larkey and Croft 1996; Lewis and Ringuette 1994; Li and Jain 1998; Moulinier and Ganascia 1996; Ruiz and Srinivasan 1999; Taira and Haruno 1999; Yang and Pedersen 1997], a informação mútua[2] [Caropeso et al. 2001; Larkey 1998; Lewis 1992a; Lewis and Ringuette 1994; Mladenić 1998; Moulinier and Ganascia 1996; Yang and Liu 1999; Yang and Pedersen 1997], a razão de vantagem[3] [Caropeso et al. 2001; Mladenić 1998; Ruiz and Srinivasan 1999], a medida de relevância[4] [Wiener et al. 1995] e o coeficiente GSS [Galavotti et al. 2000].

Aos contrário das anteriores, estas funções analisam as etiquetas das classes para seleccionar os termos. A sua maioria tenta captar a ideia de que os "melhores" termos para a classe c_i são aqueles que possuem uma distribuição mais distinta entre os conjuntos de exemplos positivos e negativos de c_i.

A Tabela C.1 sumariza as definições matemáticas destas medidas. As probabilidades são interpretadas de forma usual no espaço de documentos (por exemplo $P(\overline{t}_k, c_i)$ é a probabilidade do termo t_k não ocorrer num documento d, com d pertencente à classe c_i) e são estimadas contando o número de ocorrências no conjunto de treino. As funções $f(t_k, c_i)$ são especificadas "localmente" para uma classe específica c_i. Para obter o valor do termo t_k "globalmente", independente da classe é comum utilizar:

- a soma, $f_{sum}(t_k) = \sum_{i=1}^{|C|} f(t_k, c_i)$,
- a média pesada, $f_{med}(t_k) = \sum_{i=1}^{|C|} P(c_i) f(t_k, c_i)$, ou
- o máximo, $f_{max}(t_k) = max_{i=1}^{|C|} f(t_k, c_i)$

dos valores das categorias.

Variadas comparações experimentais destas funções têm sido aplicadas na área de CT [Caropeso et al. 2001; Galavotti et al. 2000; Mladenić 1998; Ng et al. 1997; Yang

[2]Também conhecida por ganho de informação.
[3]Do inglês, *odds ratio.*
[4]Do inglês, *relevancy score.*

Função	Notação	Forma matemática
Frequência documento	$\#(t_k, c_i)$	$\mathrm{P}(t_k \mid c_i)$
Factor DIA	$z(t_k, c_i)$	$\mathrm{P}(c_i \mid t_k)$
Chi-quadrado	$\chi^2(t_k, c_i)$	$\frac{\lvert T_r \rvert \cdot [\mathrm{P}(t_k, c_i) \cdot \mathrm{P}(\overline{t}_k, \overline{c}_i) - \mathrm{P}(t_k, \overline{c}_i)) \cdot \mathrm{P}(\overline{t}_k, c_i)]^2}{\mathrm{P}(t_k) \cdot \mathrm{P}(\overline{t}_k) \cdot \mathrm{P}(c_i) \cdot \mathrm{P}(\overline{c}_i)}$
Coeficiente NGL	$NGL(t_k, c_i)$	$\sqrt{\chi^2(t_k, c_i)}$
Dependência	$DEP(t_k, c_i)$	$\log_2 \frac{\mathrm{P}(t_k, c_i)}{\mathrm{P}(t_k) \cdot \mathrm{P}(c_i)}$
Informação Mútua	$IM(t_k, c_i)$	$\mathrm{P}(t_k, c_i) \cdot DEP(t_k, c_i) + \mathrm{P}(\overline{t}_k, c_i) \cdot DEP(\overline{t}_k, c_i)$
Razão de vantagem	$RV(t_k, c_i)$	$\frac{\mathrm{P}(t_k \mid c_i) \cdot (1 - \mathrm{P}(t_k \mid \overline{c}_i))}{(1 - \mathrm{P}(t_k \mid c_i)) \cdot \mathrm{P}(t_k \mid \overline{c}_i)}$
Medida de relevância	$MR(t_k, c_i)$	$\log_2 \frac{\mathrm{P}(t_k \mid c_i) + d}{\mathrm{P}(\overline{t}_k \mid \overline{c}_i) + d}$
Coeficiente GSS	$GSS(t_k, c_i)$	$\mathrm{P}(t_k, c_i) \cdot \mathrm{P}(\overline{t}_k, \overline{c}_i) - \mathrm{P}(t_k, \overline{c}_i) \cdot \mathrm{P}(\overline{t}_k, c_i)$

Tabela C.1: Principais funções utilizadas na selecção de termos.

and Pedersen 1997]. Nestas experiências, a maioria das funções listadas produziram melhores resultados do que a frequência dos documentos. Por exemplo, Yang and Pedersen [1997] mostraram que, com diversos classificadores e várias colecções, as técnicas $IM_{sum}(t_k, c_i)$ e $\chi^2_{max}(t_k, c_i)$ conseguem reduzir a dimensão do espaço de termos de um factor de 100 sem perdas de desempenho.

Como referido em [Sebastiani 2002] as experiências reportadas nos artigos mencionados parecem indicar que $\{RV_{sum}, NGL_{sum}, GSS_{max}\} > \{\chi^2_{max}, IM_{sum}\} > \{\#_{avg}, \chi^2_{avg}\} \gg \{DEP_{max}, DEP_{avg}\}$ (onde ">" significa "é mais eficaz que"). No entanto, estes resultados são meramente indicativos e conclusões mais gerais sobre os méritos de cada função podem apenas ser feitos como resultado de um estudo comparativo realizado com experiências em condições controladas e numa variedade de diferentes situações (classificadores diferentes, diferentes colecções,...).

C.2 Extracção de termos

A partir do conjunto original $\mathcal{T}$, a extracção de termos pretende maximizar o desempenho do classificador através da síntese de um conjunto de novos termos $\mathcal{T}'$, com $|\mathcal{T}'| \ll |\mathcal{T}|$. A razão para utilizar termos sintéticos (em vez originais que aparecem nos documentos) é que, devido à existência de palavras sinónimas, homónimas e po-

lissémicas, os termos originais podem não ser os melhores para representar do conteúdo dos documentos. Os métodos de extracção de termos tentam resolver este problema através da criação de termos artificiais que não sofram de tais problemas.

Qualquer técnica de extracção de termos deve especificar a forma de extrair os novos termos a partir dos originais e um método para converter a representação original do documento, numa nova representação baseada nas novas dimensões sintetizadas. Em CT têm sido experimentados dois métodos de extracção: agrupamento de termos[5] e indexação semântica latente[6].

O **agrupamento de termos** tem como objectivo agrupar palavras com um elevado grau de relação semântica, para que os grupos (representados pelo seu centróide ou equivalente) possam ser utilizados como dimensões do espaço vectorial em substituição dos termos originais.

Lewis [1992a] foi o primeiro a investigar a utilização do agrupamento de termos aplicando um método chamado agrupamento dos vizinhos mais próximos recíprocos[7]. Este método, que consiste em criar grupos de dois termos que sejam os mais parecidos de acordo com uma medida de semelhança, apresentou piores resultados que os obtidos se os termos não fossem agrupados. Por outro lado, Li and Jain [1998] mediram a relação semântica entre termos através da sua co-ocorrência e co-ausência nos documentos de treino, obtendo resultados com melhorias marginais. Estes são exemplos de métodos de agrupamento não supervisionado já que não utilizam a informação das etiquetas das classes durante o processo de agrupamento.

Ao contrário dos anteriores, Baker and McCallum [1998] utilizam um método supervisionado, o agrupamento distribuído[8], que agrupa os termos que tendem a indicar a presença da mesma classe ou grupos de classes. As suas experiências mostraram uma perca de 2% do desempenho para uma agressividade de 1000 e uma melhoria de resultados para níveis de agressividade mais baixos.

A **indexação semântica latente**[9] (ISL) [Deerwester et al. 1990] é uma técnica desenvolvida no âmbito da RI para tentar solucionar os problemas derivados da utilização de sinónimos, quase-sinónimos e palavras polissémicas na representação de documentos e questões. Esta técnica transforma os vectores que representam os documentos e questões em vectores num espaço de menor dimensão, cujos termos são obtidos através da combinação das dimensões originais utilizando padrões de co-ocorrência. Na prática, a ISL infere a dependência entre os termos originais de uma colecção, mapeando estas dependências num conjunto de novos termos independentes. Esta transformação é

[5]Do inglês, *term clustering*.
[6]Do inglês, *latent semantic indexing*.
[7]Do inglês, *reciprocal nearest neighbour clustering*.
[8]Do inglês, *distributional clustering*.
[9]Do inglês, *latent semantic indexing*.

obtida através da decomposição em valores singulares da matriz formada pelos vectores originais dos documentos. No contexto de CT, utiliza-se conjunto de treino para calcular a transformação; esta é depois aplicada aos documentos dos conjuntos de treino e teste para produzir uma representação no novo espaço de dimensões.

Uma característica da ISL como função de redução de dimensão é que as novas dimensões, ao contrário dos casos de selecção e agrupamento de termos, não são interpretáveis de forma intuitiva, mas tendem a evidenciar a estrutura semântica "latente" no vocabulário utilizado na colecção. Como Schütze et al. [1995] referem, se existirem muitos termos que contribuem com pouca informação crítica, a combinação de evidências torna-se um problema para um classificador baseado na selecção de termos. Por outro lado, se existir um termo original que consiga, sozinho, discriminar uma categoria, este poder de discriminação pode perder-se no novo espaço de dimensões quando se utiliza classificador baseado na ISL.

Schütze et al. [1995] comparam a extracção de termos baseada em ISL com a selecção de termos utilizando o χ^2, utilizando três algoritmos de indução (análise discriminante linear, regressão logística e redes neuronais) numa aplicação de encaminhamento. Os seus resultados mostram que a ISL é muito mais eficaz que o χ^2 utilizando os dois primeiros algoritmos, mas os métodos são equivalentes utilizando redes neuronais. Outros trabalhos de CT que utilizam a ISL ou técnicas similares são [Hull 1994; Li and Jain 1998; Schütze 1998; Weigend et al. 1999; Yang 1995].

C.3 Ponderação de características

Intuitivamente, a ponderação de características constitui uma forma "leve" de selecção. Enquanto a selecção remove certas dimensões de informação, a ponderação apenas ajusta a sua influência relativa.

Os métodos aqui apresentados foram desenvolvidos originalmente para RI [Salton 1971, 1991], mas têm sido utilizados com êxito em CT. Contêm, usualmente, três componentes [Salton and Buckley 1988]. A **componente do documento** captura estatísticas sobre um determinado termo num documento particular. A medida básica é a frequência do termo[10] $\mathrm{ft}(p_i, d_j)$, definida pelo número de vezes que a palavra p_i aparece no documento d_j. Esta medida corresponde à ideia de que as palavras que ocorrem com mais frequência num documento são mais relevantes que as pouco frequentes.

No entanto, um termo frequente que ocorra em quase todos os documentos não ajudará na discriminação entre classes. A **componente da colecção** foi desenhada para atribuir um peso menor a estes termos. A medida básica é a frequência do documento[11]

[10] Do inglês, *term frequency*.
[11] Do inglês, *document frequency*.

$\text{fd}(p_i)$, representada pelo número de documentos em que a palavra p_i aparece pelo menos uma vez. Se a frequência for grande, o peso do termo é reduzido.

Finalmente, como os documentos podem ser de diferentes tamanhos é necessário aplicar uma **componente de normalização** que ajusta os pesos para que documentos pequenos e grandes possam ser comparados na mesma escala.

A Tabela C.2 apresenta as escolhas mais frequentes para cada componente. No vector de características final $\overrightarrow{x}$, o valor de x_i correspondente à palavra p_i, é calculado multiplicando as três componentes.

Componente	Notação	Forma matemática	Descrição
Documento	b	1.0	binário (o peso é 1 para os termos presentes no documento)
	t	$\text{ft}(p_i, d)$	frequência do termo
	n	$0.5 + 0.5\frac{\text{ft}(p_i, d)}{max_j \text{ft}(p_j, d)}$	frequência do termo aumentada e normalizada (o impacto entre altas e baixas frequências é reduzido)
Colecção	x	1.0	sem componente de colecção
	f	$\ln \frac{\lvert\mathcal{D}\rvert}{\text{fd}(p_i)}$	frequência do documento inversa ($\lvert\mathcal{D}\rvert$ é o número total de documentos na colecção)
	n	$\ln \frac{\lvert\mathcal{D}\rvert - \text{fd}(p_i)}{\text{fd}(p_i)}$	frequência do documento inversa probabilística
Normalização	x	1.0	sem componente de normalização
	c	$\frac{1}{\sqrt{\sum x_j^2}}$	normalização do co-seno (vector resultado para comprimento 1, ou seja, L_2)
	a	$\frac{1}{\sum x_j}$	normalização do vector resultado em termos de L_1

Tabela C.2: Componentes de ponderação de termos mais utilizadas.

Embora existam trabalhos com diferentes combinações das três componentes, a composição mais utilizada é, sem dúvida, a resultante da notação `tfc`, a conhecida medida $tfidf$ Salton and Buckley [1988] com normalização do co-seno.

Anexo D

Espaços vectoriais e de Hilbert

Este anexo introduz os conceitos matemáticos básicos necessários ao Capítulo 5. A Secção D.1 define a estrutura dos espaços vectoriais e a Secção D.2 apresenta normas e produtos internos. Enquanto a primeira apresenta a estrutura linear dos espaços vectoriais, a segunda apresenta a sua estrutura métrica. Detalhes adicionais podem ser encontrados em qualquer livro sobre álgebra linear. Os espaços de Hilbert são definidos na Secção D.3.

D.1 Espaços vectoriais

Definição D.1 (Espaço vectorial)**.** *Um conjunto $\mathcal{H}$ designa-se por* ***espaço vectorial*** *(ou espaço linear) sobre $\mathbb{R}$ quando a adição e a multiplicação escalar estão definidas e satisfazem (para todo $\mathbf{x}, \mathbf{x}', \mathbf{x}'' \in \mathcal{H}$ e $\lambda, \lambda' \in \mathbb{R}$)*

$$\mathbf{x} + (\mathbf{x}' + \mathbf{x}'') = (\mathbf{x} + \mathbf{x}') + \mathbf{x}'', \tag{D.1}$$
$$\mathbf{x} + \mathbf{x}' = \mathbf{x}' + \mathbf{x} \in \mathcal{H}, \tag{D.2}$$
$$\exists 0 \in \mathcal{H}, \mathbf{x} + 0 = \mathbf{x}, \tag{D.3}$$
$$\exists - \mathbf{x} \in \mathcal{H}, -\mathbf{x} + \mathbf{x} = 0, \tag{D.4}$$
$$\lambda \mathbf{x} \in \mathcal{H}, \tag{D.5}$$
$$1\mathbf{x} = \mathbf{x}, \tag{D.6}$$
$$\lambda(\lambda' \mathbf{x}) = (\lambda\lambda')\mathbf{x}, \tag{D.7}$$
$$\lambda(\mathbf{x} + \mathbf{x}') = \lambda\mathbf{x} + \lambda\mathbf{x}', \tag{D.8}$$
$$(\lambda + \lambda')\mathbf{x} = \lambda\mathbf{x} + \lambda'\mathbf{x}. \tag{D.9}$$

As primeiras quatro condições correspondem a dizer que $(\mathcal{H}, +)$ constitui um grupo comutativo.

Esta definição restringe-se a espaços vectoriais sobre $\mathbb{R}$. Espaços vectoriais sobre números complexos são análogos tanto na definição, como na maioria das definições seguintes. Qualquer subespaço não vazio de $\mathcal{H}$ que seja ele próprio um espaço vectorial designa-se por **subespaço** de $\mathcal{H}$.

Num espaço vectorial é possível fazer **combinações lineares**

$$\sum_{i=1}^{m} \lambda_i \mathbf{x}_i, \text{ onde } \lambda_i \in \mathbb{R}, \mathbf{x}_i \in \mathcal{H} \tag{D.10}$$

e **combinações convexas**

$$\sum_{i=1}^{m} \lambda_i \mathbf{x}_i, \text{ onde } \lambda_i \geq 0, \sum_i \lambda_i = 1, \mathbf{x}_i \in \mathcal{H} \tag{D.11}$$

O conjunto $\{\sum_{i=1}^{m} \lambda_i \mathbf{x}_i\}$ designa-se por alcance[1] dos vectores $\mathbf{x}_1, \ldots, \mathbf{x}_m$.

Um conjunto de vectores $\mathbf{x}_i$, escolhidos de tal forma que nenhum dos $\mathbf{x}_i$ possa ser escrito como combinação linear dos restantes, designa-se por conjunto de vectores **linearmente independente**.

Um conjunto de vectores $\mathbf{x}_i$ que permite a escrita única de cada elemento de $\mathcal{H}$ como uma combinação linear designa-se por **base** de $\mathcal{H}$. Para que a unicidade se mantenha é necessário que que os vectores sejam linearmente independentes. Todas as bases de um espaço vectorial $\mathcal{H}$ têm o mesmo número de elementos, designado por **dimensão** de $\mathcal{H}$.

O exemplo padrão de um espaço vectorial de dimensão finita é $\mathbb{R}^N$, o espaço dos vectores coluna $([\mathbf{x}]_1, \ldots, [\mathbf{x}]_N)^\top$. Em $\mathbb{R}^N$, a adição e multiplicação escalar são definidas elemento a elemento e a sua base canónica é $\{\mathbf{e}_1, \ldots, \mathbf{e}_N\}$ onde, para $j = 1, \ldots, N$, $[\mathbf{e}_j]_i = \delta_{ij}$ sendo δ_{ij} o símbolo de Kronecker definido como

$$\delta_{ij} = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{caso contrário.} \end{cases} \tag{D.12}$$

Definição D.2 (Transformação linear)**.** *Dados dois espaços vectoriais reais, $\mathcal{H}_1$ e $\mathcal{H}_2$, uma **transformação linear** é uma transformação*

$$L : \mathcal{H}_1 \to \mathcal{H}_2 \tag{D.13}$$

que, para todo o $\mathbf{x}$, $\mathbf{x}' \in \mathcal{H}_1$ e $\lambda, \lambda' \in \mathbb{R}$, satisfaz

$$L(\lambda \mathbf{x} + \lambda' \mathbf{x}') = \lambda L(\mathbf{x}) + \lambda' L(\mathbf{x}'). \tag{D.14}$$

[1] Do inglês, *span*.

Nas transformações lineares é costume omitir os parêntesis. Assim, normalmente, escreve-se $L\mathbf{x}$ *em vez de* $L(\mathbf{x})$.

Por questões de simplicidade, os conceitos seguintes são apresentados utilizando o caso em que $\mathcal{H}_1$ e $\mathcal{H}_2$ são idênticos e designados por $\mathcal{H}$ e têm dimensão N.

Uma transformação linear L é completamente determinada pelos seus valores numa base de $\mathcal{H}$. Se descrevermos um exemplo arbitrário como combinação linear dos vectores da base $\mathbf{e}_j$, e aplicarmos L obtém-se (utilizando (D.14))

$$L\sum_{j=1}^{N}\lambda_j\mathbf{e}_j=\sum_{j=1}^{N}\lambda_j L\mathbf{e}_j. \tag{D.15}$$

A imagem de cada vector base $L\mathbf{e}_j$ é, por sua vez, completamente determinada pelos seus coeficientes de expansão $A_{ij}, i=1,\ldots,N$

$$L\mathbf{e}_j=\sum_{i=1}^{N}A_{ij}\mathbf{e}_i. \tag{D.16}$$

Os coeficientes (A_{ij}) formam a matriz A de L com respeito à base $\{\mathbf{e}_1,\ldots,\mathbf{e}_N\}$. É comum pensar nas transformações lineares como matrizes e utilizar o mesmo símbolo para as referir. Denota-se por $\mathbf{1}$ a **matriz identidade** (ou unidade).

O **produto de matrizes** corresponde à composição de duas transformações lineares

$$(AB)_{ij}=\sum_{n=1}^{N}A_{in}B_{nj}. \tag{D.17}$$

A **transposta** é definida como $(A^\top)_{ij}:=A_{ji}$. As matrizes A tal que $A=A^\top$ chamam-se **simétricas**.

A **inversa** da matriz A escreve-se A^{-1} e satisfaz $AA^{-1}=A^{-1}A=\mathbf{1}$. A **pseudo-inversa** $A^\dagger$ satisfaz $AA^\dagger A=A$. Enquanto todas as matrizes têm pseudo-inversa, nem todas têm inversa. Aquelas que têm chamam-se **invertíveis** ou **não-singulares** e a sua inversa coincide com a pseudo-inversa.

D.2 Normas e produtos internos

Definição D.3 (Norma)**.** *Uma função* $\|\cdot\|:\mathcal{H}\to\mathbb{R}_0^+$ *designa-se* ***norma*** *em* $\mathcal{H}$ *se, para todo o* $\mathbf{x},\mathbf{x}'\in\mathcal{H}$ *e* $\lambda\in\mathbb{R}$ *satisfaz*

$$\begin{aligned} \|\mathbf{x}+\mathbf{x}'\| &\leq \|\mathbf{x}\|+\|\mathbf{x}'\|, && \text{(D.18)}\\ \|\lambda\mathbf{x}\| &= |\lambda|\|\mathbf{x}\|, && \text{(D.19)}\\ \|\mathbf{x}\| &> 0 \ \textit{se}\ \mathbf{x}\neq 0. && \text{(D.20)} \end{aligned}$$

Ao substituir, na última desigualdade, o "$>$" por "$\geq$", obtém-se uma ***semi-norma****.*

Qualquer norma define uma **métrica** d através de

$$d(\mathbf{x},\mathbf{x}') := \|\mathbf{x}-\mathbf{x}'\|. \tag{D.21}$$

De igual forma, uma semi-norma define uma **semi-métrica**. A (semi-)métrica herda certas propriedades da (semi-)norma, em particular a da desigualdade triangular e a da positividade.

Enquanto a norma dá origem a uma métrica, o contrário não é verdade. Neste sentido, o conceito de norma é mais forte que o de métrica. De igual forma, todo o produto interno (introduzido a seguir) dá origem a uma norma, mas não o contrário.

Definição D.4 (Forma bilinear)**.** *Uma* ***forma bilinear*** *num espaço vectorial $\mathcal{H}$ é uma função*

$$\begin{aligned} Q:\mathcal{H}\times\mathcal{H} &\to \mathbb{R}\\ (\mathbf{x},\mathbf{x}') &\to Q(\mathbf{x},\mathbf{x}') \end{aligned} \tag{D.22}$$

onde, para todo o $\mathbf{x},\mathbf{x}',\mathbf{x}''\in\mathcal{H}$ e todo o $\lambda,\lambda'\in\mathbb{R}$, verifica-se que

$$\begin{aligned} Q((\lambda\mathbf{x}+\lambda'\mathbf{x}'),\mathbf{x}'') &= \lambda Q(\mathbf{x},\mathbf{x}'')+\lambda' Q(\mathbf{x}',\mathbf{x}''), && \text{(D.23)}\\ Q(\mathbf{x}'',(\lambda\mathbf{x}+\lambda'\mathbf{x}')) &= \lambda Q(\mathbf{x}'',\mathbf{x})+\lambda' Q(\mathbf{x}'',\mathbf{x}'). && \text{(D.24)} \end{aligned}$$

A forma bilinear é simétrica se, para todo o $\mathbf{x},\mathbf{x}'\in\mathcal{H}$, satisfizer também

$$Q(\mathbf{x},\mathbf{x}') = Q(\mathbf{x}',\mathbf{x}) \tag{D.25}$$

Definição D.5 (Produto interno)**.** *Um* ***produto interno*** *num espaço vectorial $\mathcal{H}$ é uma forma bilinear simétrica,*

$$\begin{aligned} \langle\cdot,\cdot\rangle:\mathcal{H}\times\mathcal{H} &\to \mathbb{R}\\ (\mathbf{x},\mathbf{x}') &\to \langle\mathbf{x},\mathbf{x}'\rangle, \end{aligned} \tag{D.26}$$

que com a característica de ser ***definida positiva****. Por outras palavras, tem a pro-*

priedade de, para todo o $\mathbf{x} \in \mathcal{H}$*,*

$$\langle \mathbf{x}, \mathbf{x} \rangle \geq 0 \text{ } \textit{com a igualdade apenas para } \mathbf{x} = 0. \tag{D.27}$$

Definição D.6 (Espaço normado e espaço com produto interno)**.** *Um* ***espaço normado*** *é um espaço vectorial dotado de uma norma; um* ***espaço com produto interno***[2] *é um espaço vectorial dotado de um produto interno.*

Qualquer produto interno define uma norma correspondente através de

$$\|\mathbf{x}\| := \sqrt{\langle \mathbf{x}, \mathbf{x}' \rangle}. \tag{D.28}$$

A desigualdade de Chauchy-Schwarz diz que, para todo o $\mathbf{x}, \mathbf{x}' \in \mathcal{H}$,

$$|\langle \mathbf{x}, \mathbf{x}' \rangle| \leq \|\mathbf{x}\| \|\mathbf{x}'\|, \tag{D.29}$$

com a igualdade a ocorrer apenas quando $\mathbf{x}$ e $\mathbf{x}'$ são linearmente dependentes. Nalguns casos, o lado esquerdo pode ser muito menor que o direito; um caso extremo ocorre quando $\mathbf{x}$ e $\mathbf{x}'$ são **ortogonais** e $\langle \mathbf{x}, \mathbf{x}' \rangle = 0$.

Uma das construções mais úteis em espaços com produto interno são as expansões de base ortonormada. Suponha-se que $\mathbf{e}_1, \ldots, \mathbf{e}_N$, com $N \in \mathbb{N}$, formam um conjunto ortonormado, ou seja, são mutuamente ortogonais e têm norma 1. Se também formarem uma base de $\mathcal{H}$, o conjunto designa-se por **base ortonormada**. Neste caso, qualquer $\mathbf{x} \in \mathcal{H}$ pode ser escrito como uma combinação linear

$$\mathbf{x} = \sum_{i=1}^{N} \langle \mathbf{x}, \mathbf{e}_j \rangle \mathbf{e}_j. \tag{D.30}$$

O exemplo padrão de um espaço com produto interno é, mais uma vez, $\mathbb{R}^N$. Normalmente utiliza-se o produto interno canónico

$$\langle \mathbf{x}, \mathbf{x}' \rangle := \sum_{i=1}^{N} [\mathbf{x}]_i [\mathbf{x}']_i = \mathbf{x}^\top \mathbf{x}', \tag{D.31}$$

e refere-se $\mathbb{R}^N$ como **espaço Euclidiano de dimensão** N. Utilizando este produto interno e a base canónica de $\mathbb{R}^N$, cada coeficiente $\langle \mathbf{x}, \mathbf{e}_j \rangle$ de (D.30) selecciona uma entrada do vector coluna $\mathbf{x}$ e, assim, $\mathbf{x} = \sum_{i=1}^{N} [\mathbf{x}]_j \mathbf{e}_j$.

A definição de produto interno, permite introduzir um conjunto de alguns factos úteis

[2] Também conhecido como espaço pré-Hilbert.

sobre matrizes:

- Com o produto interno canónico é fácil verificar que, para todo o $\mathbf{x}, \mathbf{x}' \in \mathcal{H}$,

$$\langle \mathbf{x}, A\mathbf{x}' \rangle = \left\langle A^\top \mathbf{x}, \mathbf{x}' \right\rangle . \tag{D.32}$$

- Devido a (D.32), as matrizes simétricas podem ser trocadas entre os dois argumentos do produto interno sem alterar o seu valor.

- As matrizes simétricas A que satisfazem

$$\langle \mathbf{x}, A\mathbf{x} \rangle \geq 0 \tag{D.33}$$

  para todo o $\mathbf{x} \in \mathcal{H}$, designam-se por matrizes **definidas positivas**.

- Uma matriz **unitária**[3] U é caracterizada pela sua inversa U^{-1} ser igual à sua transposta $U^\top$. Assim, as matrizes unitárias satisfazem

$$\langle U\mathbf{x}, U\mathbf{x}' \rangle = \left\langle U^\top U\mathbf{x}, \mathbf{x}' \right\rangle = \left\langle U^{-1}U\mathbf{x}, \mathbf{x}' \right\rangle = \langle \mathbf{x}, \mathbf{x}' \rangle \tag{D.34}$$

  para todo o $\mathbf{x}, \mathbf{x}' \in \mathcal{H}$. Por outras palavras, o produto interno canónico torna-se invariante.

- Suponha-se que A é uma transformação linear. Se existir uma base $\mathbf{v}_1, \ldots, \mathbf{v}_N$ de $\mathcal{H}$ tal que, para todo o $i = 1, \ldots, N$,

$$A\mathbf{v}_i = \lambda_i \mathbf{v}_i, \tag{D.35}$$

  com $\lambda_i \in \mathbb{R}$, então A pode ser **diagonalizada**: escrita na base $\mathbf{v}_1, \ldots, \mathbf{v}_N$, tem-se que $A_{ij} = 0$ para todo $i \neq j$ e $A_{ii} = \lambda_i$ para todo o i. Os coeficientes λ_i chamam-se **valores próprios** e os vectores $\mathbf{v}_i$, **vectores próprios** de A.

D.3 Espaços de Hilbert

A análise funcional combina os conceitos de álgebra linear e análise estudando, assim, as questões de convergência e continuidade. Uma discussão detalhada pode ser encontrada em [Kolmogorov and Fomin 1961].

Definição D.7 (Sequência de Cauchy)**.** *Uma sequência* $(\mathbf{x}_i)_i := (\mathbf{x}_i)_{i \in \mathbb{N}} = (\mathbf{x}_1, \mathbf{x}_2, \ldots)$ *num espaço normado* $\mathcal{H}$ *é uma* ***sequência de Cauchy*** *se, para todo o* $\epsilon > 0$*, existe um* $n \in \mathbb{N}$ *tal que para todo o* $n', n'' > n$*, tem-se que* $\|\mathbf{x}_{n'} - \mathbf{x}_{n''}\| < \epsilon$*.*

[3] Também designada por matriz ortogonal.

Uma sequência de Cauchy ***converge*** *para um ponto* $\mathbf{x} \in \mathcal{H}$ *se* $\|\mathbf{x}_n - \mathbf{x}\| \to 0$ *quando* $n \to \infty$.

Definição D.8 (Completude, espaço de Banach, espaço de Hilbert)**.** *Um espaço* $\mathcal{H}$ *é* ***completo*** *se todas as sequências de Cauchy no espaço convergem.*

Um ***espaço de Banach*** *é um espaço normado completo; um* ***espaço de Hilbert*** *é um espaço com produto interno completo.*

O exemplo mais simples de um espaço de Hilbert é, mais uma vez, $\mathbb{R}^N$. Outros espaços Hilbert mais interessantes têm dimensão **infinita**. Neste caso, de uma maneira geral, assume-se que os espaços de Hilbert utilizados são **separáveis**, o que significa que existe um subconjunto denso contável. Um **subconjunto denso** é um conjunto S tal que cada elemento de $\mathcal{H}$ é o limite de uma sequência em S. De forma equivalente, num subconjunto denso a realização de S é igual a $\mathcal{H}$, onde a **realização** $\bar{S}$ é obtida adicionando todos os pontos limite das sequências de Cauchy ao conjunto.

A utilização de espaços de Hilbert separáveis permite trabalhar com único espaço de Hilbert já que, em certo sentido, todos são equivalentes. Por outro lado, qualquer subespaço fechado de um espaço de Hilbert é, por sua vez, um espaço de Hilbert.

Exemplo D.9 (Espaço de Hilbert de Funções)**.** Para $f, g \in C[a, b]$, onde $C[a, b]$ denota as funções reais contínuas no intervalo $[a, b]$,

$$\langle f, g \rangle := \int_a^b f(x)g(x)dx \tag{D.36}$$

define um produto interno. A realização de $C[a, b]$ na norma correspondente é o espaço de Hilbert $L_2[a, b]$ de funções mensuráveis quadradas integráveis

$$\int_a^b f(x)^2 dx < \infty. \tag{D.37}$$

Anexo E

Etiquetas do PALAVRAS

Neste anexo apresentam-se as etiquetas que distinguem as classes gramaticais das palavras e suas flexões e os grupos funcionais e constituintes sintácticos da oração. Em `http://visl.sdu.dk/visl/pt/info/symbolset-manual.html` podem ser consultadas outras etiquetas.

Tabela E.1: PALAVRAS – As etiquetas das classes gramaticais

Etiqueta	Classe gramatical
n	nome
prop	nome próprio
adj	adjectivo
v-fin	verbo finito
v-inf	infinitivo
v-pcp	particípio
v-ger	gerúndio
art	artigo
pron-pers	pronome pessoal
pron-det	pronome determinativo (demonstrativo, indefinido, relativo, interrogativo, possessivo)
pron-indp	pronome independente (demonstrativo, indefinido, relativo, interrogativo)
adv	advérbio
num	numeral
prp	preposição
intj	interjeição
conj-s	conjunção subordinativa
conj-c	conjunção coordenativa

Tabela E.2: PALAVRAS – As etiquetas das flexões

Flexão	Etiqueta	Aplicável a...
Género	M, F, M/F	n, prop, adj, v-pcp, art, pron
Número	S, P, S/P	n, prop, adj, v-pcp, art, pron
Pessoa	1, 2, 3, 0	pron-pers, v-fin, v-inf
Tempo	PR, IMPF, PS, FUT	v-fin
Modo	IND, SUBJ	v-fin, v-inf
	IMP	v-fin

M: masculino, F: feminino
S: singular, P: plural
1: 1ª pessoa, 2: 2ª pessoa, 3: 3ª pessoa
PR: presente, IMPF: imperfeito, PS: perfeito simples, FUT: futuro
IND: indicativo, SUBJ: conjuntivo, IMP: imperativo

Tabela E.3: PALAVRAS – As etiquetas dos constituintes sintácticos

Etiqueta	Categoria	Núcleo
np	sintagma nominal	nome
propp		nome próprio
pronp		pronome
adjp	sintagma adposicional	adjectivo
advp		advérbio
detp		determinante
vp	sintagma verbal	verbo principal ou auxiliar
pp	sintagma preposicional	preposição

Tabela E.4: PALAVRAS – As etiquetas das funções sintácticas

Etiqueta	Categoria
SUBJ	sujeito
ACC	complemento directo (acusativo)
DAT	complemento indirecto (pronomial)
PIV	complemento preposicional
ADVS, ADVO	complemento adverbial
SC	predicativo do sujeito
OC	predicativo do objecto
P	predicado
FMV, IMV	verbo principal
FAUX, IAUX	verbo auxiliar

Tabela E.5: PALAVRAS – As etiquetas dos tipos de frase

Etiqueta	Tipo
STA	frase declarativa
QUE	frase interrogativa
EXC	frase exclamativa
COM	frase imperativa

Tabela E.6: PALAVRAS – As etiquetas dos tipos de orações

Etiqueta	Tipo
fcl	oração finita
icl	oração infinita
acl	oração averbal
cu	paratagma

Notação e símbolos

$\mathbb{R}$	conjunto dos reais
$\mathbb{N}$	conjunto dos números naturais, $\mathbb{N} = \{1, 2, \ldots\}$
$\mathcal{X}$	domínio de entrada
N	dimensão de $\mathcal{X}$ (utilizado se $\mathcal{X}$ for um espaço vectorial)
x_i	exemplo de entrada
y_i	valor objectivo $y_i \in \mathbb{R}$ ou classe $y_i \in \{\pm 1\}$
m	número de exemplos de treino
$[m]$	notação compacta para $\{1, \ldots, m\}$
i, j	índices, por omissão percorrem $[m]$
X	uma amostra de exemplos de entrada, $X = (x_1, \ldots, x_m)$
Y	uma amostra de valores objectivo, $Y = (y_1, \ldots, y_m)$
$\mathcal{H}$	espaço de características
Φ	mapa de características, $\Phi : \mathcal{X} \to \mathcal{H}$
$\mathbf{x}_i$	vector com entradas $[\mathbf{x}_i]_j$; normalmente um exemplo embutido em $\mathcal{H}$, $\mathbf{x}_i = \Phi(x_i)$
$\mathbf{w}$	vector de pesos no espaço de características
b	valor de compensação[1] (ou valor de limiar)
κ	núcleo (definido positivo)
K	matriz de núcleo ou matriz de Gram, $K_{ij} = \kappa(x_i, x_j)$
$\mathrm{P}\{\cdot\}$	probabilidade de uma fórmula lógica
$\mathrm{P}(C)$	probabilidade de um conjunto C
$p(x)$	densidade de probabilidade em $x \in \mathcal{X}$
$\mathcal{N}(\mu, \sigma)$	distribuição normal com média μ e variância σ
α_i	multiplicador de Lagrange ou coeficiente de expansão
β_i	multiplicador de Lagrange
$\boldsymbol{\alpha}$, $\boldsymbol{\beta}$	vectores de multiplicadores de Lagrange
ξ_i	variáveis permissivas
$\boldsymbol{\xi}$	vector de todas as variáveis permissivas
$\langle \mathbf{x}, \mathbf{x}' \rangle$	produto interno entre $\mathbf{x}$ e $\mathbf{x}'$
$\| \cdot \|$	norma-2, $\|\mathbf{x}\| := \sqrt{\langle \mathbf{x}, \mathbf{x}' \rangle}$
ln	logaritmo na base e

$\log_2$	logaritmo na base 2
f	uma função $\mathcal{X} \to \mathbb{R}$ ou $\mathcal{X} \to \{\pm 1\}$
$\mathcal{F}$	família de funções
h	dimensão VC
$x \in [a, b]$	intervalo $a \leq x \leq b$
$x \in (a, b]$	intervalo $a < x \leq b$
$x \in (a, b)$	intervalo $a < x < b$
A^{-1}	matriz inversa (nalguns casos, pseudo-inversa
$A^\top$	matriz transposta (ou vector)
$(x_j)_j$ ou (x_j)	abreviatura para para a sequência $(x_j) = (x_1, x_2, \ldots)$
I_A	função característica (ou indicador) no conjunto A,
	i.e., $I_A(x) = 1$ se $x \in A$ e 0 caso contrário
$\mathbf{1}$	matriz unidade, ou mapa identidade ($\mathbf{1}(x) = x$ para todo o x)
$\lvert C \rvert$	cardinalidade do conjunto C
	(para conjuntos finitos, o número de elementos)
δ_{ij}	δ de Kronecker

Referências

R. Aires, A. Manfrin, S. Aluísio, and D. Santos. What is my style? stylistic features in portuguese web pages according to IR users' needs. In *LREC-04, 4th International Conference on Language resources and Evaluation*, pages 1943–1946, Lisbon, PT, May 2004.

J. Allen. *Natural Language Understanding*. Benjamin Cummings, 2nd edition, 1994.

E. Allwein, R. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. In *ICML-00, 17th International Conference on Machine Learning*, pages 9–16. Morgan Kaufmann, San Francisco, CA, 2000.

I. Androutsopoulos, J. Koutsias, K. Chandrinos, and C. Spyropoulos. An experimental comparison of naïve Bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval*, pages 160–167, Athens, GR, 2000.

C. Apté, J. Damerau, and S. Weiss. Automated learning of decision rules for text categorization. *ACM Transactions on Information Systems*, 12(3):233–251, 1994.

G. Attardi, D. Di Marco, and D. Salvi. Categorization by context. *Journal of Universal Computer Science*, 4(9):719–736, 1998.

J. Backus. The syntax and semantics of the proposed international algebraic of the Zurich ACM-GAMM Conference. In *Proceedings of the International Conference on Information Processing – IFIP Congress*, pages 125–132. UNESCO, Paris, 1959.

L. Baker and A. McCallum. Distributional clustering of words for text classification. In *SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 96–103, Melbourne, AU, 1998.

R. Basili, A. Moschitti, and M. Pazienza. A text classifier based on linguistic processing. In *IJCAI-99, Workshop on Machine Learning for Information Filtering*, 1999.

R. Basili, A. Moschitti, and M. Pazienza. Complex linguistic features for text classification: A comprehensive study. In *ECIR-04, 26th European Conference on Information Retrieval Research*, pages 181–196, Sunderland, UK, 2004.

R. Basili, M. Cammisa, and A. Moschitti. A semantic kernel to exploit linguistic knowledge. In *AI*IA-05, 9th Conference of the Italian Association for Artificial Intelligence*, pages 290–302, Milan, IT, 2005.

N. Belkin and W. Croft. Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.

E. Bick. *The Parsing System PALAVRAS – Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework*. Aarhus University Press, 2000. ISBN 87-7288-910-1.

E. Bick. A constraint grammar based question answering system for portuguese. In *Proceedings of the 11th Portuguese Conference of Artificial Intelligence – EPIA'03*, pages 414–418. LNAI Springer Verlag, 2003.

D. Blair. Information retrieval and the philosophy of language. *The Computer Journal*, 35 (3):200–207, 1992.

H. Borko and M. Bernick. Automatic document classification. *Journal of the Association for Computing Machinery*, 10(2):151–161, 1963.

B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *WCLT-92, 5th ACM Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, US, 1992. ACM Press.

R. Brachman and J. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, 1985.

Y. Camargo. Abordagem linguística na classificação de textos em Português. Master's thesis, Universidade Federal do Rio de Janeiro, Rio de Janeiro, BR, Junho 2007.

M. Caropeso, S. Matwin, and F. Sebastiani. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In A. Chin, editor, *Text Databases and Document Management: Theory and Practice*, pages 78–102. Idea Group Publishing, Hershey, US, 2001.

W. Cavnar and J. Trenkle. N-gram based text categorization. In *SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Las Vegas, US, 1994.

S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *SIGMOD-98, ACM International Conference on Management of Data*, pages 307–318, Seattle, US, 1998.

N. Chomsky. Three models for the description of language. *IRI Transactions on Information Theory*, 2(3):113–124, 1956.

C. Clack, J. Farringdon, P. Lidwell, and T. Yu. Autonomous document classification for business. In *1st International Conference on Autonomous Agents*, pages 201–208, Marina del Ray, US, 1997.

W. Cohen. Learning to classify english text with ILP methods. In L. D. Raedt, editor, *Advances in inductive logic programming*, pages 124–143. IOS Press, Amsterdam, NL, 1995.

W. Cohen. Learning rules that classify e-mail. In *AAAI-96, Spring Symposium on Machine Learning in Information Access*, pages 18–25, Stanford, US, 1996.

W. Cohen and H. Hirsh. Joins that generalize: text classification using Wirl. In *KDD-98, 4th International Conference on Knowledge Discovery and Data Mining*, pages 169–173, New York, US, 1998.

W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems*, 17(2):141–173, 1999.

M. Collins and N. Duffy. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *ACL-02, 30th Annual Meeting of the Association for Computational Linguistics*, pages 263–270, 2002.

C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. *Machine Learning*, 47(2):201–233, 2002.

R. Creecy, B. Masand, S. Smith, and D. Waltz. Trading MIPS and memory for knowledge engineering: classifying census returns on the Connection Machine. *Communications of the ACM*, 35(8):48–63, 1992.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

N. Cristianini, H. Lodhi, and J. Shawe-Taylor. Latent semantic kernels. *Journal of Intelligent Information Systems*, 18:127–152, 2002.

W. Croft and D. Lewis. Term clustering of syntactic phrases. In J. Vidick, editor, *SIGIR-90, 13th ACM International Conference on Research and Development in Information Retrieval*, pages 385–404, Brussels, BE, 1990.

C. Cunha and L. Cintra. *Nova Gramática do Português Contemporâneo*. Edições João Sá da Costa, 1984. ISBN 972-9230-00-5.

I. Dagan, Y. Karov, and D. Roth. Mistake-driven learning in text categorization. In *EMNLP-97, 2nd Conference on Empirical Methods in Natural Language Processing*, pages 55–63, Providence, US, 1997.

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic indexing. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

T. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

H. Drucker, V. Vapnik, and D. Wu. Automatic text categorization and its applications to text retrieval. *IEEE Transactions on Neural Networks*, 10(5):1048–1054, 1999.

S. Dumais and H. Chen. Hierarchical classification of web content. In *SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval*, pages 256–263, Athens, GR, 2000.

S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *CIKM-98, 7th ACM International Conference on Information and Knowledge Management*, pages 148–155, Bethesda, US, 1998.

J. Fagan. *Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparison of Syntactic and non-Syntactic Methods*. PhD thesis, Department of Computer Science, Cornell University, US, 1987.

B. Field. Towards automatic indexing: automatic assignment of controlled-language indexing and classification from free indexing. *Journal of Documentation*, 31(4):246–265, 1975.

R. Forsyth. New directions in text classification. In A. Gammerman, editor, *Causal models and intelligent data management*, pages 151–185. Springer, Heidelberg, DE, 1999.

J. Friedman. Another approach to polychotomous classification. Technical report, Stanford University, Stanford, US, 1996.

N. Fuhr. Models for retrieval with probabilistic indexing. *Information Processing and Management*, 25(1):55–72, 1989.

N. Fuhr and G. Knorz. Retrieval test evaluation of a rule-based automated indexing (AIR/PHYS). In *SIGIR-84, 7th ACM International Conference on Research and Development in Information Retrieval*, pages 391–408, Cambridge, UK, 1984.

N. Fuhr and U. Pfeifer. Probabilistic information retrieval as combination of abstraction inductive learning and probabilistic assumptions. *ACM Transactions on Information Systems*, 12(1):92–115, 1994.

N. Fuhr, S. Hartmann, G. Knorz, D. Lustig, M. Schwantner, and K. Tzeras. AIR/X – a rule-based multistage indexing system for large subject fields. In *RIAO'91, 3rd International Conference Recherche d'Information Assistee par Ordinateur*, pages 606–623, Barcelona, SP, 1991. Elsevier Science Publishers.

J. Fürnkranz. Round robin classification. *Journal of Machine Learning Research*, 2:721–747, 2002.

J. Fürnkranz, T. Mitchell, and E. Riloff. A case study in using linguistic phrases for text categorization on the WWW. In M. Sahami, M. Craven, T. Joachims, and A. McCallum, editors, *AAAI/ICML Workshop on Learning for Text Categorization*. AAAI Press, 1998.

L. Galavotti, F. Sebastiani, and M. Simi. Experiments on the use of feature selection and negative evidence in automated text categorization. In *ECDL-00, 4th European Conference on Research and Advanced Technology for Digital Libraries*, pages 59–68, Lisbon, PT, 2000.

W. Gale, K. Church, and D. Yarowsky. A method for disambiguating word senses in a large corpus. *Computers and Humanities*, 26(5):415–439, 1993.

T. Gonçalves and P. Quaresma. A preliminary approach to the multilabel classification problem of Portuguese juridical documents. In F. Moura-Pires and S. Abreu, editors, *EPIA-03, 11th Portuguese Conference on Artificial Intelligence*, LNAI 2902, pages 435–444, Évora, PT, December 2003. Springer-Verlag.

T. Gonçalves and P. Quaresma. Enhancing a Portuguese text classifier using part-of-speech tags. In M. Klopotek, S. Weirzchon, and K. Trojanowski, editors, *IIPWM-05, Intelligent Information Processing and Web Mining*, Advances in Soft Computing, pages 189–198, Gdansk, PL, June 2005a. Springer-Verlag.

T. Gonçalves and P. Quaresma. Is linguistic information relevant for the classification of legal texts? In G. Sartor, editor, *ICAIL-05, 10th International Conference on Artificial Intelligence and Law*, pages 168–176, Bologna, IT, June 2005b. ACM. ISBN 1-59593-081-7.

T. Gonçalves, C. Silva, P. Quaresma, and R. Vieira. Analysing part-of-speech for Portuguese text classification. In *CICLing-06, 7th international Conference on Intelligent Text Processing and Computational Linguistics*, pages 551–562, Mexico City, MX, February 2006. Springer-Verlag.

W. Gray and A. Harley. Computer-assisted indexing. *Information Storage and Retrieval*, 7 (4):167–174, 1971.

L. Guthrie, E. Walker, and J. Guthrie. Document classification by machine: theory and practice. In *COLING-94, 15th International Conference on Computational Linguistics*, pages 1059–1063, Kyoto, JP, 1994.

I. Guyon, B. Boser, and V. Vapnik. Automatic capacity tuning of very large VC-dimension classifiers. In S. Hanson, J. Cowan, and C. Giles, editors, *Advances in Neural information Processing Systems*, volume 5, pages 147–155. Morgan Kaufmann, San Mateo, CA, 1993.

T. Gärtner. Exponential and geometric kernels for graphs. In *NIPS-02, 16th Neural Information Processing Systems – Workshop on Unreal Data: Principles of Modeling Nonvectorial Data*, 2002.

T. Gärtner, P. Flach, and S. Wrobel. On graph kernels: Hardness results and efficient alternatives. In B. Schölkopf and M. Warmuth, editors, *COLT-03, 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop*, LNAI 2777, pages 129–143. Springer-Verlag Berlin Heidelberg, 2003.

N. Gövert, M. Lalmas, and N. Fuhr. A probabilistic description-oriented approach for categorising web documents. In *CIKM-99, 8thACM International Conference on Information and Knowledge Management*, pages 475–482, Kansas City, US, 1999.

K. Hamill and A. Zamora. The use of titles for automatic document classification. *Journal of the American Society for Information Science*, 33(6):396–402, 1980.

T. Hastie and R. Tibshirani. Classification by pairwise coupling. *Advances in Neural Information Processing Systems*, 1998.

D. Haussler. Convolution kernels on discrete structures. Technical report, Department of Computer Science, University of California at Santa Cruz, 1999.

P. Hayes and P. Andersen. TCS: a shell for content-based text categorization. In *CAIA-90, 6th IEEE Conference on Artificial Intelligent Applications*, pages 320–326, Santa Barbara, US, 1990.

H. Heaps. A theory of relevance for automatic document classification. *Information and Control*, 22(3):268–278, 1973.

M. Hearst. Noun homograph disambiguation using local context in large corpora. In *7th Annual Conference of the University of Waterloo Centre for the New Oxford English Dictionary*, pages 1–22, Oxford, UK, 1991.

W. Hersh, C. Buckley, T. leone, and D. Hickman. OHSUMED: an interactive retrieval evaluation and new large collection for research. In *SIGIR-94, 17th ACM International Conference on Research and Developement in Information Retrieval*, pages 192–201, Dublin, IE, 1994.

W. Hoyle. Automatic indexing and generation of classification by algorithm. *Information Storage and Retrieval*, 9(4):233–242, 1973.

S. Huffman. Acquaintance: language-independent document categorization by n-grams. In *TREC-4, 4th Text Retrieval Conference*, pages 359–371, Gaithersburg, US, 1995.

D. Hull. Improving text retrieval for the routing problem using latent semantic indexing. In *SIGIR-94, 17th ACM International Conference on Research and Developement in Information Retrieval*, pages 282–289, Dublin, IE, 1994.

W. Imrich and S. Klavzar. *Product Graphs: Structure and Recognition*. John Wiley, 2000.

D. Ittner, D. Lewis, and D. Ahn. Text categorization of low quality images. In *SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 301–315, Las Vegas, US, 1995.

M. Iwayama and T. Tokunaga. Cluster-based text categorization: a comparison of category search strategies. In *SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, pages 273–281, Seattle, US, 1995.

T. Joachims. *Learning to Classify Text Using Support Vector Machines*. Kluwer Academic Publishers, 2002.

T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *ICML-97, 14th International Conference on Machine Learning*, pages 143–151, Nashville, US, 1997.

T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *ECML-98, 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, DE, 1998.

T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.

G. John and P. Langley. Estimating continuous distributions in Bayesian classifiers. In *UAI-95, 11th Conference on Uncertainty in Artificial Intelligence*, pages 338–345, Montreal, CA, 1995. Morgan Kaufmann.

D. Jurafsky and J. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 2000. ISBN 0-13-095069-6.

H. Kamp and U. Reyle. *From Discourse to Logic: An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, volume 42 of *Studies in Linguistics and Philosophy*. Springer, 1993. ISBN 079-2310-280.

F. Karlsonn, A. Voutilainen, J. Heikkilä, and A. Anttila. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Number 4 in Natural Language Processing. Mouton de Gruyter, Berlin and New York, 1995. ISBN 3-11-014179-5.

H. Kashima and A. Inokuchi. Kernels for graph classification. In *ICDM-02, IEEE International Conference on Data Mining – Workshop on Active Mining*, 2002.

B. Kessler, G. Nunberg, and H. Schütze. Automatic detection of text genre. In *ACL-97, 35th Annual Meeting of the Association for Computational Linguistics*, pages 32–38, Madrid, SP, 1997.

R. Klinkenberg and T. Joachims. Detecting concept drift with support vector machines. In *ICML-00, 17th International Conference on Machine Learning*, pages 487–494, Stanford, US, 2000.

D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *ICML-97, 14th International Conference on Machine Learning*, pages 170–178, Nashville, US, 1997. Morgan Kaufmann Publishers.

A. Kolmogorov and S. Fomin. *Functional Analysis*. Graylock Press, Albany, US, 1961.

W. Lam and C. Lo. Using a generalized instance set for automatic text categorization. In *SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 81–89, Melbourne, AU, 1998.

W. Lam, K. Low, and C. Ho. Using a Bayesian network induction approach for text categorization. In *IJCAI-97, 15th International Joint Conference on Artificial Intelligence*, pages 745–750, Nagoya, JP, 1997.

W. Lam, M. Ruiz, and P. Srinivasan. Automatic text categorization and its applications to text retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 11(6):865–879, 1999.

K. Lang. NewsWeeder: learning to filter netnews. In *ICML-95, 12th International Conference on Machine Learning*, pages 331–339, Lake Tahoe, US, 1995.

L. Langie. Um estudo sobre a aplicação do algoritmo kNN a categorização hierárquica de textos. Master's thesis, Faculdade de Informática, PUCRS, Porto Alegre, BR, 2004.

L. Larkey. Automatic essay grading using text categorization techniques. In *SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 90–95, Melbourne, AU, 1998.

L. Larkey. A patent search and classification system. In *4th ACM Conference on Digital Libraries*, pages 179–187, Berkeley, US, 1999.

L. Larkey and W. Croft. Combining classifiers in text categorization. In *SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pages 289–297, Zürich, CH, 1996.

Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines. In *33rd Symposium on the Interface*, 2001.

E. Leopold and J. Kinderman. Text categorization with support vector machines. How to represent texts in input space? *Machine Laerning: Special Issue on Support Vector Machines and Kernel Methods*, 46(1–3):423–444, 2002.

D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval*, pages 37–50, 1992a.

D. Lewis. *Representation and Learning in Information Retrieval*. PhD thesis, Department of Computer and Information Science, University of Massachusetts, US, 1992b.

D. Lewis. Naïve Bayes at forty: The independence assumption in information retrieval. In *ECML-98, 10th European Conference on Machine Learning*, pages 4–15, Chemnitz, DE, 1998.

D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *SIGIR-94, 17th ACM International Conference on Research and Development in Information Retrieval*, pages 3–12, Dublin, IE, 1994.

D. Lewis and K. S. Jones. Natural language processing for information retrieval. *Communications of the ACM*, 39(1):92–101, 1997. Reprinted in From Classification to 'Knowledge Organisation'. Dorking Revisited or 'Past is Prelude', (Ed. A. Gilchrist), FID 714, The Hague: Federation International de Documentation.

D. Lewis and M. Ringuette. A comparison of two learning algorithms for text categorization. In *SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 81–93, Las Vegas, US, 1994.

D. Lewis, R. Shapire, J. Callan, and R. Papka. Training algorithms for linear text classifiers. In *SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pages 298–306, Zürich, CH, 1996.

H. Li and K. Yamanishi. Text classification using ESC-based stochastic decision lists. In *CIKM-98, 7th ACM International Conference on Information and Knowledge Management*, pages 122–130, Kansas City, US, 1999.

Y. Li and A. Jain. Classification of text documents. *ACM Transactions on Information Systems*, 41(8):537–546, 1998.

R. Liere and P. Tadepalli. Active learning with commitees for text categorization. In *AAAI-97, 14th Conference of the American Associationfor Artificial Intelligence*, Providence, US, 1997.

H. Lodhi, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2001.

H. Lodhi, C. Saunders, J. Shawe-Taylor, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.

J. Lopes, N. Marques, and V. Rocio. Polaris: POrtuguese Lexicon Acquisition and Retrieval Interactive System. In *The Practical Applications of Prolog*, page 665. Royal Society of Arts, 1994.

C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, US, 1999.

M. Maron. Automatic indexing: an experimental inquiry. *Journal of the Association for Computing Machinery*, 8(3):404–417, 1961.

B. Masand. Optimising confidence of text classification by evolution of symbolic expressions. In K. Kinnear, editor, *Advances in genetic programming*, chapter 21. The MIT Press, Cambridge, US, 1994.

B. Masand, G. Linoff, and D. Waltz. Classifying news stories using memory-based reasoning. In *SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval*, pages 59–65, Kobenhavn, DK, 1992.

A. McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI-99 Workshop on Text Learning*, 1999.

A. McCallum, R. Rosenfeld, T. Mitchell, and A. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *ICML-98, 15th International Conference on Machine Learning*, pages 359–367, Madison, US, 1998.

T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

D. Mladenić. Feature subset selection in text learning. In *ECML-98, 10th European Conference on Machine Learning*, pages 95–100, Chemnitz, DE, 1998.

D. Mladenić and M. Grobelnik. Word sequences as features in text-learning. In *ERK-98, 7th Electrotechnical and Computer Science Conference*, pages 145–148, Ljubljana, SL, 1998.

S. Moraes and V. Lima. Um estudo sobre categorização hierárquica de uma grande colecção de textos em língua Portuguesa. In *TIL-07, 5º workshop em Tecnologia da Informação e da Linguagem Humana*, pages 1659–1668, Rio de Janeiro, BR, July 2007.

A. Moschitti. A study on optimal parameter tuning for Rocchio text classifier. In *ECIR-03, 25th European Conference on Information Retrieval Research*, pages 546–547, Pisa, IT, 2003a.

A. Moschitti. *Natural Language Processing and Text Categorization: a study on the reciprocal beneficial interactions*. PhD thesis, University of Rome Tor Vergata, Rome, IT, 2003b.

A. Moschitti. A study on convolution kernels for shallow semantic parsing. In *ACL-04, 42nd Annual Meeting on Association for Computational Linguistics*, pages 335–342, Barcelona, SP, 2004.

A. Moschitti. Efficient convolution kernels for dependency and constituent syntactic trees. In *ECML-06, 17th European Conference on Machine Learning*, pages 318–329, Berlin, DE, 2006a.

A. Moschitti. Making tree kernels practical for natural language learning. In *EACL-06, 11th International Conference on European Association for Computational Linguistics*, pages 113–120, Trento, IT, 2006b.

I. Moulinier and J. Ganascia. Applying an existing machine learning algorithm to text categorization. In S. Wermter, R. Riloff, and G. Scheler, editors, *Connectionist, statistical, and symbolic approaches to learning for natural language processing*, pages 343–354. Springer Verlag, Heibelberg, DE, 1996.

K. Myers, M. Kearns, S. Singh, and M. Walker. A boosting approach to topic spotting on subdialogues. In *ICML-00, 17th International Conference on Machine Learning*, Stanford, US, 2000.

G. Neumann and S. Schmeier. Combining sallow text processing and machine learning in real world applications. In T. Joachims, A. McCallum, M. Sahami, and L. Ungar, editors, *IJCAI-99, Workshop on Machine Learning for Information Filtering*, 1999.

H. Ng, W. Gow, and K. Low. Feature selection, perceptron learning and a usability case study for text categorization. In *SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval*, pages 67–73, Philadelphia, US, 1997.

K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labelled and unlabelled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

E. Oliveira, P. Ciarelli, W. Henrique, L. Veronese, F. Pedroni, and A. Souza. Intelligent classification of economic activities from free text descriptions. In *TIL-07, 5º workshop em Tecnologia da Informação e da Linguagem Humana*, pages 1635–1639, Rio de Janeiro, BR, July 2007.

G. Paass, E. Leopold, M. Larson, J. Kindermann, and S. Eickeler. SVM classification using sequences of phonemes and syllables. In T. Elomaa, H. Manilla, and H. Toivonen, editors, *PDMKD-02, 6th European Conference on Principles and od Data Mining and Knowledge Discovery*, pages 373–384. Springer-Verlag, 2002.

D. Pavelec, E. Justino, and C. Freitas. Identificação da autoria de documentos digitais com base em atributos estilométricos da Língua Portuguesa. In *TIL-06, 4º workshop em Tecnologia da Informação e da Linguagem Humana*, pages 1659–1668, Ribeirão Preto, BR, October 2006.

J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. Burges, , and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 185–208. MIT Press, 1999.

P. Quaresma, L. Quintano, I. Rodrigues, and P. Salgueiro. The University of Évora approach to QA@CLEF-2004. In C. Peters, P. Clough, J. Gonzalo, G. Jones, M. Kluck, and B. Magnini, editors, *Multilingual Information Access for Text, Speech and Images*, Lecture Notes in Computer Science LNCS 3491, pages 534–543. Springer, 2005. Revised and selected papers of the 5th Workshop of the Cross-Language Evaluation Forum, CLEF.

R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, US, 1993.

H. Ragas and C. Koster. Four text classification algorithms compared on a dutch corpus. In *SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 369–370, Melbourne, AU, 1998.

R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

S. Rigo, J. Oliveira, and C. Barbieri. Classificação de textos baseada em ontologias de domínio. In *TIL-07, 5º workshop em Tecnologia da Informação e da Linguagem Humana*, pages 1640–1649, Rio de Janeiro, BR, July 2007.

E. Riloff. Little words can make big difference for text classification. In *SIGIR-95, 18th ACM International Conference on Research and Development in Information Retrieval*, pages 130–136, Seattle, US, 1995.

E. Riloff and W. Lehnert. Information extraction as a basis for high-precision text classification. *ACM Transactions on Information Systems*, 12(3):296–333, 1994.

S. Robertson and P. Harding. Probabilistic automatic indexing by learning from human indexers. *Journal of Documentation*, 40(4):264–270, 1984.

S. Robertson and K. Spark Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

D. Roth. Learning to resolve natural language ambiguities: an unified approach. In *AAAI-98, 15th Conference of the American Association for Artificial Intelligence*, pages 806–813, Madison, US, 1998.

M. Ruiz and P. Srinivasan. Hierarchical neural networks for text categorization. In *SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*, pages 281–282, Berkeley, US, 1999.

C. Sable and V. Hatzivassiloglou. Text-based approaches for non-topical image categorization. *International Journal of Digital Libraries*, 3(3):261–275, 2000.

P. Salgueiro. Um sistema pergunta/resposta para a língua portuguesa. Master's thesis, Universidade de Évora, Dezembro 2004.

G. Salton, editor. *The SMART retrieval system: experiments in automatic document processing*. Prentice-Hall, Englewood Cliffs, NJ, 1971.

G. Salton. Developements in automatic text retrieval. *Science*, 253:974–979, 1991.

G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

G. Salton, A. Wang, and C. Yang. A vector space model for information retrieval. *Journal of the American Society for Information Retrieval*, 18:613–620, 1975.

M. Sasaki and K. Kita. Rule-based text categorization using hierarchical categories. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 2827–2830, La Jolla, US, 1998. IEEE Computer Society Press, Los Alamitos, US.

R. Schapire and Y. Singer. Boostexter: a boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

R. Schapire, Y. Singer, and A. Singhal. Boosting and Rocchio applied to text filtering. In *SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 215–223, Melbourne, AU, 1998.

B. Scholkopf, A. Smola, R. Williamson, and p.L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000.

H. Schütze. Automatic word sense discrimination. *Computational Linguistics*, 4(1):97–124, 1998.

H. Schütze, D. Hull, and J. Pedersen. A comparison of classifiers and document representations for the routing problem. In *SIGIR-95, 18th ACM International Conference on Research and Developement in Information Retrieval*, pages 229–237, Seattle, US, 1995.

B. Schölkopf and A. Smola. *Learning with kernels*. The MIT Press, 2002.

F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

F. Sebastiani, A. Sperduti, and N. Valdambrini. An improved boosting algorithm and its application to automated text categorization. In *CIKM-00, 9th ACM International Conference on Information and Knowledge Management*, pages 78–85, McLean, US, 2000.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

C. Silva and R. Vieira. Categorização de textos da língua Portuguesa com árvores de decisão, SVM e informações linguísticas. In *TIL-07, 5º workshop em Tecnologia da Informação e da Linguagem Humana*, pages 1650–1658, Rio de Janeiro, BR, July 2007.

C. Silva, R. Vieira, F. Osorio, and P. Quaresma. Mining linguistically interpreted texts. In *5th International Workshop on Linguistically Interpreted Corpora of COLING-04, Computational Linguistic International Conference*, Geneva, Switzerland, August 2004.

A. Singhal, m. Mitra, and c. Buckley. Learning routing queries in a query zone. In *SIGIR-97, 20th ACM International Conference on Research and Development in Information Retrieval*, pages 25–32, Philadelphia, US, 1997.

S. Soames. Presupposition. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume IV, pages 553–616. Reidel, 1984.

A. Sun and E. Lim. Hierarchical text classification and evaluation. In *ICDM-01, International Conference On Data Mining*, pages 521–528, California, US, 2001.

H. Taira and M. Haruno. Feature selection in SVM text categorization. In *AAAI-99, 16th Conference of the American Society for Artificial Intelligence*, pages 480–486, Orlando, US, 1999.

C. Tan, Y. Wang, and C. Lee. The use of bigrams to enhance text categorization. *Journal Information Processing and Management*, 30(4):529–546, 2002.

K. Tzeras and S. Hartmann. Automatic indexing based on Bayesian inference networks. In *SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*, pages 22–34, Pittsburg, US, 1993.

V. Vapnik. *The nature of statistical learning theory*. Springer, NY, 1995.

V. Vapnik and A. Chervonenkis. A note on one class of perceptrons. *Automation and Remote Control*, 25, 1964.

V. Vapnik and A. Lerner. Pettern recognition using generalized portrait method. *Automation and Remote Control*, 24:774–780, 1963.

S. Vishwanathan and A. Smola. Fast kernels on strings and trees. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 569–576. MIT Press, Cambridge, MA, 2003.

C. Watkins. Dynamic alignment kernels. Technical report, Department of Computer Science, Royal Holloway, University of London, 1999.

A. Weigend, E. Wiener, and J. Pederson. Exploiting hierarchy in text categorization. *Information Retrieval*, 1(3):193–216, 1999.

S. Weiss, C. Apté, F. Johnson, D. Oles, T. Goetz, and T. Hampp. Maximizing text-mining performance. *IEEE Intelligent Systems*, 14(4):63–69, 1999.

J. Weston and C. Watkins. Multi-class support vector machines. Technical report, Department of Computer Science, Royal Holloway, University of London, UK, 1998.

E. Wiener, J. Pedersen, and A. Weigend. A neural network approach to topic spotting. In *SDAIR-95, 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 317–332, Las Vegas, US, 1995.

I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, US, 2nd edition, 2005.

S. Wong, W. Ziarko, and P. Wong. Generalized vector space model in information retrieval. In *SIGIR-85, 8th ACM International Conference on Research and Developement in Information Retrieval*, pages 18–25, 1985.

Y. Yang. Noise reduction in a statistical approach to text categorization. In *SIGIR-95, 18th ACM International Conference on Research and Developement in Information Retrieval*, pages 256–263, Seattle, US, 1995.

Y. Yang and C. Chute. An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems*, 12(3):252–277, 1994.

Y. Yang and X. Liu. A re-examination of text categorization methods. In *SIGIR-99, 22nd ACM International Conference on Research and Developement in Information Retrieval*, pages 42–49, Berkeley, US, 1999.

Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

D. Zhang and W. Lee. Question classification using support vector machines. In *SIGIR-03, 26th ACM International Conference on Research and Developement in Information Retrieval*, pages 26–32, 2003.