UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

FERNANDO CHAGAS SANTOS

Variações do Método kNN e suas Aplicações na Classificação Automática de Textos

Goiânia
2009

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

AUTORIZAÇÃO PARA PUBLICAÇÃO DE DISSERTAÇÃO EM FORMATO ELETRÔNICO

Na qualidade de titular dos direitos de autor, **AUTORIZO** o Instituto de Informática da Universidade Federal de Goiás – UFG a reproduzir, inclusive em outro formato ou mídia e através de armazenamento permanente ou temporário, bem como a publicar na rede mundial de computadores (*Internet*) e na biblioteca virtual da UFG, entendendo-se os termos "reproduzir" e "publicar" conforme definições dos incisos VI e I, respectivamente, do artigo 5º da Lei nº 9610/98 de 10/02/1998, a obra abaixo especificada, sem que me seja devido pagamento a título de direitos autorais, desde que a reprodução e/ou publicação tenham a finalidade exclusiva de uso por quem a consulta, e a título de divulgação da produção acadêmica gerada pela Universidade, a partir desta data.

Título: Variações do Método kNN e suas Aplicações na Classificação Automática de Textos

Autor(a): Fernando Chagas Santos

Goiânia, 01 de Outubro de 2009.

Fernando Chagas Santos – Autor

Dr. Cedric Luiz de Carvalho – Orientador

Dr. Thierson Couto Rosa – Co-Orientador

FERNANDO CHAGAS SANTOS

Variações do Método kNN e suas Aplicações na Classificação Automática de Textos

Dissertação apresentada ao Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Sistemas de Informação.

Orientador: Prof. Dr. Cedric Luiz de Carvalho

Co-Orientador: Prof. Dr. Thierson Couto Rosa

Goiânia
2009

FERNANDO CHAGAS SANTOS

Variações do Método kNN e suas Aplicações na Classificação Automática de Textos

Dissertação defendida no Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Mestre em Ciência da Computação, aprovada em 01 de Outubro de 2009, pela Banca Examinadora constituída pelos professores:

Prof. Dr. Cedric Luiz de Carvalho
Instituto de Informática – UFG
Presidente da Banca

Prof. Dr. Thierson Couto Rosa
Instituto de Informática – UFG

Prof. Dr. Wellington Santos Martins
Instituto de Informática – UFG

Prof. Dr. Marcos André Gonçalves
Departamento de Ciência da Computação – DCC/UFMG

Agradecimentos

Em primeiro lugar, eu gostaria de agradecer a Deus, por me iluminar e sempre me amparar. Junto a Ele, agradeço aos meus pais, Nildo e Vânia, por me apoiarem em todas as minhas decisões e serem minha infraestrutura, sem os quais nada em minha vida seria possível. Ao meu avô José Goulart, a minha avó Tereza, aos meus tios, em especial as minhas tias Rosana e Lilian, e aos meus primos, em especial ao Higor, ao Deivid e à Raflézia. Aos meus amigos da UEG, em especial ao Vinicius, ao Thiago, ao José Olimpio e ao Alan. Aos meus amigos do LabTime, em especial à Adelaide, à Elvia, à Maria Dalva, à Maria Amélia e ao Rui. Aos meus amigos Marcelo e Everton, pelo companheirismo e amizade construída nesses últimos anos. Em especial, eu gostaria de agradecer ao meu amigo Junior (*in memorian*), um cara apaixonado pela aviação, exemplo de caráter, carisma e companheirismo. Deus leva os bons mais cedo! Espero ter aprendido muito com você Junão. Aos colegas André, Diego, Enio, Edir, Marcos, Jesmmer, Rommel, Wallid, Elisângela, Halley e aos professores Diane, Humberto, Ana Paula, João Carlos e Juliano do Instituto de Informática que se tornaram meus amigos. Em especial, à Luciana, o Lucas e o Rafael, que possibilitaram tornar esse período mágico. Sou muito grato ao Prof. Thierson, por ter me orientado na realização deste trabalho e disposto seu precioso tempo para me guiar e, muitas vezes, colocar a 'mão na massa' junto comigo. Sou também muito grato ao Prof. Cedric, por ter acreditado no meu potencial e me dado liberdade para a realização deste trabalho. Agradeço também à CAPES pelo suporte financeiro, fundamental para me subsidiar neste período. Enfim, agradeço a todos que auxiliaram de alguma maneira na realização deste trabalho.

Resumo

Santos, Fernando Chagas. **Variações do Método kNN e suas Aplicações na Classificação Automática de Textos**. Goiânia, 2009. 94p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

Grande parte das pesquisas relacionadas com a classificação automática de textos (CAT) tem procurado melhorar o desempenho (*eficácia* ou *eficiência*) do classificador responsável por classificar automaticamente um documento d, ainda não classificado. O método dos k vizinhos mais próximos (kNN, do inglês *k nearest neighbors*) é um dos métodos de classificação automática mais simples e eficazes já propostos. Neste trabalho foram propostas duas variações do método kNN, o kNN invertido (kINN) e o kNN simétrico (kSNN) com o objetivo de melhorar a eficácia da CAT. Os métodos kNN, kINN e kSNN foram aplicados nas coleções Reuters, 20NG e Ohsumed e os resultados obtidos demonstraram que os métodos kINN e kSNN tiveram eficácia superior ao método kNN ao serem aplicados nas coleções Reuters e Ohsumed e eficácia equivalente ao método kNN ao serem aplicados na coleção 20NG. Além disso, nessas coleções foi possível verificar que o desempenho obtido pelo método kNN é mais estável a variação do valor k do que os desempenhos obtidos pelos métodos kINN e kSNN. Um estudo paralelo foi realizado para gerar novas características em documentos a partir das matrizes de similaridade resultantes dos critérios de seleção dos melhores resultados obtidos na avaliação dos métodos kNN, kINN e kSNN. O método SVM, considerado um método de classificação do estado da arte em relação à eficácia, foi aplicado nas coleções Reuters, 20NG e Ohsumed - antes e após aplicar a abordagem de geração de características nesses documentos e os resultados obtidos demonstraram ganhos estatisticamente significativos em relação à coleção original.

Palavras–chave

Classificação de Textos, Aprendizagem de Máquina, Método kNN, Critérios de Seleção, Geração de Características, Geração de Termos

Abstract

Santos, Fernando Chagas. **kNN Method Variations and its applications in Text Classification**. Goiânia, 2009. 94p. MSc. Dissertation. Instituto de Informática, Universidade Federal de Goiás.

Most research on Automatic Text Categorization (ATC) seeks to improve the classifier performance (effective or efficient) responsible for automatically classifying a document d not yet rated. The k nearest neighbors (kNN) is simpler and it's one of automatic classification methods more effective as proposed. In this paper we proposed two kNN variations, Inverse kNN (kINN) and Symmetric kNN (kSNN) with the aim of improving the effectiveness of ACT. The kNN, kINN and kSNN methods were applied in Reuters, 20ng and Ohsumed collections and the results showed that kINN and kSNN methods were more effective than kNN method in Reuters and Ohsumed collections. kINN and kSNN methods were as effective as kNN method in 20NG collection. In addition, the performance achieved by kNN method is more stable than kINN and kSNN methods when the value k change. A parallel study was conducted to generate new features in documents from the similarity matrices resulting from the selection criteria for the best results obtained in kNN, kINN and kSNN methods. The SVM (considered a state of the art method) was applied in Reuters, 20NG and Ohsumed collections - before and after applying this approach to generate features in these documents and the results showed statistically significant gains for the original collection.

Keywords

Text Classification, Machine Learning, kNN Method, Feature Selection, Feature Construction

Sumário

Lista de Figuras **8**

Lista de Tabelas **9**

1 Introdução **11**
- 1.1 Contextualização 11
- 1.2 Problemas e Objetivos 13
  - 1.2.1 Variações do método kNN 13
  - 1.2.2 Geração de características 15
- 1.3 Principais contribuições do trabalho 17
- 1.4 Organização do trabalho 17

2 Trabalhos Relacionados **19**
- 2.1 Critérios de seleção para o método kNN 19
- 2.2 Representação dos documentos 20
  - 2.2.1 Geração de características em textos 21
  - 2.2.2 Outras formas de gerar características 23

3 Conceitos Relacionados **24**
- 3.1 Preparação de documentos 24
  - 3.1.1 Representação de documentos 25
  - 3.1.2 Medidas da importância dos termos 26
  - 3.1.3 Medidas de similaridade entre documentos 27
- 3.2 Dimensionalidade de documentos 28
  - 3.2.1 Filtragem 30
  - 3.2.2 Conflação 30
  - 3.2.3 Seleção de características 30
    - Ganho de Informação 33
- 3.3 Classificação de documentos 35
  - 3.3.1 Classificação automática de documentos 36
  - 3.3.2 Formas de classificação 37
  - 3.3.3 k-vizinhos mais próximos 37
  - 3.3.4 Máquinas de vetores suporte 40
    - Fundamentação teórica 40
    - Dimensão VC e minimização do risco estrutural 42
    - SVMs lineares 43
  - 3.3.5 Avaliação dos classificadores 44

4 Abordagens Propostas **48**
4.1 Variações do método kNN 48
4.1.1 Método kINN 50
4.1.2 Método kSNN 52
4.2 Geração de Características 53

5 Metodologia Experimental **57**
5.1 Visão geral da metodologia 57
5.2 Coleção de documentos 58
5.2.1 Reuters 58
5.2.2 20 Newsgroups 59
5.2.3 Ohsumed 60
5.3 Preparação dos documentos 61
5.4 Método de avaliação 63
5.5 Métodos de classificação 64
5.5.1 Métodos kNN, kINN e kSNN 64
5.5.2 Método SVM 64

6 Resultados Experimentais **65**
6.1 Variações do método kNN 65
6.1.1 Análise da variação do valor de k 71
6.2 Geração de Características 76
6.2.1 Análise dos melhores termos 78

7 Conclusão e Trabalhos Futuros **80**
7.1 Trabalhos Futuros 82

Referências Bibliográficas **83**

Lista de Figuras

3.1 Cosseno θ entre os documentos d_1 e d_2. (Adaptado de [80]) 28
3.2 Espaço de busca de um conjunto com quatro características [19] 31
3.3 Documentos de treino mais próximos do documento de teste d_i 40
3.4 Possíveis separações de três pontos por uma reta [21] 42
3.5 Hiperplano separador com maior margem de separação entre duas categorias distintas 43

4.1 Critérios de seleção kNN, kINN e kRNN com o valor de $k = 3$ 50
4.2 Distribuição dos pontos da coleção P no espaço euclidiano R^2 52
4.3 Distribuição dos pontos da coleção P no espaço euclidiano R^2 53

5.1 Esquema do método adotado nos experimentos 57
5.2 Distribuição de documentos da coleção Reuters-21578 R8 (categoria x quant. de documentos) 59
5.3 Distribuição de documentos da coleção 20 *Newsgroups* (categoria x quant. de documentos) 60
5.4 Distribuição de documentos da coleção Ohsumed-18302 (categoria x quant. de documentos) 61
5.5 Ganho de informação no conjunto de treino da coleção Reuters-AT 63
5.6 Valores obtidos em $microF_1$ ao aplicar os métodos kNN, kINN e kSNN nas subcoleções de documentos da coleção Reuters-AT 63

6.1 Valores obtidos em $macroF_1$ na coleção Reuters-NS 73
6.2 Valores obtidos em $microF_1$ na coleção Reuters-NS 73
6.3 Valores obtidos em $macroF_1$ na coleção 20NG-AT 74
6.4 Valores obtidos em $microF_1$ na coleção 20NG-AT 74
6.5 Valores obtidos em $macroF_1$ na coleção Ohsumed-ST 75
6.6 Valores obtidos em $microF_1$ na coleção Ohsumed-ST 75

Lista de Tabelas

3.1 *Matriz documento-termo com a frequência absoluta de ocorrência de termos* 26
3.2 *Matriz de frequência absoluta de ocorrência de termos da coleção MSG* 39
3.3 *Tabela de contingência para a categoria A* 46

4.1 *Estrutura da representação BOW da coleção* Ω 54
4.2 *Estrutura da matriz de similaridade completa da coleção* Ω 54
4.3 *Estrutura da matriz SBOW da coleção* Ω 55
4.4 *Representação BOW da coleção M* 56
4.5 *Matriz de similaridade da coleção M utilizando o critério de seleção kNN com o valor de* $k = 3$ 56
4.6 *Representação SBOW da coleção M* 56

6.1 Ganhos obtidos em $macroF_1$ e $microF_1$ sobre o método kNN ao aplicar o método kINN ou kSNN nas coleções Reuters-AT, 20NG-AT e Ohsumed-AT 66
6.2 Ganhos obtidos em $macroF_1$ e $microF_1$ sobre o método kNN ao aplicar os métodos kINN ou kSNN nas coleções Reuters-NS, 20NG-NS e Ohsumed-NS. 66
6.3 Ganhos obtidos em $macroF_1$ e $microF_1$ sobre o método kNN ao aplicar o método kINN ou kSNN nas coleções Reuters-ST, 20NG-ST e Ohsumed-ST. 66
6.4 Ganhos obtidos em $macroF_1$ e $microF_1$ sobre o método kNN ao aplicar o método kINN ou kSNN nas coleções Reuters-FS, 20NG-FS e Ohsumed-FS. 67
6.5 Maiores valores obtidos em $macroF_1$ e $microF_1$ na aplicação dos método kNN, kINN e kSNN nas versões AT, NS, ST e FS das coleções de documentos Reuters, 20NG e Ohsumed. 67
6.6 Matriz de confusão da primeira partição resultante da aplicação do método kNN com $k = 40$ na coleção Reuters-NS 70
6.7 Precisão média e cobertura média da categoria dominante das coleções Reuters-NS e Ohsumed-NS. 70
6.8 Precisão média, cobertura média e F_1 das categorias da coleção Reuters-NS ao aplicar os métodos kNN e kINN. 71
6.9 Desvio-padrão em $microF_1$ e $macroF_1$ na aplicação dos métodos kNN, kINN e kSNN nas coleções Reuters-NS, 20NG-AT e Ohsumed-ST. 72
6.10 Maiores valores obtidos em $macroF_1$ e $microF_1$ na aplicação dos métodos kNN, kINN e kSNN nas coleções de documentos Reuters, 20NG e Ohsumed. 76
6.11 Ganhos obtidos em $macroF_1$ e $microF_1$ ao aplicar a abordagem de geração de características com peso máximo nas coleções Reuters-NS, 20NG-AT e Ohsumed-ST. 77

6.12 Ganhos obtidos em $macroF_1$ e $microF_1$ ao aplicar a abordagem de geração de características com peso 1 nas coleções Reuters-NS, 20NG-AT e Ohsumed-ST. 78
6.13 Valores da MRR dos conjuntos Q_n e Q_o nas coleções de documentos FC1. 79

7.1 Ganhos obtidos em $macroF_1$ e $microF_1$ sobre o método SVM ao aplicar os métodos kINN ou kSNN nas coleções Reuters, 20NG e Ohsumed. 81

CAPÍTULO 1

Introdução

1.1 Contextualização

A organização da informação é uma preocupação dos seres humanos desde o surgimento das primeiras civilizações, há cerca de 4.000 anos [9]. Naquele período, registros contábeis, ordenanças do governo, contratos e sentenças judiciais eram conservados e organizados em tábulas de argila. Com o passar dos anos, essas tábulas foram substituídas pelo papel, a quantidade de documentos aumentou consideravelmente e a atividade de localizá-los com agilidade tornou-se um grande desafio para a organização da informação.

Na tentativa de localizar documentos com agilidade foram criadas ferramentas. A mais importante dessas ferramentas, denominada índice, possibilita referenciar documentos (ou partes deles) para posteriormente identificá-los e/ou localizá-los [42]. Nas bibliotecas, por exemplo, o índice é utilizado por profissionais especializados (bibliotecários) para organizar livros, enciclopédias, dicionários, manuais, periódicos, entre outros documentos escritos em folhas de papel.

O avanço tecnológico e o surgimento dos computadores possibilitaram o desenvolvimento das bibliotecas digitais, onde é possível armazenar os índices e, em alguns casos, os documentos das bibliotecas tradicionais em formato digital. As bibliotecas digitais criaram novas demandas para a organização da informação que influenciaram no surgimento da Recuperação de Informação (RI), uma área de pesquisa que lida com o problema de representar, organizar e armazenar informações para o usuário acessá-las com o uso do computador [9]. Recuperação de Informação

Até o início da década de 90, as pessoas geralmente utilizavam os sistemas de RI para pesquisar por informações em coleções especializadas, tais como as bibliotecas digitais sobre publicações científicas [80]. Essas coleções são organizadas em áreas do conhecimento, possuem vocabulário e estrutura, controlados e padronizados por um profissional especializado em uma etapa denominada controle editorial. Além disso, os usuários desses sistemas geralmente possuem treinamento para formular consultas, permitindo-lhes expressar melhor as suas necessidades de informação. Nesse período, a

área de RI não possuía muitos desafios, uma vez que os sistemas de RI forneciam um suporte adequado e atendiam, em grande parte, às necessidades dos seus usuários.

No início da década de 90 surgiu a *Web*, um sistema distribuído de hipermídia, onde as pessoas geralmente procuram por informações das mais variadas áreas do conhecimento. A volumosa quantidade de documentos da *Web* e a impossibilidade de realizar um controle editorial generalizado nesse sistema, contribuíram para o surgimento de um dos maiores desafios enfrentados pela área de RI atualmente: a organização dos documentos da *Web*.

Os diretórios *Web* (por exemplo, *Yahoo! Directory*[1], *dmoz Open Directory Project - ODP*[2] e *Google Directory*[3]) são aplicações que tentam organizar os documentos da *Web* em uma hierarquia de tópicos para facilitar a navegação nos documentos desse sistema. A expansão e a manutenção desses diretórios tem sido feita manualmente por editores que analisam o conteúdo dos documentos da *Web* e classificam-nos em determinados tópicos. Entretanto a classificação manual desses documentos é ineficaz e ineficiente devido, principalmente, à quantidade de documentos publicados na *Web*.

Além dos diretórios *Web*, atualmente diversas aplicações requerem alguma forma de classificação automática, como por exemplo, a filtragem de mensagens eletrônicas, que possibilita identificar e excluir mensagens maliciosas ou indesejáveis (também denominadas *spans*); a personalização de conteúdo, que possibilita organizar notícias em canais temáticos e encaminhar para os usuários somente as notícias relacionadas aos seus perfis de interesse; o direcionamento de publicidade, que apresenta ao usuário apenas publicidade relacionada à categoria de seu interesse (por exemplo, esporte e diversão) e o auxílio no diagnóstico de doenças, que possibilita a identificação de uma doença ou do quadro clínico de um paciente conforme o seu histórico clínico [41] [49] [112].

A área de pesquisa que lida com o problema de classificar documentos automaticamente é a classificação automática de textos (CAT). Essa área multidisciplinar[4] está em evidência e tem despertado o interesse de diversos pesquisadores e empreendedores, principalmente após a popularização das aplicações da Internet, tais como a *Web* e o correio eletrônico. Para classificar documentos automaticamente é utilizada tradicionalmente a abordagem de aprendizagem supervisionada [86].

A abordagem de aprendizagem supervisionada consiste, resumidamente, no processo de construir um modelo utilizando documentos pré-classificados em determinadas categorias por um especialista (denominados *documentos de treino*) e avaliar esse mo-

[1] http://www.yahoo.com/
[2] http://www.dmoz.org
[3] http://www.google.com.br/dirhp
[4] A CAT utiliza técnicas de várias áreas, tais como: inteligência artificial, estatística, linguística computacional, recuperação de informação, mineração de dados, entre outras disciplinas.

delo utilizando novos documentos (denominados *documentos de teste*). Ao final desse processo, espera-se que as classificações dos documentos de teste realizadas pelo modelo coincidam com as classificações que seriam realizadas pelo especialista.

1.2 Problemas e Objetivos

Desde o início dos anos 90, grande parte das pesquisas realizadas na área de CAT têm procurado melhorar o desempenho (*eficácia* ou *eficiência*) do classificador automático [112]. A eficácia mensura a habilidade de um classificador automático decidir corretamente a categoria (ou classe) de determinado documento. A eficiência geralmente mensura o tempo gasto por um classificador automático para decidir a categoria de determinado documento.

Para um classificador automático ser eficaz, os seguintes aspectos devem ser observados:

- qualidade e quantidade de documentos previamente classificados por um especialista;
- qualidade do método responsável por gerar o classificador automático;
- qualidade das características dos documentos. Características são componentes dos documentos utilizados como informações no processo de classificação. As características mais comuns da CAT são os termos dos documentos. As ligações *hiperlinks* e as *tags*, quando disponíveis nos documentos, também podem ser utilizadas como características, apesar deste trabalho não utilizá-las.

Este trabalho atua nesses dois últimos aspectos na tentativa de propor melhorias para a CAT. Especificamente, este trabalho pretende melhorar a eficácia da CAT a partir da investigação de variações do método de classificação kNN e da investigação da geração de novas características em documentos. A seguir, são apresentados os dois problemas de pesquisa e os objetivos, relativos a essas investigações, que direcionaram o trabalho.

1.2.1 Variações do método kNN

O método dos k vizinhos mais similares (kNN, do inglês *k nearest neighbors*) tem sido aplicado na solução de problemas de CAT desde o início das pesquisas nessa área e, apesar de simples, tem se mostrado um dos métodos mais eficazes já propostos [134]. Para classificar um documento d, ainda não classificado (denominado *documento de teste*), esse método tradicionalmente realiza as seguintes atividades:

1. A similaridade entre o documento de teste d e cada um dos documentos que foram previamente classificados por um especialista (denominados *documentos de treino*)

é calculada utilizando alguma medida de similaridade entre documentos, tal como a medida do cosseno (Seção 3.1.3) [107].
2. Os k documentos de treino mais similares ao documento d são selecionados (k vizinhos mais próximos).
3. O documento d é classificado em determinada categoria de acordo com algum critério de agrupamento dos k vizinhos mais próximos selecionados na etapa anterior (por exemplo, a categoria que possuir a maioria dos k vizinhos mais próximos ao documento de teste d).

O critério de similaridade é um aspecto que possui grande influência no desempenho do método kNN [113]. Esse critério é composto pela medida de similaridade, ou função de distância (conforme a primeira atividade realizada pelo método kNN), e pelo critério de seleção (conforme a segunda atividade realizada pelo método kNN). O critério de seleção determina a forma de escolha dos k vizinhos de um documento de teste d. Por exemplo, selecionar os 5 documentos de treino mais similares ao documento de teste d é um critério de seleção.

A maior parte dos trabalhos relacionados ao critério de similaridade tem estudado diferentes medidas de similaridade na tentativa de aumentar a eficácia da CAT utilizando o método kNN [32] [48] [54] [98] [114]. Entretanto, apesar da relevância, existem poucos estudos sobre o critério de seleção na tentativa de aumentar a eficácia desse método [11] [58] [129].

O critério de seleção tradicionalmente adotado pelo método kNN consiste em selecionar os k documentos de treino mais similares ao documento de teste d. Tendo em vista a constatação da importância do critério de seleção na eficácia do método kNN e a escassez de pesquisas anteriores relacionadas ao assunto, levantou-se a seguinte questão:

Problema de pesquisa 1 *A adoção de novos critérios de seleção pelo método kNN pode aumentar a eficácia desse método?*

Em relação aos novos critérios de seleção, duas hipóteses foram levantadas e investigadas experimentalmente neste trabalho. A primeira delas consiste na seguinte ideia: selecionar os documentos de treino que possuem o documento de teste d entre os seus k vizinhos mais próximos pode gerar mais vizinhos do documento d que o critério de seleção tradicionalmente utilizado pelo método kNN e, portanto, esse novo critério é mais confiável que o critério utilizado pelo kNN, dado que a decisão quanto à categoria do documento d se baseia em uma quantidade maior de documentos de treino. A segunda hipótese é que um novo critério que corresponda a uma combinação do critério sugerido na primeira hipótese com o critério tradicional utilizado pelo método kNN possibilita selecionar os vizinhos "mais similares" ao documento de teste d, proporcionando uma decisão mais confiável em relação à categoria desse documento.

Para confirmar ou refutar essas hipóteses foram propostos dois critérios de seleção a serem adotados pelo método kNN que ainda não tinham sido explorados na CAT:

- O primeiro critério de seleção proposto consiste em selecionar os documentos de treino que possuem o documento de teste *d* entre os seus *k* vizinhos mais similares. Esse critério foi denominado *kNN invertido* (kINN, do inglês *k-Inverse Nearest Neighbors*) e para o método kNN adotá-lo foi proposta uma variação do kNN denominada kINN (homônimo do critério de seleção kINN).
- O segundo critério de seleção proposto, denominado *kNN simétrico* (kSNN, do inglês *k-Symmetric Nearest Neighbors*), é uma combinação do critério kNN com o critério kINN e consiste na interseção dos documentos selecionados pelos critérios kNN e kINN. Em outras palavras, o critério kSNN seleciona um documento de treino desde que ele esteja entre os *k* documentos mais próximos ao documento *d* e o documento *d* esteja entre os *k* documentos mais similares a esse documento de treino.

A eficácia do método kNN e dos métodos propostos (kINN e kSNN) foi avaliada aplicando-os na CAT em três diferentes coleções de documentos que são referências na literatura: Reuters, 20NG e Ohsumed.

1.2.2 Geração de características

Outro aspecto que possui grande influência na eficácia da CAT é a forma de representação dos documentos de uma coleção. Esses documentos geralmente são representados como uma matriz documento-termo conhecida como conjunto de palavras (BOW, do inglês *bag-of-words*).

A incorporação de novas características nos documentos pode aumentar a eficácia da CAT [51] [67]. A geração de características (do inglês *feature construction*) é uma das técnicas utilizadas com esse propósito. Essa técnica consiste em gerar novas características na matriz documento-termo que representa os documentos de uma coleção [51].

Diversos estudos estão relacionados à extensão da abordagem BOW para a CAT. Entre esses estudos, alguns buscam estender essa matriz utilizando *n*-gramas[5] [22] [85] [90] [96] [103] ou modelos estatísticos do idioma [97]. Outros estudos buscam gerar características a partir da informação sintática fornecida pelos documentos, tal como no etiquetamento da parte do discurso (POS, do inglês *part-of-speech*) ou na análise

[5]Um n-grama é uma sequência de *n* letras ou palavras, onde *n* geralmente é 1, 2 ou 3, respectivamente monograma, bigrama e trigrama

gramatical [12] [105]. Entretanto, nenhum trabalho que explorasse as informações sobre os documentos mais similares providas pelos critérios de seleção para gerar características em documentos foi encontrado.

Tendo em vista as pesquisas anteriores realizadas sobre geração de características e que as informações sobre os documentos mais similares providas pelos critérios de seleção poderiam influenciar na eficácia da CAT, levantou-se a seguinte questão:

Problema de pesquisa 2 *A informação de documentos mais similares pode ser utilizada para gerar características em documentos e aumentar a eficácia da CAT?*

Em relação a essa questão, a hipótese é que os identificadores dos documentos de treino que estão entre os vizinhos mais próximos de um documento, de acordo com algum critério de seleção (kNN, kINN ou kSNN), podem ser utilizados como novas características para expandir o conjunto de termos dos documentos e aumentar a eficácia do classificador automático. Para verificar ou refutar essa hipótese foi definido o seguinte objetivo:

- Propor uma abordagem para gerar características em documentos utilizando as informações de documentos mais similares providas pelos critérios de seleção.

Especificamente, a abordagem de geração de características proposta consiste em expandir a representação BOW dos documentos de uma coleção com identificadores de documentos da matriz de similaridade resultante dos melhores resultados obtidos na aplicação do critério de seleção kNN, kINN ou kSNN nessa coleção.

Para avaliar a qualidade dessa abordagem, o método SVM (do inglês *Support Vector Machine*) [59] foi aplicado nas coleções Reuters, 20NG e Ohsumed (antes e após a aplicação da abordagem de geração de características nessas coleções). Esse método foi escolhido por ser considerado como método estado da arte na classificação de documentos[6] [19] [40] [51].

Os documentos da *Web* possuem características especiais tais como, *hiperlinks*, metadados e estruturação que diferem-nos dos documentos puramente textuais [49]. Essas características especiais são exploradas em muitos trabalhos na tentativa de aumentar a eficácia da CAT na *Web* [101]. Entretanto, este trabalho pretende melhorar a eficácia da CAT utilizando somente as características textuais dos documentos, independente do ambiente. Por exemplo, o ambiente poderia ser a *Web*, as bibliotecas digitais ou o correio eletrônico.

[6]Neste trabalho, o termo 'classificação de documentos' é considerado sinônimo do termo 'classificação de textos'.

1.3 Principais contribuições do trabalho

As principais contribuições deste trabalho são as seguintes:

- Proposição de dois novos critérios de seleção para o método kNN (critério de seleção kINN e critério de seleção kSNN) e duas novas variações do método kNN (método kINN e método kSNN) que utilizam, respectivamente, os critérios kINN e kSNN.
- Proposição e avaliação experimental de uma nova abordagem que utiliza a informação de documentos mais similares para gerar características em documentos.
- Estudo experimental comparando a eficácia da CAT utilizando os métodos propostos com a eficácia da CAT utilizando o método kNN.
- Estudo experimental comparando a eficácia da CAT utilizando o método SVM sem aplicar a abordagem de geração de características proposta com a eficácia da CAT utilizando o método SVM após aplicar essa abordagem.

A seguir é apresentada a organização deste trabalho.

1.4 Organização do trabalho

Neste capítulo, o contexto, a justificativa, as hipóteses, os objetivos e as principais contribuições deste trabalho foram apresentados. Os próximos capítulos desta dissertação estão organizados conforme descrito nos próximos parágrafos.

O Capítulo 2 apresenta os estudos relacionados à este trabalho em duas partes. A primeira parte apresenta os estudos relacionados aos critérios de seleção do método kNN e a segunda parte apresenta os estudos relacionados à geração de características.

O Capítulo 3 apresenta os principais conceitos relacionados a este trabalho em três partes. A primeira parte apresenta os conceitos relacionados à preparação de documentos, a segunda parte apresenta os conceitos relacionados à dimensionalidade dos documentos e a terceira parte apresenta os conceitos relacionados à classificação de documentos.

O Capítulo 4 apresenta as abordagens propostas neste trabalho em duas partes. A primeira parte apresenta os critérios de seleção e métodos propostos e a segunda parte apresenta a abordagem proposta para gerar características em documentos.

O Capítulo 5 apresenta o método adotado neste trabalho em cinco partes. A primeira parte apresenta uma visão geral do método adotado, a segunda parte apresenta as coleções utilizadas nos experimentos, a terceira parte apresenta as atividades relacionadas à preparação de documentos, a quarta parte apresenta o método de avaliação adotado

para avaliar o desempenho dos classificadores e a quinta parte apresenta os métodos de classificação utilizados nos experimentos.

O Capítulo 6 apresenta os resultados obtidos nos experimentos realizados neste trabalho em duas partes. A primeira parte apresenta os resultados experimentais relacionados às variações propostas do método kNN e a segunda parte apresenta os resultados experimentais relacionados à abordagem de geração de características em documentos proposta.

Por fim, o Capítulo 7 apresenta as conclusões obtidas neste trabalho e propõe possíveis trabalhos futuros.

CAPÍTULO 2

Trabalhos Relacionados

Este capítulo apresenta os estudos relacionados a este trabalho. Na Seção 2.1, são descritos os estudos relacionados aos critérios de seleção do método kNN e na Seção 2.2.1, são descritos os estudos relacionados à geração de características em textos.

2.1 Critérios de seleção para o método kNN

O critério de similaridade é um aspecto utilizado pelo método kNN que possui grande influência no desempenho desse método [113]. Esse critério é composto pela medida de similaridade e pelo critério de seleção dos vizinhos. A maior parte dos trabalhos relacionados ao critério de similaridade tem estudado diferentes medidas de similaridade na tentativa de aumentar a eficácia da CAT utilizando o método kNN [32] [48] [54] [98] [114] [126]. Entretanto, apesar da relevância, poucas pesquisas têm estudado o critério de seleção dos vizinhos na tentativa de aumentar a eficácia desse método [11] [58] [129].

Xie et al. [129] propuseram o método "vizinhança seletiva por redes bayesianas" (SNNBS, do inglês *selective neighborhood naive Bayes*). O método SNNB testa diferentes valores para o valor de k vizinhos mais próximos de um documento de teste d. Para cada valor de k testado, um classificador bayesiano local é gerado e avaliado para os k vizinhos mais próximos do documento de teste d. Após isso, o classificador bayesiano mais eficaz é utilizado para classificar o documento d. Esse método pode ser visto como um método hibrido (kNN e redes bayesianas) e embora o SNNB demonstre ganhos em eficácia com relação a alguns métodos de classificação, o SNNB demora muito tempo para finalizar a sua execução.

Baoli et al. [11] propuseram um método semelhante ao método kNN probabilístico [50], denominado ADAPT. O método ADAPT consiste em utilizar a informação fornecida pelo conjunto de treino para melhorar o desempenho do método kNN. Para isto, o ADAPT utiliza diferentes valores de k vizinhos mais próximos para predizer diferentes categorias. Dado um documento de teste d, o ADAPT obtém os k vizinhos mais próximos do documento d da mesma maneira que o método kNN. Após isso, o ADAPT calcula a

probabilidade do documento d pertencer à categoria c a partir dos k_c documentos de treino mais próximos que pertencem à categoria c. O documento d é classificado de acordo com a categoria que possuir maior probabilidade. Os experimentos realizados com o ADAPT mostraram que esse método é menos sensível à variação do parâmetro k do que o método kNN. Entretanto, a eficácia do método kNN mostrou-se equivalente ao comparar com a eficácia do método ADAPT.

Jiang et al. [58] propuseram o método kNN dinâmico por redes bayesianas com atributos ponderados (DKNN, do inglês *dynamic K-Nearest-Neighbor Naive Bayes with attribute weighted*). Esse método é denominado dinâmico pois o valor de k do método kNN varia dinamicamente de acordo com os documentos de treino. O método DKNN é executado em duas etapas: na etapa de treino, o melhor valor de k é aprendido para determinado conjunto de treino; na etapa de classificação, uma rede bayesiana local é gerada para o melhor valor de k obtido na etapa de treino e o documento de teste d é classificado de acordo com a categoria que possuir maior probabilidade nessa rede. Os experimentos realizados com o DKNN mostraram que esse método é mais eficaz do que o método kNN. Entretanto, esses experimentos não mostraram o ganho percentual do DKNN em relação ao método kNN.

Neste trabalho, foram propostas duas variações do método kNN. A primeira variação desse método é denominada de método *kNN invertido* (kINN) e consiste em classificar o documento de teste d de acordo com os documentos de treino que possuem o documento d entre os seus k vizinhos mais próximos. A segunda variação desse método, denominada kNN simétrico (kSNN), é basicamente uma combinação do método kNN tradicional com o método kINN e consiste em classificar o documento de teste d de acordo com a intersecção entre os documentos de treino selecionados pelos métodos kNN e kINN.

2.2 Representação dos documentos

Alguns estudos modificam a abordagem conjunto de palavras (BOW, do inglês *bag of words*). Em particular, representações baseadas em frases [36] [45] [74], identificação de entidades [71] (do inglês *named entities*) e aglomeração de termos [75] (do inglês *term clustering*) têm sido exploradas.

Lewis [73] verificou que as frases possibilitam representar a ideia de contexto e fornecem maior informação semântica do que os termos. Entretanto, o estudo concluiu que a utilização de termos como características é mais eficaz do que a utilização de frases na classificação de documentos.

Fuhr [43] introduziu a abordagem de indexação Darmstadt (DIA, do inglês *Darmstadt Indexing Approach*), que define características como propriedades de termos,

de documentos ou de categorias. Assim, a metainformação, tal como as posições dos termos nos documentos e o tamanho dos documentos podem ser considerados como características. A abordagem DIA pode ser utilizada em conjunto com outras representações baseadas em termos ou frases, tal como a BOW [112].

Krupka e Tishby [68] propuseram um arcabouço, representado por meta características, para incorporar conhecimento na etapa de aprendizagem na tentativa de melhorar a eficácia da classificação.

Bekkerman et al. [13] representaram documentos por um conjuntos de palavras no âmbito da abordagem do gargalo da informação (do inglês, *information bottleneck*) [99] [118]. Os conjuntos resultantes foram utilizados como novas características (centroides) em substituição aos termos originais.

2.2.1 Geração de características em textos

As técnicas relacionadas com a geração de características são úteis em diversas áreas da aprendizagem de máquina [37] [82] [83]. Essas técnicas consistem na identificação e geração de novas características com o objetivo de melhorar a descrição de determinado conceito do que utilizar somente as características presentes nos exemplos do treino. Nesse sentido, foram propostos alguns algoritmos para gerar características que melhoram o desempenho da classificação significativamente [10] [56] [84] [93]. Entretanto, poucos trabalhos se aplicam ao processamento de texto [25] [69] [85]. Nesta dissertação, propomos uma abordagem para gerar características nesse cenário.

Diversos estudos estão relacionados com a extensão da abordagem BOW para a classificação automática de textos. Entre esses estudos, alguns buscam estender essa abordagem utilizando n-gramas [22] [85] [90] [96] [103] ou modelos estatísticos do idioma [97]. Outros estudos buscam gerar características a partir da informação sintática fornecida pelos documentos, tal como no etiquetamento da parte do discurso (POS, do inglês *part-of-speech*) ou na análise gramatical [12] [105].

Mladenic e Grobelnik [89] [91] [92] utilizaram uma rede bayesiana para classificar documentos da *Web* com o objetivo de melhorar a eficácia do mecanismo de busca Yahoo! Além dos termos originais dos documentos, n-gramas (acima de 5 gramas) foram adicionadas à representação BOW dos documentos.

Caropreso et al. [22] utilizaram uma ideia mais sofisticada de n-gramas, em que cada n-grama correspondia a uma sequência de n raízes de termos ordenadas alfabeticamente. Por exemplo, de acordo com essa ideia, expressões como 'classificar textos' e 'a classificação de textos' correspondem às mesmas características. Esse estudo concluiu que a inclusão de n-gramas pode não melhorar a eficácia de um classificador e, em alguns casos, pode até piorar sua eficácia.

Mikheev [85] utilizou uma estrutura de concatenação de características como motor para gerar características em um arcabouço de maximização da entropia e aplicou-o na classificação de documentos, na detecção do limite de uma sentença e no etiquetamento da POS. Esse estudo utilizou a informação sobre unigramas, bigramas e trigramas para construir o espaço de característica e posteriormente, selecionar um conjunto de características de acordo com medidas probabilísticas.

Kudenko e Hirsh [69] propuseram um método para gerar características, denominado FGEN, que gera características booleanas para verificar a presença ou a ausência de determinadas subsequências selecionadas heuristicamente. Nesse estudo, foram realizados experimentos em três domínios diferentes: sequências de DNA, sequências de comandos UNIX e documentos textuais.

Cohen [25] realizou um estudo na tentativa de descobrir características a partir de um conjunto de exemplos, sem características, classificados em determinadas categorias. A classificação de artistas, dado um gênero musical, é um exemplo de aplicação que poderia se beneficiar dessa abordagem. Nesse estudo, alguns documentos da *Web* foram coletados e para identificar as características desses documentos foram utilizados os termos dos cabeçalhos HTML (do inglês *HyperText Markup Language*) que co-ocorriam, dada uma categoria. Além disso, esse estudo identificou outra fonte de características baseada em posições no código HTML. Por exemplo, se um nome aparece frequentemente em tabelas, este nome pode ser definido como uma característica.

Sahami et al. [106] utilizaram um conjunto de características, tais como a hora do dia que uma mensagem foi recebida ou se a mensagem possuía algum arquivo anexo, para filtrar mensagens eletrônicas inválidas. Esse estudo definiu aproximadamente 20 características, geradas manualmente e combinadas com os termos originais das mensagens. A classificação foi realizada a partir da seleção das melhores características, definidas de acordo com o critério de seleção informação mútua. Por fim, esse estudo sugeriu utilizar características de determinadas áreas para auxiliar a tarefa de classificação automática de textos nessas áreas.

Algumas abordagens para gerar características lidam com a situação em que os documentos de uma coleção possuem poucos termos [109] [117] [136] [137]. Zelikovitz e Hirsh [136] utilizaram um conjunto de exemplos não rotulados (exemplos virtuais) para intermediar a comparação de exemplos de teste com exemplos de treino. Quando um exemplo de teste era distinto de todos os exemplos de treino, os exemplos virtuais eram utilizados como ‘pontes’, influenciando no cálculo da similaridade entre os exemplos.

Em outro estudo, Zelikovitz e Hirsh [137] propuseram uma maneira alternativa de utilizar exemplos virtuais. Nesse estudo, documentos virtuais foram incluídos no conjunto de treino para realizar a análise semântica latente (LSA, do inglês *Latent Semantic Analysis* [30]) dos documentos desse conjunto. Os resultados da LSA facilitaram a com-

paração entre documentos de teste com documentos de treino. Entretanto, a utilização da LSA dificilmente pode melhorar a eficácia da classificação utilizando o método SVM e, em alguns casos, pode até diminuir a eficácia da classificação [128] [78].

Sassano [109] propôs técnicas para gerar exemplos virtuais para a classificação de textos. Nesse estudo, documentos virtuais foram criados a partir do acréscimo ou exclusão de um pequeno número de termos. A eficácia da classificação aumentou em situações que o conjunto de treino possuía poucos exemplos, mas à medida que a quantidade de exemplos de treino aumentou, os ganhos na eficácia da classificação não foram significativos.

Por fim, Taskar et al. [117] propuseram um método baseado em modelos probabilísticos para gerar características, denominado de características invisíveis. Essas características apareciam no conjunto de teste mas não apareciam no conjunto de treino e foram utilizadas na tarefa de classificar notícias e documentos da *Web*. Além disso, para prever o 'papel' das características invisíveis, foram utilizadas meta características, baseadas na vizinhança das características invisíveis.

A proposta de geração de características desta dissertação também pode ser utilizada para melhorar a eficácia da classificação quando há escassez de termos, embora não tenha sido especificamente planejada com este objetivo.

2.2.2 Outras formas de gerar características

Na recuperação de informação tem sido utilizadas técnicas para expandir as consultas com termos adicionais. A WordNet [38] é frequentemente utilizada como fonte de conhecimento externo e as consultas são enriquecidas com termos, escolhidos consultando dicionários e enciclopédias [123] [124], analisando o contexto em torno do termo da consulta [39] ou de acordo com a relevância da retroalimentação (do inglês, *relevance feedback*) [88] [130].

Por fim, existem diversos estudos para acrescentar conhecimento em técnicas de aprendizagem de máquina. As abordagens de transferência de conhecimento (do inglês, *Transfer learning*), transferem informações de diferentes tarefas relacionadas [17] [31]. A retroalimentação pseudo-relevante (do inglês, *Pseudo-Relevance Feedback* [104]) utiliza a informação dos documentos mais relevantes em uma consulta. Estudos recentes sobre métodos semisupervisionados [5] [6] [47] inferem informações para exemplos não rotulados, que estão disponíveis em quantidade maior do que exemplos rotulados.

Neste trabalho, é proposta uma abordagem para expandir o modelo BOW com identificadores de documentos obtidos nas matrizes de similaridades dos melhores resultados obtidos na avaliação do método kNN e suas variações propostas.

CAPÍTULO 3

Conceitos Relacionados

Este Capítulo apresenta os conceitos relacionados a esta dissertação. Na Seção 3.1, são descritos os principais conceitos relacionados à preparação de documentos, tais como a representação de documentos, as medidas da importância dos termos e as medidas de similaridade entre documentos, na Seção 3.2, são descritos os principais conceitos relacionados à dimensionalidade dos documentos, e na Seção 3.3, são descritos os conceitos relacionados à classificação de textos, tais como os paradigmas dessa área, a classificação automática de documentos, os principais métodos de classificação e os métodos normalmente utilizados para avaliar o desempenho de um classificador de documentos.

3.1 Preparação de documentos

A constituição de uma coleção de documentos é o primeiro passo da classificação automática de textos (CAT). Essa coleção pode ser composta por um conjunto de documentos sobre uma área específica do conhecimento de interesse de uma comunidade de usuários, tal como os documentos provenientes das bibliotecas digitais, ou pode ser composta por documentos de diferentes áreas do conhecimento, tal como os documentos provenientes da Internet.

Dada uma coleção de documentos, para um classificador automático de documentos acessá-la, antes é necessário indexar os documentos dessa coleção. A indexação é um processo que consiste em analisar os documentos de uma coleção e mapear o conteúdo desses documentos em uma representação padrão [112].

A indexação de documentos tem sido realizada de acordo com duas abordagens: linguística ou estatística. A abordagem linguística é dependente do idioma e realiza a análise textual de um documento em diferentes níveis linguísticos, tais como: léxico, sintático, semântico e pragmático discursivo. Já a abordagem estatística realiza a análise textual de um documento de acordo com cálculos baseados nos termos de um documento.

Algumas abordagens de indexação priorizam frases ao invés de termos [18] [22] [87], a semântica do termo ao invés do relacionamento entre os termos [24] [61] ou a

estrutura hierárquica do texto ao invés do próprio texto [8]. Os modelos mais expressivos possibilitam capturar o significado de um documento melhor que os modelos baseados em palavras. Entretanto, esses modelos são mais complexos e possuem qualidade estatística inferior aos modelos baseados em palavras. O modelo espaço vetorial (VSM, do inglês *Vector Space Model*) possui uma boa relação entre expressividade e complexidade [74].

O VSM foi o modelo adotado para representar os documentos das coleções utilizadas nesta dissertação por ser muito utilizado na CAT, por possibilitar analisar documentos estatisticamente e realizar comparações entre documentos.

3.1.1 Representação de documentos

O VSM, ou modelo vetorial, é um modelo simples, tradicional e efetivo que possibilita representar documentos como vetores e realizar qualquer operação algébrica para comparar documentos [108]. Contudo, esse modelo não possibilita determinar a ordem de exibição dos termos de um documento nem as relações semânticas entre esses termos [77].

Os documentos de uma coleção D são representados no VSM como pontos em um espaço euclidiano multidimensional, onde cada dimensão corresponde a um termo distinto dessa coleção. O conjunto T de termos distintos da coleção D, denominado vocabulário da coleção D, é obtido em um processo denominado análise léxica, após a realização das seguintes atividades:

- remover marcas de pontuação.
- substituir marcas de tabulação e outros caracteres não textuais por espaços em branco.
- converter termos para minúsculo.
- excluir caracteres que não sejam alfanuméricos.

Cada termo do conjunto T pode ser composto por apenas uma palavra (unigramas), várias palavras (bigramas, trigramas ou n-gramas) ou frases, e possui um peso associado para determinar o seu grau de importância [107].

Dado um documento $d_i \in D$, esse documento é formalmente representado no VSM da seguinte forma:

$$d_i = \{p_{i,1}, p_{i,2}, p_{i,3}, \ldots, p_{i,|T|}\}$$

onde T é o conjunto do vocabulário da coleção D e $p_{i,j} (1 \leq j \leq |T|)$ é o peso do termo t_j no documento d_i, tal que $p_{i,j} = 0$ se o termo t_j não ocorre no documento d_i.

A representação de documentos mais utilizada na CAT, também conhecida como representação do conjunto de termos (BOW, do inglês *bag of words*), considera apenas as

palavras como termos. Conforme essa representação, cada elemento (i, j) de uma matriz documento-termo corresponde ao peso $p_{i,j}$ do termo t_j no documento d_i. A Tabela 3.1 mostra o exemplo da representação BOW dos documentos da coleção D'.

D'	t_1	t_2	t_3	t_4	t_5
d_1	68	56	46	203	92
d_2	1	82	289	0	25
d_3	1	0	225	0	54
d_4	430	392	1	54	121

Tabela 3.1: *Matriz documento-termo com a frequência absoluta de ocorrência de termos*

Na matriz representada na Tabela 3.1, para atribuir um valor para cada elemento $p_{i,j}$, utilizou-se a medida 'frequência absoluta'. Entretanto, existem diferentes medidas de importância que podem ser atribuídas ao conjunto de pesos de uma matriz BOW. A sessão seguinte trata desse assunto.

3.1.2 Medidas da importância dos termos

As métricas mais conhecidas e utilizadas na classificação automática de documentos são: binária, frequência dos termos (TF, do inglês *Term Frequency*), frequência invertida dos documentos (IDF, do inglês *Inverse Document Frequency*)) e TF-IDF (do inglês *Term Frequency - Inverse Document Frequency* [60]).

A métrica binária é a maneira mais simples de atribuir o peso do termo t_j ao documento d_i ($p_{i,j}$). Essa métrica utiliza os valores 1 e 0 para determinar, respectivamente, se um termo aparece no documento (é importante) ou não aparece nesse documento (não é importante). A métrica binária de um termo é calculada pela Equação 3-1:

$$bin(d_i, t_j) = \begin{cases} 1, \text{ se o termo } t_j \text{ aparece no documento } d_i \\ 0, \text{ caso contrário} \end{cases} \tag{3-1}$$

Entretanto, a métrica binária não apresentou bons resultados ao mensurar a importância dos termos [108]. Na tentativa de melhorar esses resultados, a métrica binária pode ser substituída pela TF, calculada pela Equação 3-2 [108].

$$tf(d_i, t_j) = log(1 + f(d_i, t_j)) \tag{3-2}$$

onde $f(d_i, t_j)$ é a frequência absoluta do termo t_j no documento d_i.

Conforme a TF, quanto maior a quantidade de ocorrências do termo t_j no documento d_i, maior a importância desse termo no documento d_i. Entretanto, essa métrica pode prejudicar cálculos de similaridade entre documentos, pois termos com frequência de ocorrência menor do que outros podem possuir uma capacidade maior para discriminar

documentos. Uma das maneiras de evitar esse problema é atribuir um peso alto quando um termo ocorre em poucos documentos e um peso baixo, caso contrário. Esse é o propósito da métrica IDF, calculada pela Equação 3-3.

$$idf(t_j) = log\frac{|D|}{doc(t_j)} \tag{3-3}$$

onde D é uma coleção de documentos e $doc(t_j)$ é a quantidade de documentos da coleção D onde o termo t_j aparece.

A IDF pode atribuir pesos iguais para termos com alta frequência de ocorrência por não considerar a frequência de ocorrência de um termo no documento. Para contornar esse problema, é necessário que os termos que ocorram muito em determinado documento (frequência local alta) e, simultaneamente, que ocorram em poucos documentos (frequência global alta) recebam pesos altos. Esse é o propósito da métrica TF-IDF.

A TF-IDF é uma métrica que combina a TF com a IDF. Dessa forma, o peso total do termo t_j no documento d_i se torna a combinação do seu peso local (a métrica TF) e global (a métrica IDF). A TF-IDF é calculada pela Equação 3-4.

$$w(d_i,t_j) = tf(d_i,t_j) \times idf(t_j) \tag{3-4}$$

onde $tf(d_i,t_j)$ é a TF do termo t_j no documento d_i, calculada pela Equação 3-2, e $idf(t_j)$ é a IDF do termo t_j, calculada pela Equação 3-3.

Como exemplo, considere uma coleção de documentos $D = \{d_1, d_2, \ldots, d_{|D|}\}$, tal que $|D| = 1.000$, o termo t_1 aparece 5 vezes no documento d_1 e esse termo aparece em 100 documentos da coleção D. Conforme esse cenário, os cálculos da TF, IDF e TF-IDF são os seguintes:

$$tf(d_1,t_1) = log(6) = 0,77 \qquad idf(t_1) = log(10) = 1,00$$

$$w(d_1,t_1) = log(6) \times log(10) = 0,77$$

Alguns métodos de classificação utilizam medidas de similaridades entre documentos para classificá-los automaticamente. Uma vez atribuídos os pesos para os termos de cada um dos documentos de uma coleção, algumas métricas podem ser utilizadas para calcular a similaridade entre documentos. A sessão seguinte trata das medidas de similaridade entre documentos.

3.1.3 Medidas de similaridade entre documentos

Dado que os documentos de uma coleção D' são representados conforme o modelo VSM (Seção 3.1.1), a similaridade entre dois documentos dessa coleção pode

ser definida como a distância entre dois pontos ou o ângulo entre dois vetores no espaço euclidiano $R^{|T|}$, onde T é o conjunto do vocabulário da coleção D.

As medidas mais utilizadas para o cálculo da similaridade entre documentos na CAT são a distância euclidiana e o cosseno. Dados os documentos d_1 e d_2, a distância euclidiana entre esses documentos é calculada pela Equação 3-5:

$$euc(d_1, d_2) = \sqrt{\sum_{h=1}^{|T|} (p_{1,h} - p_{2,h})^2} \tag{3-5}$$

onde $p_{i,j}$ é o peso do termo t_j no documento d_i (Seção 3.1.1).

Outra medida muito utilizada para calcular a similaridade entre os documentos de uma coleção é o cosseno do ângulo θ entre dois documentos. Dessa forma, dados os documentos d_1 e d_2, a similaridade entre eles é calculada pela Equação 3-6:

$$cos(d_1, d_2) = \frac{\sum_{h=1}^{|T|} p_{1,h} p_{2,h}}{\sqrt{\sum_{h=1}^{|T|} {p_{1,h}}^2} \sqrt{\sum_{h=1}^{|T|} {p_{2,h}}^2}} \tag{3-6}$$

A Figura 3.1 ilustra a medida do cosseno do ângulo θ entre os documentos d_1 e d_2, onde $|T| = 2$ e $\vec{v}(d_i)$ representa o documento d_i no espaço euclidiano R^2. Ao calcular o cosseno do ângulo θ entre dois documentos, quanto mais próximo de 1 for o resultado, maior é a similaridade entre os documentos.

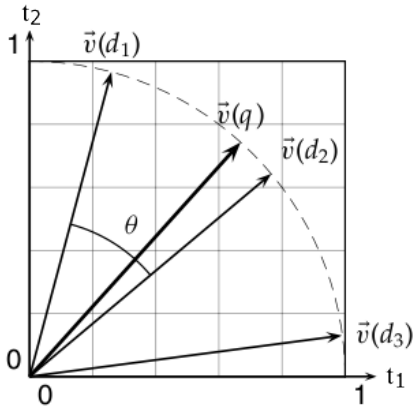

Figura 3.1: *Cosseno* θ *entre os documentos* d_1 *e* d_2. *(Adaptado de [80])*

3.2 Dimensionalidade de documentos

Um classificador automático de documentos normalmente realiza uma etapa de treino antes de ser utilizado efetivamente para classificar novos documentos. O desem-

penho desse classificador depende, entre outros fatores, da quantidade de documentos de treino e da qualidade dos termos que constituem esses documentos.

Um dos maiores desafios enfrentados pela classificação automática de textos (CAT) é a construção ou obtenção de documentos de treino [80]. Para construir um classificador com eficácia elevada é necessário cerca de 50 à 100 documentos de treino por termo existente em um conjunto de treino [44]. Além disso, caso existam termos irrelevantes, redundantes ou incorretos (ruídos), eles devem ser cuidadosamente excluídos desse conjunto.

Caso a quantidade de documentos de treino não seja suficiente em relação à quantidade de termos no conjunto de treino, a eficácia de um classificador pode ser prejudicada. Esse problema é conhecido como fenômeno do pico (do inglês *peaking phenomena*) [116].

Em muitos casos, a quantidade de documentos de treino necessários para construir um classificador com eficácia elevada pode ser exponencial em relação à quantidade de termos existentes no conjunto de treino [119]. Nessas circunstâncias, o custo computacional (memória e processamento) para realizar a classificação pode ser muito alto ou até inviável. Esse fenômeno é conhecido como maldição da dimensionalidade (do inglês *curse of dimensionality*) [15].

A maldição da dimensionalidade pode ser amenizada utilizando um espaço de termos que possua somente os termos essenciais para a representação dos documentos de uma coleção. Entretanto, um grande desafio é descobrir quais termos são essenciais. Para isto, são utilizadas técnicas para reduzir a dimensionalidade do espaço de termos e ao mesmo tempo assegurar que a eficácia da classificação não seja afetada [112]. Dessa forma, os custos computacionais diminuem e pode ser possível a construção de classificadores com baixas taxas de erro.

A redução da dimensionalidade pode consequentemente reduzir o problema do sobreajuste (do inglês *overfitting*). O sobreajuste ocorre quando um classificador se adapta aos documentos de treino, podendo reduzir a sua taxa de acerto na classificação de novos documentos. Quando ocorre esse problema, o classificador tende a ser muito bom na classificação de documentos de treino, mas muito ruim na classificação de novos documentos [86].

Para aumentar a eficácia da CAT é necessário investigar a dimensionalidade ideal dos documentos de uma coleção. Essa investigação consiste na realização de testes (tentativa e erro) utilizando métodos para a redução da dimensionalidade da representação dos documentos. Para isto, pelo menos um dos seguintes processos deve ser executado: filtragem, conflação ou extração de características, que serão tratados nas próximas seções.

3.2.1 Filtragem

A filtragem é um processo para remover termos irrelevantes em uma coleção de documentos. Esses termos possuem pouca ou nenhuma importância para a CAT. A remoção de termos geralmente se baseia em um conjunto de palavras irrelevantes denominado *stoplist* ou dicionário negativo. A *stoplist* normalmente é composta por artigos, preposições, conjunções e cada elemento da *stoplist* é denominado *stopword* [9].

Além disso, termos com alta frequência de ocorrência nos documentos de uma coleção devem ser removidos, pois geralmente não fornecem informações que possibilitam discriminar a categoria dos documentos e termos com baixa frequência de ocorrência geralmente não possuem relevância estatística e, portanto, também devem ser removidos [135].

3.2.2 Conflação

A conflação é o ato de agrupar ou combinar para igualar variantes morfológicas de termos [42]. As principais técnicas de conflação são a lematização e o *stemming*.

A lematização mapeia formas verbais para o tempo infinitivo e os substantivos para o singular. Para isso, a classe gramatical (POS, do inglês *Part of Speech*) de cada termo de um documento precisa ser atribuída a partir de uma etapa denominada etiquetagem do texto. Esse processo consome muito tempo e podem ocorrer erros no corte de árvores sintáticas. Por isso, a técnica de *stemming* é mais empregada.

O *stemming* é uma técnica que consiste em reduzir todos os termos de um documento ao mesmo *stem*, por meio da retirada de afixos (prefixos e sufixos) dos termos. O *stem* pode ser o próprio radical morfológico ou a parte essencial do item lexical do termo. O mais importante é que o *stem* seja capaz de capturar o significado do termo, sem perder muito detalhe [95]. Os algoritmos de *stemming* mais utilizados na CAT são os algoritmos Porter [100] e Lovins [79]. Um exemplo típico de um *stem* é *comput* que é o *stem* dos termos *computador*, *computar* e *computadores*.

3.2.3 Seleção de características

Mesmo após a filtragem e a conflação, a matriz conjunto de termos (BOW, do inglês *bag of words*) resultante ainda pode possuir alta dimensionalidade. Além disso, essa matriz pode possuir termos irrelevantes, redundantes ou com ruídos (erros). Para solucionar esses problemas, a seleção de características deve ser realizada.

A seleção de características é uma abordagem que consiste em selecionar um subconjunto de um conjunto de características de uma coleção de documentos que possibilite a maior redução possível na taxa de erro de classificação em relação ao

conjunto original de características [57]. No contexto deste trabalho, as características correspondem aos termos dos documentos.

Além da melhoria da eficácia de um classificador, a seleção de características pode proporcionar as seguintes vantagens [51]:

1. melhorar a eficiência do classificador.
2. economizar recursos de armazenamento de informações (memória).
3. facilitar a compreensão e a visualização das características dos documentos.

Dado um conjunto de características $T = \{t_1, t_2, t_3, \ldots, t_n\}$, onde n corresponde ao tamanho do espaço de características da coleção de treino D, a seleção de características consiste em selecionar um subconjunto do conjunto T de tamanho k para atingir um determinado objetivo, definido pela função critério $J(x)$.

Os objetivos da seleção de características podem ser divididos em três tipos [70]: No tipo A, a função $J(x)$ determina a menor taxa de erro na classificação de um subconjunto com k características. No tipo B, a função $J(x)$ determina o menor subconjunto de características que satisfaça alguma condição (por exemplo, a taxa de erro abaixo de um valor especificado). Por fim, no tipo C, a função $J(x)$ combina o tipo A e o tipo B, ou seja, procura encontrar o menor subconjunto de características que possua a menor taxa de erro de classificação.

O problema da seleção de características pode ser visualizada como um problema de busca, conforme ilustrado na Figura 3.2.

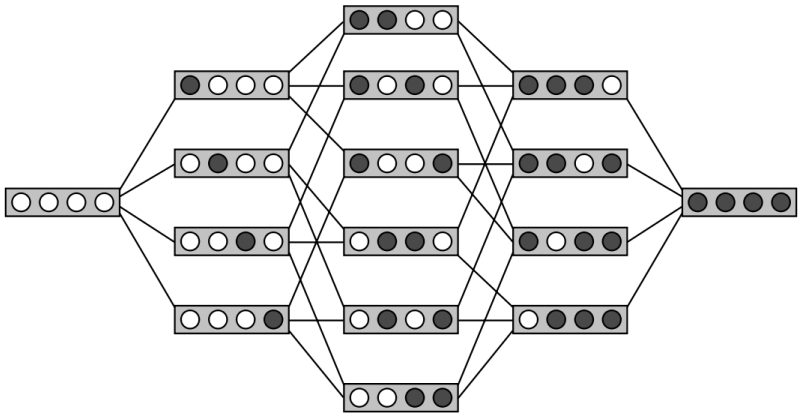

Figura 3.2: *Espaço de busca de um conjunto com quatro características [19]*

A Figura 3.2 mostra um diagrama de estados com quatro características. Cada estado determina um subconjunto de características escolhido em um determinado instante. O círculo branco indica a ausência de uma determinada característica, enquanto que o círculo preto indica a presença. O espaço de busca de um conjunto de características é determinado pelo conjunto de todos os estados possíveis no diagrama: $\sum_{e=0}^{4} C_{e,4}$, onde

$C_{4,a}$ indica de quantas formas distintas é possível escolher a elementos de um conjunto de 4 elementos.

Os algoritmos utilizam diferentes estratégias para percorrer o espaço de busca de um conjunto de características. As estratégias se diferenciam fundamentalmente quanto à eficácia em solucionar o problema de busca, que pode ser ótimo, garantindo a melhor solução entre todas as possíveis, ou subótimo, não garantindo a melhor solução. Os principais algoritmos ótimos são a busca exaustiva e o 'ramificar e limitar' (do inglês *branch-and-bound*) [94].

A busca completa ou exaustiva possibilita avaliar subconjuntos ótimos de acordo com a função critério $J(x)$. Para isso, todos os subconjuntos de características possíveis são avaliados. Entretanto, em muitos casos, o espaço de busca é grande demais para ser explorado exaustivamente tornando esse algoritmo computacionalmente intratável (essa solução é NP-Completa) [4] [51].

Uma solução ótima pode ser obtida sem precisar analisar todos os subconjuntos de características. Para isto, é necessário parar a execução do algoritmo quando for identificado que a função critério é monotônica[1]. O algoritmo 'ramificar e limitar' utiliza essa abordagem.

O algoritmo 'ramificar e limitar' modela o conjunto de características como uma árvore de busca e as folhas dessa árvore representam os subconjuntos de características possíveis. Esse algoritmo, no pior dos casos, é igual à busca exaustiva, mas possui um alto custo computacional e deve ser utilizado apenas em situações em que a coleção de documentos possui menos de 40 características [63]. Nas demais situações, os algoritmos subótimos devem ser utilizados.

Os algoritmos subótimos podem ser categorizados de acordo com a forma que os subconjuntos são expandidos: seleção para frente (do inglês *forward selection*) [62], seleção para trás (do inglês *backward selection*) [81], seleção bidirecional (do inglês *bidirectional selection*) [62] e seleção randômica (do inglês *random selection*).

Na seleção para frente, por exemplo, no início do processo, o conjunto inicial de características é vazio (estado mais à esquerda da Figura 3.2). A cada iteração do problema, características são acrescentadas e os subconjuntos de características resultantes, até atingir um tamanho k, são avaliados pela função critério $J(x)$. Ao final do processo, o subconjunto de características com até k características e com melhor função critério $J(x)$ é determinado.

A função critério $J(x)$ pode ser dependente ou independente de um algorítimo de classificação [77]. A função critério dependente, também conhecida como invólucro (do inglês *wrapper*), seleciona um subconjunto do conjunto de características Dt de

[1] Uma função é monotônica caso $f(x_1) \geq f(x_2)$ sempre que $x_1 \geq x_2$

acordo com alguma estratégia de busca e avalia esse subconjunto executando-o sobre um algoritmo de classificação.

As desvantagens da função critério dependente são: o custo computacional dessa abordagem é alto mesmo em coleções com poucos documentos, uma vez que, o algoritmo de classificação é executado para cada subconjunto de características escolhido. Por outro lado, é possível aumentar a eficácia de um classificador de documentos em cenários específicos [63].

Já a função critério independente, também conhecida como filtro, seleciona um subconjunto do conjunto de características *Dt* de acordo com alguma estratégia de busca e avalia esse subconjunto utilizando alguma heurística de avaliação [3] [53] [66] [76]. Nesse último caso, não há a execução de um algoritmo de classificação para avaliar o subconjunto de características.

O filtro pode construir um subconjunto de características de duas formas diferentes. Na primeira, cada característica é avaliada isoladamente utilizando um *ranking* de características. Dessa maneira, as características que estão posicionadas no topo do *ranking* são normalmente selecionadas para constituir o subconjunto de características. Na segunda, subconjuntos de características são avaliados iterativamente e o melhor subconjunto é escolhido para constituir o conjunto de características.

As vantagens da utilização do filtro são: as características selecionadas podem ser utilizadas por diferentes classificadores e normalmente essa abordagem é eficiente em uma coleção com muitos documentos. Por outro lado, o filtro pode levar à construção de classificadores com a eficácia aquém da desejada, uma vez que os filtros não se relacionam diretamente com um algoritmo de classificação.

Neste trabalho foi adotado o filtro, mais especificamente, o ganho de informação (do inglês *Infogain*) [102], uma medida estatística simples e bastante utilizada na classificação de documentos que avalia cada característica isoladamente utilizando um *ranking* de características para selecionar um subconjunto de características [52].

Ganho de Informação

O ganho de informação, também conhecido como informação mútua média (do inglês *average mutual information*) [135] ou perda esperada na entropia (do inglês *expected entropy loss*) [46] é uma medida baseada na avaliação da capacidade de uma característica separar documentos em categorias. O ganho de informação pode ser utilizado para avaliar cada característica individualmente e aquelas com o menor ganho de informação que um limiar l são removidas desse conjunto.

Para avaliar a capacidade de uma característica, o ganho de informação mensura a redução esperada na pureza de uma coleção de documentos de treino (redução da entropia) causada pela divisão dos documentos de treino de acordo com uma característica.

A entropia do conjunto de treino Tr é calculada pela Equação 3-7:

$$Entropy(Tr) \equiv -\sum_{i-1}^{|C|} \Pr(c_j) log \Pr(c_j) \tag{3-7}$$

onde C é o conjunto de categorias e $Pr(c_j)$ é a proporção de documentos de treino na categoria c_j sobre o total de documentos de treino [86].

O ganho de informação da característica t é calculado pela Equação 3-8:

$$IG(t) \equiv Entropy(Tr) - \sum_{v \in (t,\bar{t})} \frac{D_v}{|Tr|} Entropy(Tr) \tag{3-8}$$

onde Tr é um conjunto de treino e D_t é um subconjunto de documentos de treino que possuem o termo t e $D_{\bar{t}}$ é um subconjunto de documentos de treino que não possuem o termo t.

Substituindo a Equação 3-7 na Equação 3-8, o ganho de informação da característica t é calculado por [23]:

$$\begin{aligned} IG(t) &= -\sum_{i-1}^{K} Pr(c_j) log \Pr(c_j) \\ &+ \Pr(t) \sum_{i-1}^{K} \Pr(c_j|t) log \Pr(c_j|t) \\ &+ \Pr(\bar{t}) \sum_{i-1}^{K} \Pr(c_j|\bar{t}) log \Pr(c_j|\bar{t}) \end{aligned}$$

que é equivalente a [23]:

$$\begin{aligned} IG(t) = \Pr(t) \sum_{i-1}^{K} \Pr(c_j|t) log \frac{\Pr(c_j|t)}{\Pr(c_j)} \\ + \Pr(\bar{t}) \sum_{i-1}^{K} \Pr(c_j|\bar{t}) log \frac{\Pr(c_j|\bar{t})}{\Pr(c_j)} \end{aligned}$$

onde $Pr(t)$ é a proporção de documentos em que a característica t está presente, $Pr(\bar{t})$ é a proporção de documentos em que a característica t está ausente, $Pr(c_j|t)$ é a probabilidade condicional da categoria c_j, dada a característica t e $Pr(c_j|\bar{t})$ é a probabilidade condicional da categoria c_j, dada a ausência da característica t.

Os cálculos incluem as estimativas das probabilidades condicionais de uma categoria, dados uma característica e os cálculos da entropia. As estimativas das probabilidades possuem complexidade de $O(|Tr|)$ e os cálculos de entropia possuem complexidade de $O(|T| \times |C|)$, onde T corresponde ao conjunto de termos distintos da coleção Tr [135].

3.3 Classificação de documentos

A classificação automática de textos (CAT) é uma disciplina que surgiu na década de 60 e se tornou parte da área de sistemas de informação no começo da década de 90 [112]. Essa disciplina tem sido aplicada em muitos contextos, desde a indexação de documentos baseada em vocabulários controlados, filtragem de documentos, geração automática de metadados, construção de diretórios hierárquicos de documentos e outros cenários que precisam organizar, selecionar ou adaptar documentos.

Dados a coleção de documentos $D = \{d_1, d_2, \ldots, d_{|D|}\}$ e o conjunto de categorias ou classes $C = \{c_1, c_2, \ldots, c_{|C|}\}$, a classificação de documentos é a atividade de atribuir um valor booleano (0 ou 1) para cada par $(d_i, c_j) \in D \times C$. Quando $(d_i, c_j) = 1$, o documento d_i está rotulado com a categoria c_j e quando $(d_i, c_j) = 0$, o documento d_i não está rotulado com a categoria c_j.

Até o final da década de 80, a abordagem mais popular para a classificação de documentos foi a engenharia do conhecimento [112]. Essa abordagem consiste na construção de um sistema especialista que é capaz de decidir a categoria de determinado documento. Nesse sistema, um conjunto de regras lógicas são definidas manualmente por um engenheiro do conhecimento, com a ajuda de um especialista no domínio. As regras possuíam o seguinte formato:

se (*expressão*) **então** *categoria*

Uma grande desvantagem dos sistemas baseados na engenharia do conhecimento é o gargalo na aquisição do conhecimento (do inglês *knowledge acquisition bottleneck*). Resumidamente, esse gargalo pode ser descrito da seguinte forma [125]:

- Os canais existentes para converter o conhecimento organizacional a partir das suas fontes (especialistas ou documentos) são relativamente limitados.
- A demora na aquisição do conhecimento é normalmente acompanhada por um atraso entre o momento em que o conhecimento (ou os dados subjacentes) é criado e o momento em que esse conhecimento torna-se disponível para ser compartilhado.
- Os especialistas podem cometer erros. Ao criar uma regra incorreta na base de conhecimento, os sistemas baseados na engenharia do conhecimento podem fazer relações espúrias. Além disso, a manutenção dessas regras pode introduzir regras incoerentes na base de conhecimento.
- À medida que a base de conhecimento cresce, aumenta a necessidade de manter as regras dessa base. A manutenção incorreta das regras já existentes na base pode tornar a manutenção futura cada vez mais difícil.

A partir da década de 90, a abordagem para a classificação de documentos utilizando sistemas especialistas foi perdendo espaço para a abordagem de aprendizagem

de máquina (AM). Nessa abordagem, um processo indutivo constrói um classificador para uma categoria $c_j \in C$ a partir da observação das características de um conjunto de documentos classificados manualmente sobre c_j ou $\overline{c_j}$ por um especialista no domínio.

As vantagens da AM sobre os SE são várias. A mais importante delas está nos esforços de engenharia. Enquanto na primeira abordagem os esforços são para a construção de um construtor de classificadores (chamado aprendiz), na segunda abordagem os esforços são para a construção de classificadores. Dessa forma, o trabalho que era realizado por especialistas tem sido substituído por classificadores automáticos. Além disso, a AM possui alta eficácia na classificação, possibilita economizar tempo, custo, velocidade e minimiza problemas inerentes da subjetividade humana [64].

A abordagem de AM para a classificação de documentos tem se tornado atrativa principalmente devido ao grande número de aplicações da Internet que utilizam a tarefa de classificação de documentos. Entre essas aplicações estão: a identificação de *spams* no correio eletrônico para facilitar a exclusão dessas mensagens [7] [33] [111], a classificação hierárquica de documentos na *Web* [35] e a organização de documentos de bibliotecas digitais em tópicos [27].

É fundamental a existência de documentos pré-classificados por especialistas na abordagem de AM. Por exemplo, uma organização que deseja aplicar a atividade de classificação automática de documentos internamente precisa inicialmente realizar a classificação manual de alguns documentos para posteriormente classificar novos documentos automaticamente.

3.3.1 Classificação automática de documentos

Para construir um classificador automático de documentos é necessário, inicialmente, um *corpus* inicial $\Omega = \{d_1, d_2, \ldots, d_{|\Omega|}\} \subset D$ de documentos pré-classificados manualmente por um especialista em determinadas categorias $C = \{c_1, c_2, \ldots, c_{|C|}\}$. A partir dessa atividade, é gerado o mapeamento $\Phi : D \times C \rightarrow \{-1, 1\}$ para todo par $(d_i, c_j) \in \Omega \times C$, onde -1 indica que $d_i \neq c_j$ e 1 indica que $d_i = c_j$, $\forall d_i \in \Omega$ e $\forall c_j \in C$

A classificação automática de documentos consiste no processo de construção do modelo, hipótese ou função $\Psi : D \times C \rightarrow \{-1, 1\}$, tal que ao final desse processo, a maior quantidade possível de valores das funções Ψ e Φ coincidam [112]. Para construir e avaliar o desempenho do classificador Ψ, a abordagem de aprendizagem supervisionada realiza normalmente duas etapas: treino e teste.

A etapa de treino consiste em utilizar algum algoritmo de aprendizagem para construir a função $\Psi : D \times C \rightarrow \{-1, 1\}$ a partir das características dos documentos do conjunto $Tr = \{d_1, d_2, \ldots, d_{|Tr|}\} \subset \Omega$, chamado de conjunto de treino. Ao final dessa etapa, o classificador $\Psi : D \times C \rightarrow \{-1, 1\}$ é construído.

Para avaliar o desempenho do classificador Ψ, as características de cada documento d_i do conjunto de teste $Te = \{d_1, d_2, \ldots, d_{|Te|}\} \subset \Omega$, tal que $Te \cap Tr = \emptyset$, são enviadas para o classificador Ψ que infere a categoria do documento d_i de acordo com as características aprendidas durante a etapa de treino. Ao final dessa etapa, o valor de cada par $(d_i, c_j) \in Te \times C$ é comparado com a função $\Phi(d_i, c_j)$ para avaliar o desempenho do classificador (veja os métodos de avaliação de classificados na Seção 3.3.5).

3.3.2 Formas de classificação

Algumas aplicações impõem restrições para a tarefa de classificação de documentos. Uma dessas restrições é a quantidade de categorias que um documento pode possuir. Nos casos em que exatamente uma categoria deve ser atribuída para cada documento $d_i \in \Omega$, a tarefa de classificação é chamada de classificação objetiva.

A classificação binária é um caso especial da classificação objetiva, nesse caso, cada documento $d_i \in \Omega$ deve pertencer a categoria c_j ou a $\overline{c_j}$. Por exemplo, a classificação de mensagens eletrônicas em desejáveis ou indesejáveis (*spams*).

A classificação multi rótulo ocorre quanto quando qualquer quantidade de categorias (entre 0 e $|\Omega|$) pode ser atribuída para cada documento $d_i \in \Omega$ [112].

Quanto ao estilo, a classificação de documentos pode ser centrada no texto ou na categoria [112]. No primeiro estilo, dado um documento $d_i \in \Omega$, deseja-se obter todas as categorias $c_j \in C$ atribuídas ao documento d_i. Por outro lado, na categorização centrada na categoria, dada uma categoria $c_j \in C$, deseja-se encontrar todos os documentos $d_i \in \Omega$ em que a categoria c_j é atribuída. A maioria das técnicas de classificação pode ser aplicada para ambos estilos.

A classificação de documentos também pode ser discreta ou contínua [112]. A classificação discreta exige uma decisão 1 ou 0 para cada par (d_i, c_j). Na classificação contínua não existe essa exigência. Por exemplo, dado $d_i \in \Omega$, as categorias em $C = \{c_1, c_2, \ldots, c_{|C|}\}$ poderiam ser ordenadas de acordo com o grau de confiança da classificação do documento d_i sobre c_j. A classificação contínua pode auxiliar na classificação manual de documentos.

3.3.3 *k*-vizinhos mais próximos

O método *k*-vizinhos mais próximos (kNN, do inglês *k-Nearest Neighbors*) é considerado um dos métodos de classificação mais antigos e simples [28]. Apesar da sua simplicidade, esse método tem alcançado bom desempenho em diferentes cenários [16] [115].

O método kNN é um "aprendiz preguiçoso" (do inglês *lazy learning*) [2]. Um aprendiz preguiçoso simplesmente armazena os documentos de treino e realiza uma única etapa para classificar documentos.

Dado um documento de teste d, para classificá-lo o método kNN tradicionalmente realiza as seguintes atividades:

1. A distância entre o documento d e cada um dos documentos de treino é calculada utilizando alguma medida de similaridade entre documentos, tal como a medida do cosseno (Seção 3.1.3).
2. Os k documentos de treino mais próximos, isto é, mais similares ao documento d são selecionados.
3. O documento d é classificado em determinada categoria de acordo com algum critério de agrupamento das categorias dos k documentos de treino selecionados na etapa anterior.

Ao realizar as atividades descritas anteriormente, surgem duas questões importantes que influenciam no desempenho do método kNN [113]:

- Qual o critério de similaridade será utilizado?
- Como as categorias dos k vizinhos mais próximos serão agrupadas?

Em relação à primeira questão, o critério de similaridade é um aspecto utilizado pelo método kNN que possui grande influência no desempenho desse método [113]. Esse critério é composto pela medida de similaridade, ou função de distância e pelo critério de seleção dos vizinhos. O critério de seleção determina a forma de escolha dos k vizinhos de um documento. Por exemplo, selecionar os k documentos de treino mais próximos do documento de teste d para um valor de k fixo é um critério de seleção tradicionalmente adotado pelo método kNN.

Em relação à segunda questão, uma das formas mais comuns de agrupar as categorias dos k vizinhos mais próximos é atribuir para o documento d_i a categoria com maior pontuação de acordo com a Equação 3-9 [131].

$$pnt(d_i, c_j) = \sum_{d_t \in N_k(d_i)} sim(d_i, d_t) \times ver(c_j, d_t) \tag{3-9}$$

onde $N_k(d_i)$ são os k documentos de treino mais próximos de d_i, $sim(d_i, d_t)$ é o valor da similaridade entre d_i e d_t e $ver(c_j, d_t)$ é uma função que retorna 1, caso o documento de treino d_t pertença a categoria c_j e 0, caso contrário.

Outra decisão importante que tem bastante influência no desempenho do método kNN é a definição do valor mais adequado para k. De um modo em geral, quando o conjunto de treino possui muitos elementos classificados incorretamente por um especialista

(ruídos) é preferível utilizar o método kNN com $k = 1$, caso contrário, com $k > 1$. Entretanto, para determinar o valor de k que o método kNN possui melhor desempenho é necessário a realização de experimentos (tentativa e erro) escolhendo diferentes valores para k.

Para calcular a distância entre um documento de teste d_i e os seus k vizinhos mais próximos, normalmente é necessário calcular todas as distâncias entre os documentos de teste e de treino. Esse problema pode ser visto como um problema de busca e caso seja utilizada a estratégia 'força bruta sem ordenação' em sua solução, cada documento de teste realiza $k \times |Tr|$ operações - $O(|Tr|)$ [26]. Caso haja muitos documentos de treino, o método kNN pode se tornar computacionalmente inviável.

Existem diversos trabalhos relacionados à análise do desempenho do método kNN em conjuntos de treino com tamanho finito e, embora ainda não exista um limite de erro independente da distribuição utilizada, é possível caracterizar o risco de um classificador kNN baseado nas propriedades do espaço de entrada e na distribuição dos dados [113]. Além disso, apesar da falta de garantias teóricas, o classificador kNN possui na prática um desempenho muito bom.

Para exemplificar as atividades realizadas pelo método kNN, foi construída a coleção de documentos *MSG*, que corresponde a mensagens eletrônicas classificadas em duas categorias: SPAN e EMAIL. A Tabela 3.2 mostra a representação conjunto de palavras (BOW, do inglês *bag of words*) da coleção *MSG*.

MSG	docID	categoria	t_1	t_2	t_3	t_4
d_0	0	EMAIL	2	1	0	0
d_1	1	SPAM	0	1	1	2
d_2	2	EMAIL	3	0	7	0
d_3	3	SPAM	5	0	3	2
d_4	4	SPAM	2	1	0	1
d_5	5	EMAIL	3	2	2	0

Tabela 3.2: *Matriz de frequência absoluta de ocorrência de termos da coleção MSG*

A coleção MSG, conforme mostrado na Tabela 3.2, possui seis documentos que foram divididos em dois conjuntos, o conjunto de teste $Te = \{d_0, d_1\} \subset MSG$ e o conjunto de treino $Tr = \{d_2, d_3, d_4, d_5\} \subset MSG$. Os documentos do conjunto Tr foram classificados manualmente em duas categorias: SPAM e EMAIL. Os números que aparecem nas quatro últimas colunas dessa tabela, correspondem à frequência que cada termo t_i aparece em determinado documento da coleção.

Dada a coleção MSG, $k = 3$ e adotando a medida do cosseno (Seção 3.1.3), o método kNN calcula a similaridade entre cada documento do conjunto Te com cada do-

cumento do conjunto Tr para selecionar os k vizinhos mais próximos de cada documento do conjunto Te.

A Figura 3.3 ilustra os três documentos de treino mais próximos dos documentos do conjunto de teste $Te = \{d_0, d_1\}$ e seus respectivos valores de similaridade, calculados utilizando o cosseno como medida.

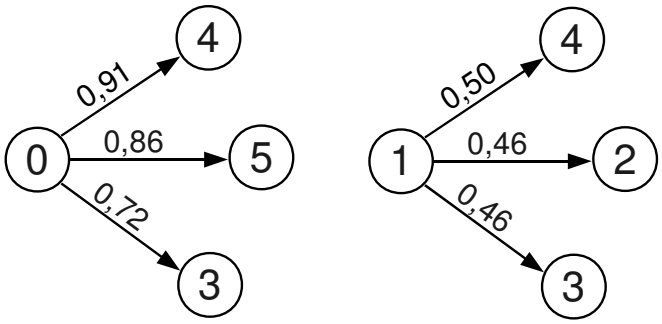

Figura 3.3: *Documentos de treino mais próximos do documento de teste d_i*

Para classificar os documentos do conjunto de teste Te, o método kNN agrupa as categorias dos k documentos de treino mais próximos de cada documento $d \in Te$ de acordo com a pontuação $pnt(d_i, c_j)$ (conforme a Equação 3-9). Para a categoria SPAM, a pontuação do documento d_0 foi $1,63$ e para o documento d_1 foi $0,96$ e para a categoria EMAIL, a pontuação do documento d_0 foi $0,86$ e para o documento d_1 foi $0,46$. Conforme esses resultados, os documentos d_0 e d_1 foram classificados na categoria SPAM.

3.3.4 Máquinas de vetores suporte

O método Máquinas de Vetores Suporte (SVM, do inglês *Support Vector Machines*) [121] é um dos métodos mais eficazes e utilizados na classificação de objetos. Esse método é um "aprendiz ansioso" (do inglês *eager learning*), pois gera um modelo explícito e estima as distribuições de probabilidades das categorias na etapa de treino e posteriormente realiza a etapa de teste.

Algumas características justificam a utilização do método SVM na classificação de documentos, tais como: boa capacidade de generalização, robustez em alta dimensionalidade, capacidade de lidar com dados ruidosos e uma base teórica matemática e estatística solidamente fundamentada [21] [110] [121].

Fundamentação teórica

Dados o conjunto de treino $Tr = \{d_i, c_j\}_{i=1}^{|Tr|} \subset D$ e o conjunto de teste $Te = \{d_i, c_j\}_{i=1}^{|Te|} \subset D$, tal que $d_i \in R^{|T|}$, onde D é uma coleção de documentos, T é o conjunto de termos distintos da coleção D e $c_j \in \{-1, 1\}$. Cada documento da coleção D é

representado como um ponto d_i no espaço euclidiano $R^{|T|}$ e gerado de forma independente e identicamente distribuída em relação a uma probabilidade desconhecida $Pr(d_i, c_j)$ [110].

O objetivo do processo de aprendizagem estatística, assim como da SVM, é alcançar uma função indicadora α que minimize a complexidade e o erro de um classificador por meio das relações extraídas do conjunto de treino Tr.

No processo de escolha da melhor função que se ajusta ao conjunto de treino Tr, é necessária a criação de uma medida de discrepância ou perda, que sinaliza ao classificador quando houve erros ou acertos durante a aprendizagem [122]. A função de perda normalmente empregada em problemas de classificação binária é a seguinte:

$$L(f(d_i,\alpha),c_j) = \begin{cases} 1 \text{ se } f(d_i,\alpha) \neq c_j \\ 0 \text{ se } f(d_i,\alpha) = c_j \end{cases} \tag{3-10}$$

onde α é uma função indicadora, $f(d_i,\alpha)$ é a saída do classificador cuja entrada é d_i.

O risco funcional ou esperado mensura a taxa de erro de um classificador para os documentos do conjunto de teste Te. Essa medida possibilita verificar a capacidade de generalização de um classificador. O risco funcional é calculado pela Equação 3-11.

$$R(\alpha) = \int \frac{1}{2} L(f(d_i,\alpha),c_j) dPr(d_i,c_j) \tag{3-11}$$

onde $Pr(d_i,c_j)$ é a probabilidade do documento d_i pertencer a categoria c_j.

Normalmente, antes da etapa de treino, não é possível determinar a distribuição de probabilidade $Pr(d_i,c_j)$ de um classificador. Isto impossibilita o cálculo do risco funcional para os documentos do conjunto de teste Te.

Para alcançar o objetivo do processo de aprendizagem estatística, visto que não é possível calcular o risco funcional, recorre-se à minimização do risco empírico (ERM, do inglês *empirical risk minimization*) [21]. A ERM, calculada pela Equação 3-12, possibilita mensurar a taxa de erro de um classificador para os documentos do conjunto de treino Tr.

$$R_{emp}(\alpha) = \frac{1}{2|Tr|} \sum_{i=1}^{|Tr|} L(f(d_i,\alpha),c_j) \tag{3-12}$$

Entretanto, minimizar o risco empírico nem sempre é suficiente para obter um classificador com bom desempenho, uma vez que esse risco não considera a complexidade das funções indicadoras. Para cada função indicadora α existe uma dimensão Vapnik-Chervonenkis (VC) com capacidade adequada.

Dimensão VC e minimização do risco estrutural

A dimensão VC pode ser compreendida como a capacidade de aprendizado de uma classe de funções F que classifica corretamente a maior quantidade de documentos de treino [21]. Para funções lineares no espaço R^2 (retas) essa capacidade é 3 para qualquer padrão de rotulação binária que as amostras possam admitir, conforme apresentado na Figura 3.4. Para funções lineares no espaço R^n, com $n \geq 2$, a dimensão VC é $n+1$.

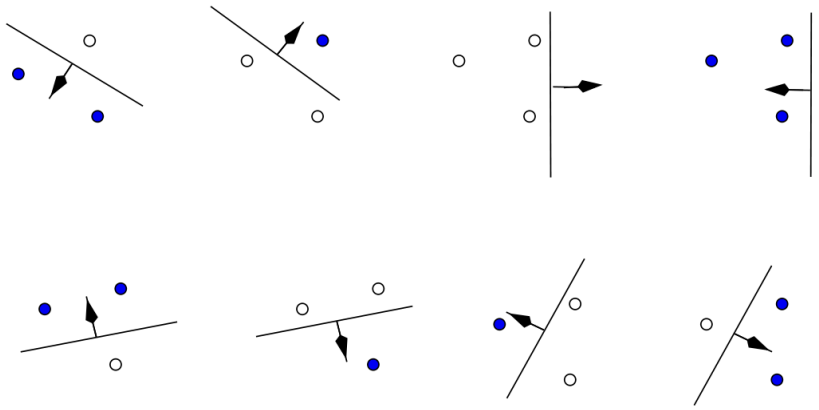

Figura 3.4: *Possíveis separações de três pontos por uma reta [21]*

Além das funções lineares, a dimensão VC do conjunto de funções F pode possuir diferentes capacidades tais como: exponenciais e polinomiais. Quanto maior essa capacidade, maior a complexidade das funções indicadoras que podem ser induzidas a partir do conjunto F e maior a tendência ao sobreajuste (do inglês *overfitting*) e quanto menor essa capacidade, maior a restrição para classificar novos documentos [29].

O risco esperado de um classificador pode ser minimizado pela escolha adequada do algoritmo de aprendizado, de uma função indicadora α que minimize o risco empírico e que pertença a uma classe de funções F com baixa dimensão VC. Esses requisitos definem um princípio de indução conhecido como minimização do risco estrutural (SRM, do inglês *strutural risk minimization*) [122].

A SRM consiste em dividir o conjunto de funções F em subconjuntos de funções com dimensão VC crescente [120]. Esses subconjuntos são definidos como estruturas dadas por [110]:

$$F_1 \subset F_2 \subset \ldots \subset F_k \subset F \tag{3-13}$$

onde h_k é a dimensão VC de cada subconjunto F_k com a propriedade $h_k \leq h_{k+1}$.

O princípio da SRM consiste em treinar uma série de classificadores, um para cada subconjunto F_k, com o objetivo de minimizar o risco empírico [21]. A função indicadora α escolhida será aquela cuja soma do risco empírico e da capacidade da função for a menor entre os subconjuntos de funções.

Para um subconjunto particular F_k, seja $\hat{f}_k$ o classificador com o menor risco empírico. À medida que k cresce, o risco empírico de $\hat{f}_k$ diminui e aumenta a complexidade das funções. Entretanto, em determinado momento se obtém um valor ótimo para k, em que a soma do risco empírico e da razão $\frac{h}{|Tr|}$ seja a menor, onde h é a dimensão VC.

SVMs lineares

O método SVM foi inicialmente definido para a classificação de padrões linearmente separáveis. Em outras palavras, padrões cujas categorias possam ser separadas por um hiperplano.

Dado o conjunto de treino $Tr = \{d_i, c_j\}_{i=1}^{|Tr|} \subset D$, onde $c_j \in \{\text{círculo, quadrado}\}$. O objetivo do método SVM é construir um hiperplano como superfície de decisão, que separe as categorias distintas do conjunto Tr com a maior margem de separação possível. A Figura 3.5 ilustra o exemplo desse hiperplano, representado por uma linha tracejada.

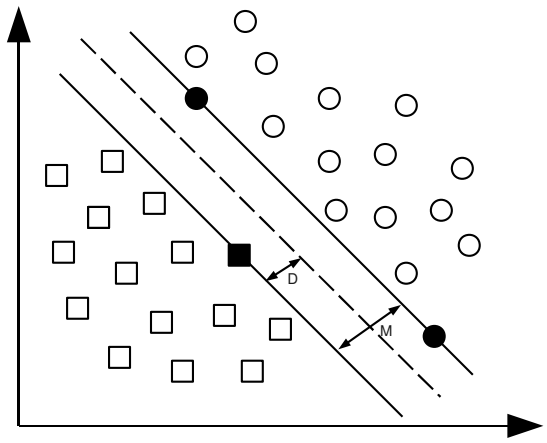

Figura 3.5: *Hiperplano separador com maior margem de separação entre duas categorias distintas*

O hiperplano separador do conjunto Tr é calculado pela Equação 3-14.

$$(\overrightarrow{w} \cdot \overrightarrow{x}) + b = 0, \tag{3-14}$$

onde $\overrightarrow{x}$ é um ponto arbitrário que representa um padrão a ser classificado, o vetor $\overrightarrow{w}$ define a direção do hiperplano perpendicular ao ponto $\overrightarrow{x}$ e o termo b possibilita deslocar o hiperplano paralelamente a esse ponto.

Para determinar a categoria que um determinado padrão $\overrightarrow{x}$ pertence, é necessário verificar a sua posição relativa ao hiperplano através da seguinte relação:

$$y_i = \begin{cases} +1 \text{ (círculo) se } (\overrightarrow{w} \cdot \overrightarrow{x}) + b \geq 0 \\ -1 \text{ (quadrado) se } (\overrightarrow{w} \cdot \overrightarrow{x}) + b < 0 \end{cases} \tag{3-15}$$

A partir de determinados elementos do conjunto Tr, chamados de vetores suporte (representados na Figura 3.5 por pontos escuros), é possível encontrar a maior margem de separação entre duas categorias distintas. Os vetores suporte representam os pontos mais próximos do hiperplano separador e são calculados pela Equação 3-16.

$$(\overrightarrow{w} \cdot \overrightarrow{x}) + b = \{+1, -1\} \tag{3-16}$$

Após a obtenção dos vetores suporte, é possível encontrar a margem de separação (representada por M na Figura 3.5). Essa margem é calculada pela Equação 3-17 e representa a distância entre dois vetores suporte pertencentes a categorias diferentes.

$$M = \frac{2}{\|\overrightarrow{w}\|} \tag{3-17}$$

A partir do cálculo da margem de separação, é possível obter a distância entre um vetor suporte e um hiperplano. A distância entre o vetor suporte $\overrightarrow{x}$ e o hiperplano D (Figura 3.5) é calculada pela Equação 3-18.

$$D = \frac{M}{2} = \frac{|(\overrightarrow{w} \cdot \overrightarrow{x}) + b|}{\|\overrightarrow{w}\|} = \frac{1}{\overrightarrow{w}} \tag{3-18}$$

Em muitos casos, os padrões não são linearmente separáveis. Nesses casos, os padrões de entradas (espaço de entrada) são transformados em um vetor de características com alta dimensionalidade, cujo objetivo é separar linearmente as características no espaço através do uso de funções não lineares especiais chamadas de *Kernel*. O *Kernel* possibilita a construção de um hiperplano de separação ótimo no espaço de características sem considerar explicitamente o próprio espaço de características [55].

3.3.5 Avaliação dos classificadores

Os métodos de avaliação são normalmente utilizados para avaliar o desempenho de um classificador de documentos. Os métodos a seguir, são os mais conhecidos e utilizados:

- ***Holdout***: divide aleatoriamente uma coleção Ω de documentos pré-classificados em dois conjuntos (conjunto de treino e teste), sendo que o percentual de p documentos constitui o conjunto de treino e $1-p$ documentos constitui o conjunto de teste. Normalmente $p = 33\%$ de documentos são utilizados para teste e o restante para treino. O problema dessa abordagem é que os documentos de teste selecionados podem não ser representativos. Por exemplo: uma determinada categoria pode estar ausente no conjunto de teste.
- **Validação cruzada com q partições** (do inglês *q-Fold Cross Validation*): consiste em construir q classificadores diferentes $(\Psi_1, \Psi_2, \ldots, \Psi_q)$ a partir da divisão de

uma coleção de documentos Ω em q conjuntos disjuntos $(Te_1, Te_2, \ldots, Te_q)$ com aproximadamente $|\Omega|/q$ documentos em cada conjunto [86]. Após essa divisão, o classificador Ψ_i é treinado utilizando o conjunto de treino $Tr = \Omega - Te_i$ e avaliado utilizando o conjunto de teste Te_i. Ao final da avaliação dos q classificadores, a média das medidas de avaliação dos q classificadores é calculada para a avaliação final da classificação. Quando $q = |\Omega|$, a validação cruzada é denominada 'deixe um de fora' (do inglês *leave-one-out*). Essa validação é computacionalmente custosa e frequentemente utilizada quando a coleção de documentos é pequena. Outro tipo de validação cruzada com q partições, que tem se tornado um padrão na avaliação dos classificadores de documentos, é a validação cruzada com 10 partições (do inglês 10-*Fold Cross Validation*)

- **Validação cruzada estratificada com q partições** (do inglês *Stratified q-Fold Cross Validation*): similar à validação cruzada com q partições, sendo que ao dividir a coleção de documentos Ω em q conjuntos, a proporção de documentos em cada uma das categorias é considerada na constituição dos conjuntos. Por exemplo: uma coleção de documentos que possui duas categorias, sendo que a distribuição de documentos nessas categorias é de 20% e 80%. Ao dividir essa coleção em q conjuntos, cada um deles também deverá possuir aproximadamente essa mesma proporção de categorias.
- ***Bootstrap***: dada uma coleção de documentos Ω de tamanho n, esse método consiste em selecionar n documentos com reposição da amostra e construir uma nova coleção de documentos Ω' de tamanho n. Os documentos da coleção Ω' são utilizados na constituição do conjunto de treino e os documentos da coleção Ω, que não fazem parte do conjunto de treino, são utilizados na constituição do conjunto de teste. Esse processo é repetido q vezes, utilizando diferentes conjuntos com reposição da amostra. O resultado final é a média dos resultados obtidos nas q vezes que o processo foi repetido. Esse método é considerado um dos melhores métodos de avaliação quando a coleção de documentos é pequena.

Para mensurar o desempenho de um classificador, normalmente são utilizadas medidas que podem ser compreendidas a partir da análise da tabela de contingência. Essa tabela possibilita registrar e analisar o relacionamento entre variáveis.

Dada a categoria A, a tabela de contingência, conforme a Tabela 3.3, mostra as possibilidades de respostas de um classificador em relação à categoria A e as demais categorias, representadas por $\overline{A}$.

Na Tabela 3.3, $F\oplus_A$ (falso positivo sobre a categoria A - erros de concessão) corresponde à quantidade de documentos de teste que foram classificados na categoria A mas que deveriam ser classificados em outra categoria ($\overline{A}$), $F\ominus_A$ (falso negativo sobre a categoria A - erros de omissão) corresponde à quantidade de documentos de teste que

categoria A	A correto	$\overline{A}$ correto
decide A	$V\oplus_A$	$F\oplus_A$
decide $\overline{A}$	$F\ominus_A$	$V\ominus_A$

Tabela 3.3: *Tabela de contingência para a categoria A*

deveriam ser classificados na categoria A mas que foram classificados na categoria $\overline{A}$, $V\oplus_A$ (verdadeiro positivo sobre a categoria A) corresponde à quantidade de documentos de teste classificados na categoria A e que foram corretamente classificados e $V\ominus_A$ (verdadeiro negativo sobre a categoria A) corresponde à quantidade de documentos de teste classificados como $\overline{A}$ e que foram corretamente classificados.

Duas medidas básicas para avaliar o desempenho dos classificadores são: a precisão (do inglês *precision*) e a cobertura (do inglês *recall*). Essas medidas são definidas com base nas variáveis da tabela de contingência (Tabela 3.3) e são calculadas respectivamente pelas Equações 3-19 e 3-20.

$$p(c) = \frac{V\oplus_c}{V\oplus_c + F\oplus_c} \tag{3-19}$$

$$r(c) = \frac{V\oplus_c}{V\oplus_c + F\ominus_c} \tag{3-20}$$

onde c é a categoria do conjunto C.

Quando a quantidade de categorias é grande, para evitar muitos valores ao calcular a precisão e a cobertura, é conveniente combinar o cálculo dessas medidas em uma única medida denominada Métrica-F (do inglês *F-measure*), calculada pela Equação 3-21 [134]:

$$F_\alpha(c) = \frac{(\alpha^2+1)p(c)r(c)}{\alpha^2 p(c) + r(c)} \tag{3-21}$$

A Métrica-F possibilita atribuir diferentes pesos para a precisão e a cobertura. Quando $\alpha = 0$, apenas a precisão é considerada, quando $\alpha = \sigma$, apenas a cobertura é considerada, quando $\alpha = 0.5$, a cobertura possui a metade da importância da precisão e quando $\alpha = 1$ as medidas possuem a mesma importância [112]. Essa última atribuição, conhecida como F_1, é calculada pela Equação 3-22:

$$F_1(c) = \frac{2p(c)r(c)}{p(c) + r(c)} \tag{3-22}$$

A F_1 considera o desempenho de um classificador em relação a uma categoria. Para considerar todas as categorias, um único valor para F_1 pode ser derivado. Duas médias são normalmente utilizadas com esse propósito: $macroF_1$ e $microF_1$ [134].

A $macroF_1$ mensura o desempenho do classificador de acordo com a média dos resultados obtidos em F_1 para cada uma das categorias do conjunto de teste [134]. A $macroF_1$ é calculada pela Equação 3-23:

$$macroF_1 = \frac{\sum_{c=1}^{|C|} F_1(c)}{|C|} \tag{3-23}$$

A $microF_1$ mensura o desempenho do classificador de acordo com a precisão e a cobertura globais. A precisão global e a cobertura global são respectivamente calculadas conforme as Equações 3-24 e 3-25:

$$p_g = \frac{\sum_{c=1}^{|C|} V \oplus_c}{\sum_{c=1}^{|C|} (V \oplus_c + F \oplus_c)} \tag{3-24}$$

$$r_g = \frac{\sum_{c=1}^{|C|} V \oplus_c}{\sum_{c=1}^{|C|} (V \oplus_c + F \ominus_c)} \tag{3-25}$$

Desta forma, a $microF_1$ é calculada pela Equação 3-26:

$$microF_1 = \frac{2 p_g r_g}{p_g + r_g} \tag{3-26}$$

Para comparar dois resultados, o ganho é uma medida bastante utilizada. O ganho de b em relação a a é calculado pela Equação 3-27.

$$ganho(b,a) = \frac{AVG(b) - AVG(a)}{AVG(a)} \tag{3-27}$$

onde AVG é o resultado obtido em $microF_1$ ou $macroF_1$.

Exemplificando, dado um conjunto de teste T, tal que $|T| = 10$, 6 documentos desse conjunto pertencem a categoria A e 4 documentos desse conjunto pertencem a categoria B. Considerando que um classificador atribuiu 5 documentos como pertencentes a categoria A, 5 como pertencente a categoria B e que tenha acertado na atribuição de 4 documentos da categoria A. Conforme esse cenário, os exemplos a seguir demonstram os cálculos da precisão, cobertura, F_1, $macroF_1$ e $microF_1$.

$p(A) = \frac{4}{4+1} = 0,80$ $\quad$ $p(B) = \frac{3}{3+2} = 0,60$

$r(A) = \frac{4}{4+2} = 0,66$ $\quad$ $r(B) = \frac{3}{3+1} = 0,75$

$F_1(A) = \frac{2(0,80)(0,66)}{(0,80)+(0,66)} = 0,72$ $\quad$ $F_1(B) = \frac{2(0,60)(0,75)}{(0,60)+(0,75)} = 0,66$

$macroF_1 = \frac{0,72+0,66}{2} = 0,69$ $\quad$ $microF_1 = \frac{2(0,70)(0,70)}{(0,70)+(0,70)} = 0,70$

CAPÍTULO 4

Abordagens Propostas

Este capítulo apresenta as abordagens propostas neste trabalho. Na Seção 4.1, são descritos os dois critérios de seleção propostos para o método kNN (critérios de seleção kINN e kSNN) e as duas variações do método kNN resultantes da adoção desses critérios (métodos kINN e kSNN). Na Seção 4.2, é descrita a abordagem proposta para gerar características em documentos. A leitura do Capítulo 3 é fundamental para compreender os conceitos relacionados às abordagens propostas neste capítulo.

Para descrever os critérios de seleção e as variações do método kNN propostos, serão utilizadas as seguintes notações: Ω representa uma coleção de documentos, $Tr = \{a_1, a_2, \ldots, a_{|Tr|}\}$ representa o conjunto de treino da coleção Ω, $Te = \{b_1, b_2, \ldots, b_{|Te|}\}$ representa o conjunto de teste da coleção Ω e $C = \{c_1, c_2, \ldots, c_{|C|}\}$ representa o conjunto de categorias da coleção Ω.

4.1 Variações do método kNN

A diferença fundamental entre o método kNN e as duas variações desse método propostas neste trabalho está no critério de seleção. O critério de seleção é uma das etapas realizadas pelo método kNN na classificação de documentos. O Algoritmo 4.1 formaliza as etapas realizadas pelo método kNN.

Algoritmo 4.1: $kNN(Tr, Te, k)$

Entrada: Tr, Te, k;
Saída: $(d_i, c_j) \in Te \times C$;

1 **para cada** $(d \in Te)$ **faça**
2 $\quad Prox[d] \leftarrow SelecionarVizinhos(d, Tr, k)$;
3 $\quad Classificar(d, Prox[d])$;
4 **fim**

Na linha 2 do Algoritmo 4.1, a função $SelecionarVizinhos(d, Tr, k)$ é responsável por computar a similaridade entre o documento d e cada um dos documentos do conjunto

Tr, organizar os documentos do conjunto Tr em ordem decrescente de similaridade com o documento d e selecionar os k documentos do conjunto Tr mais similares ao documento d. Os documentos selecionados são armazenados na estrutura $Prox[d]$. Na linha 3, a função $Classificar(d, Prox[d])$ é responsável por classificar o documento d de acordo com as categorias dos documentos armazenados na estrutura $Prox[d]$.

O critério de seleção adotado pelo método kNN é representado na linha 2 do Algoritmo 4.1 pela função $SelecionarVizinhos(d, Tr, k)$. Dados o conjunto de teste Te, o conjunto de treino Tr e o valor k de vizinhos mais similares, esse critério pode ser especificado da seguinte forma:

- **Critério de seleção kNN**: para cada documento $d \in Te$, computar as similaridades entre d e cada documento do conjunto Tr. Selecionar os k documentos do conjunto Tr mais similares ao documento d.

O critério de seleção kNN tem sido tradicionalmente adotado pelo método kNN. Na tentativa de aumentar a eficácia da classificação automática de textos (CAT), neste trabalho foram propostos dois novos critérios de seleção para substituir esse critério: critério de seleção kINN e critério de seleção kSNN. Os critérios propostos foram baseados na abordagem vizinhos mais similares reversos (RNN, do inglês *Reverse Nearest Neighbor*) e no próprio critério de seleção kNN.

O RNN é uma abordagem proposta para solucionar problemas sobre a influência de determinado ponto sobre um conjunto de objetos. Conforme essa abordagem, dado um ponto $q \in S$, o método $RNN(q, k)$ retorna os elementos do conjunto S que possuem o ponto q entre os seus k vizinhos mais similares [65]. Essa abordagem tem sido aplicada em diferentes áreas, tais como em pesquisas de *marketing*, em sistemas de informação geográfica, em redes de tráfego, em jogos computacionais e na biologia molecular [1] [65].

Por exemplo, uma consulta RNN pode obter um grupo de clientes afetados pela abertura de um novo estabelecimento (uma loja de roupas, por exemplo), para determinar a viabilidade da abertura do estabelecimento, informar potenciais clientes sobre a inauguração do estabelecimento ou enviar mensagens promocionais para esses clientes. Entretanto, apesar da importância da abordagem RNN, não se tem o conhecimento de outros trabalhos que aplicam essa abordagem na classificação automática de textos (CAT).

Os critérios de seleção propostos neste trabalho podem ser especificados da seguinte maneira:

- **Critério de seleção kINN**: para cada documento $p \in Tr$, computar as similaridades entre p e cada documento do conjunto $Tr \cup \{d\}$, tal que $d \in Te$. Se d estiver entre os k documentos mais similares à p, então selecione p.

- **Critério de seleção kSNN**: para cada documento $p \in Tr$, computar as similaridades entre p e cada documento do conjunto $Tr \cup \{d\}$, tal que $d \in Te$. Computar também as similaridades entre d e cada documento do conjunto Tr. Se d estiver entre os k documentos mais similares a p e p estiver entre os k documentos mais similares à d, então selecione p.

A Figura 4.1 ilustra os documentos de treino selecionados pelos critérios de seleção kNN, kINN e kSNN com o valor do parâmetro k igual a 3. Nessa figura, o círculo branco representa o documento de teste d, os círculos cinzas representam os documentos de treino selecionados e os círculos pretos representam os documentos de treino não selecionados pelo critério. Uma aresta direcionada partindo de um documento d_1 e chegando a um documento d_2, indica que d_2 está entre os k documentos mais similares a d_1. Se a aresta que liga o documento d_1 ao documento d_2 é bidirecionada, então d_1 está entre os k documentos mais similares a d_2 e vice-versa.

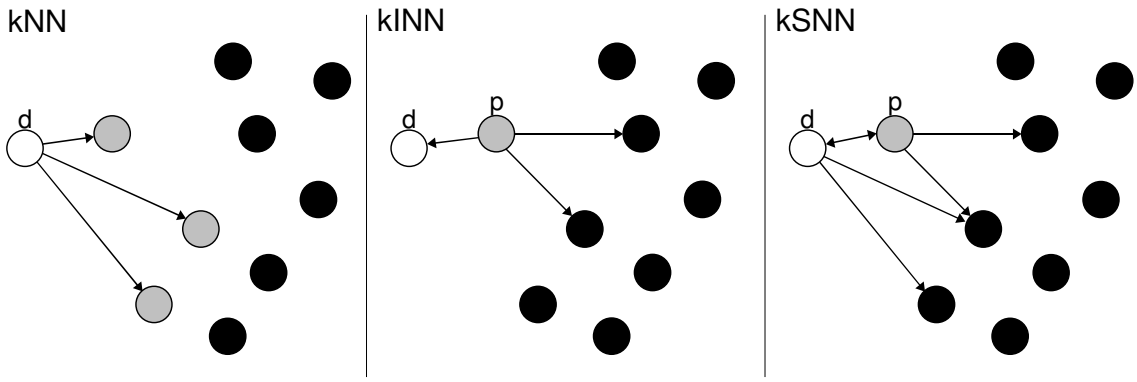

Figura 4.1: *Critérios de seleção kNN, kINN e kRNN com o valor de* $k = 3$

4.1.1 Método kINN

A primeira variação do método kNN proposta consiste em substituir o critério de seleção adotado pelo método kNN (critério de seleção kNN) pelo critério de seleção kINN. Para isso, foi proposta uma nova variação do método kNN, denominada método kNN invertido (kINN, do inglês *k-Inverse Nearest Neighbors*).

Dados o conjunto de teste $Te \subset \Omega$, o conjunto de treino $Tr \subset \Omega$ e o valor k de vizinhos mais similares, o Algoritmo 4.2 formaliza as etapas realizadas pelo método kINN para classificar os documentos do conjunto Te.

Algoritmo 4.2: $kINN(Te, Tr, k)$

Entrada: Tr, Te, k;
Saída: $(d_i, c_j) \in Te \times C$;

```
1  para cada (d ∈ Te) faça
2      para cada (p ∈ Tr) faça
3          Tr_d ← (Tr − {p}) ∪ {d};
4          Prox[p] ← SelecionarVizinhos(p, Tr_d, k);
5          se (d ∈ Prox[p]) então
6              ProxI[d] ← ProxI[d] ∪ {p};
7          fim
8      fim
9      Classificar(d, ProxI[d]);
10 fim
```

Na linha 4 do Algoritmo 4.2, a função $SelecionarVizinhos(p, Tr_d, k)$ é responsável por computar a similaridade entre o documento p e cada um dos documentos do conjunto Tr_d, ordenar os documentos do conjunto Tr_d em ordem decrescente de similaridade com o documento p e selecionar os k documentos do conjunto Tr_d mais similares ao documento p. Os documentos selecionados são armazenados na estrutura $Prox[p]$. Na linha 5, caso o documento d esteja armazenado na estrutura $Prox[p]$, então na linha 6, a estrutura $ProxI[d]$ armazena o documento de treino p. Na linha 9, a função $Classificar(d, ProxI[d])$ é responsável por classificar o documento d de acordo com as categorias dos documentos armazenados na estrutura $ProxI[d]$.

Ao contrário do método kNN, o critério de seleção adotado pelo método kINN não é representado somente pela função $SelecionarVizinhos(p, Tr_d, k)$, mas por todas as instruções da linha 3 até a linha 7 do Algoritmo 4.2.

Para demonstrar a aplicação do método kINN, seja dada a coleção de pontos P, representada pelos pontos $p_1, \ldots, p_5$ no espaço euclidiano R^2, conforme ilustrado na Figura 4.2, tal que p_2 é um ponto de teste, $Tr = \{p_1, p_3, p_4, p_5\} \in P$ é um conjunto de treino e $k = 2$. Ao aplicar o método kINN para classificar o ponto p_2 ocorrem os seguinte passos:

- A similaridade entre cada um dos pontos do conjunto Tr e cada um dos pontos do conjunto $Tr \cup \{p_2\}$ é calculada utilizando o cosseno como medida.
- Os pontos p_1, p_3 e p_4 são identificados como os que possuem o ponto p_2 entre os dois (k) pontos mais similares.
- O ponto p_2 é classificado de acordo com a categoria dos pontos p_1, p_3, p_4.

Figura 4.2: *Distribuição dos pontos da coleção P no espaço euclidiano R^2*

4.1.2 Método kSNN

A segunda variação do método kNN proposta neste trabalho consiste em substituir o critério de seleção kNN pelo critério de seleção kSNN. Para isso, foi proposta uma nova variação do método kNN, denominada método kNN simétrico (kSNN, do inglês *k-Symmetric Nearest Neighbors*).

Dados o conjunto de teste $Te \subset \Omega$, o conjunto de treino $Tr \subset \Omega$ e o valor k de vizinhos mais similares, o Algoritmo 4.3 formaliza as etapas realizadas pelo método kSNN para classificar os documentos do conjunto Te.

Algoritmo 4.3: $kSNN(Te, Tr, k)$

Entrada: Tr, Te, k;
Saída: $(d_i, c_j) \in Te \times C$;

```
1  para cada (d ∈ Te) faça
2      Prox[d] ← SelecionarVizinhos(d, Tr, k);
3      para cada (p ∈ Prox[d]) faça
4          Tr_d ← (Tr − {p}) ∪ {d};
5          Prox[p] ← SelecionarVizinhos(p, Tr_d, k);
6          se (d ∈ Prox[p]) então
7              ProxR[d] ← ProxR[d] ∪ {p};
8          fim
9      fim
10     Classificar(d, ProxR[d]);
11 fim
```

No Algoritmo 4.3, nas linhas 2 e 5, a função *SelecionarVizinhos* possui a mesma responsabilidade das funções utilizadas na linha 2 do Algoritmo 4.1 (método kNN) e na

linha 4 do Algoritmo 4.2 (método kINN). Na linha 10, a função $Classificar(d, ProxR[d])$ é responsável por classificar o documento d de acordo com as categorias dos documentos armazenados na estrutura $ProxR[d]$. O critério de seleção adotado pelo método kSNN é representado no Algoritmo 4.3, pelas linha 2 até a linha 9.

Para demonstrar a aplicação do método kSNN, seja dada a coleção de pontos P, representada pelos pontos $p_1, \ldots, p_5$ no espaço euclidiano R^2, conforme ilustrado na Figura 4.3, tal que p_2 é um ponto de teste, $Tr = \{p_1, p_3, p_4, p_5\} \in P$ é um conjunto de treino e $k = 2$. Ao aplicar o método kSNN para classificar o ponto p_2 ocorrem os seguinte passos:

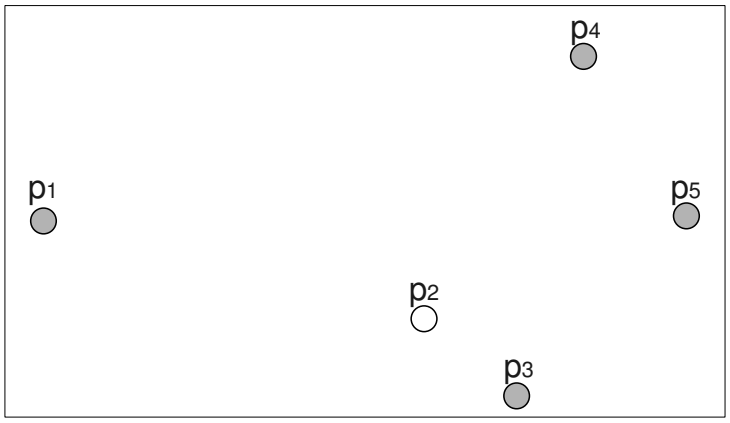

Figura 4.3: *Distribuição dos pontos da coleção P no espaço euclidiano R^2*

- As similaridades entre o ponto p_2 e cada um dos pontos do conjunto de treino Tr são calculadas utilizando o cosseno como medida.
- Os pontos p_3 e p_5 são identificados como os dois pontos mais similares ao ponto p_2.
- Entre os pontos identificados no passo anterior (p_3 e p_5), apenas o ponto p_3 possui o ponto p_2 entre os dois pontos mais similares.
- O ponto p_2 é classificado de acordo com a categoria do ponto p_3.

4.2 Geração de Características

A segunda abordagem proposta neste trabalho consiste em expandir com novas características a representação conjunto de palavras (BOW, do inglês *bag of words*) dos documentos de uma coleção.

A Tabela 4.1 mostra a estrutura da representação BOW de uma coleção Ω. Nessa tabela, d_i é o identificador do documento i na coleção Ω, $p_{i,j}$ é a frequência do termo t_j no documento d_i (Seção 3.1.2) e T é o conjunto do vocabulário da coleção Ω, tal que $1 \leq i \leq |\Omega|$ e $1 \leq j \leq |T|$.

doc	t_1	t_2	...	$t_{\|T\|}$
d_1	$p_{1,1}$	$p_{1,2}$	...	$p_{1,\|T\|}$
d_2	$p_{2,1}$	$p_{2,2}$	...	$p_{2,\|T\|}$
...	...	...	...	...
$d_{\|\Omega\|}$	$p_{\|\Omega\|,1}$	$p_{\|\Omega\|,2}$	...	$p_{\|\Omega\|,\|T\|}$

Tabela 4.1: *Estrutura da representação BOW da coleção* Ω

Para expandir a representação BOW da coleção Ω, novas características foram geradas nessa coleção, resultando em uma nova representação denominada *Similaridade BOW* (SBOW, do inglês *Similarity BOW*).

As novas características geradas na coleção Ω correspondem aos identificadores dos documentos de determinada matriz de similaridade. Essa matriz armazena o valor de similaridade do documento d_i em relação a outros documentos dessa coleção de acordo com algum critério de seleção: kNN, kINN ou kSNN (Seção 4.1).

A Tabela 4.2 mostra a estrutura da matriz de similaridade S que armazena os valores de similaridade resultantes do seguinte critério de seleção: dada a coleção Ω, selecionar os $|\Omega|$ documentos mais similares ao documento d_i. Essa matriz também é conhecida como matriz de similaridade completa da coleção Ω.

doc	doc	sim	doc	sim	...	...	doc	sim
d_1	$d_{1,1}$	$s_{1,1}$	$d_{1,2}$	$s_{1,2}$	...	...	$d_{1,\|\Omega\|}$	$s_{1,\|\Omega\|}$
d_2	$d_{2,1}$	$s_{2,1}$	$d_{2,2}$	$s_{2,2}$	...	...	$d_{2,\|\Omega\|}$	$s_{2,\|\Omega\|}$
...	...	...	...	...	...	...	...	...
$d_{\|\Omega\|}$	$d_{\|\Omega\|,1}$	$s_{\|\Omega\|,1}$	$d_{\|\Omega\|,2}$	$s_{\|\Omega\|,2}$	...	...	$d_{\|\Omega\|,\|\Omega\|}$	$s_{\|\Omega\|,\|\Omega\|}$

Tabela 4.2: *Estrutura da matriz de similaridade completa da coleção* Ω

Na Tabela 4.2, d_i é o identificador do documento i na coleção Ω, $d_{i,l}$ é o identificador do documento que está entre os l documentos mais similares ao documento d_i e $s_{i,l}$ é o valor de similaridade entre d_i e $d_{i,l}$, tal que $1 \leq i \leq |\Omega|$ e $1 \leq l \leq |\Omega|$.

As matrizes de similaridade resultantes dos critérios de seleção kNN, kINN e kSNN são derivadas da matriz de similaridade S. Por exemplo, na matriz de similaridade resultante do critério de seleção kNN, caso o documento d_l não esteja entre os documentos mais similares ao documento d_i, os valores $d_{i,l}$ e $s_{i,l}$ são nulos na matriz S.

Após aplicar a abordagem de geração de características proposta neste trabalho, a representação BOW de uma coleção é expandida com os identificadores dos documentos de determinada matriz de similaridade (resultante do critério de seleção kNN, kINN ou kSNN). A Tabela 4.3 mostra a matriz SBOW da coleção Ω, que foi expandida com os identificadores dos documentos da matriz de similaridade S (Tabela 4.2). Nessa tabela, d_i

é o identificador do documento i na coleção Ω, T é o conjunto do vocabulário da coleção Ω e:

$$p_{i,v} = \begin{cases} \text{se } (v \leq |T|), p_{i,j} \text{ (TF da representação BOW da coleção } \Omega) \\ \text{se } (v > |T|), p_{i,v} \text{ (peso da nova característica gerada)} \end{cases} \quad (4\text{-}1)$$

d_1	t_1	t_2	$\ldots$	$t_{\|T\|}$	$t_{\|T\|+1}$	$\ldots$	$t_{\|T\|+\|\Omega\|}$
d_1	$p_{1,1}$	$p_{1,2}$	$\ldots$	$p_{1,\|T\|}$	$p_{1,\|T\|+1}$	$\ldots$	$p_{1,\|T\|+\|\Omega\|}$
d_2	$p_{2,1}$	$p_{2,2}$	$\ldots$	$p_{2,\|T\|}$	$p_{2,\|T\|+1}$	$\ldots$	$p_{2,\|T\|+\|\Omega\|}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$d_{\|\Omega\|}$	$p_{\|\Omega\|,1}$	$p_{\|\Omega\|,2}$	$\ldots$	$p_{\|\Omega\|,\|T\|}$	$p_{\|\Omega\|,\|T\|+\|\Omega\|}$	$\ldots$	$p_{1,\|T\|+\|\Omega\|}$

Tabela 4.3: *Estrutura da matriz SBOW da coleção* Ω

O Algoritmo 4.4 formaliza a nova abordagem de geração de características proposta.

Algoritmo 4.4: *GerarCaracteristicas*(Ω, S)

Entrada:
Ω; //matriz BOW da coleção Ω, veja Tabela 4.1
S; //matriz de similaridade da coleção Ω, veja Tabela 4.2
Saída: Ω'; //matriz SBOW da coleção Ω, veja Tabela 4.3.

```
1 Ω' = Ω;
2 para cada (d_i ∈ Ω) faça
3     se (d_i ∈ S) então
4         para cada (d_{i,l} ∈ S) faça
5             AtribuirPeso(t_{|T|+d_{i,l}}, d_i);
6         fim
7     fim
8 fim
```

Na linha 5 do Algoritmo 4.4, a função $AtribuirPeso(t_{|T|+d_{i,l}}, d_i)$ é responsável por atribuir ao termo $t_{|T|+d_{i,l}}$ o valor do peso $p_{i,|T|+d_{i,l}}$ no documento d_i da coleção Ω'. Nos experimentos realizados foram atribuídos dois pesos distintos às novas características:

$$p_{i,v} = \begin{cases} 1 \text{ (estratégia peso 1)} \\ max(p_{i,j}) \text{ (estratégia peso máximo)} \end{cases} \quad (4\text{-}2)$$

onde $max(p_{i,j})$ é o valor do maior peso encontrado na matriz BOW da coleção Ω.

Para demonstrar a aplicação da abordagem de geração de características, foi construída uma coleção de documentos M, conforme mostrada na Tabela 4.4. A coleção M possui seis documentos que foram divididos em dois conjuntos, o conjunto de teste $Te = \{d_1, d_2\} \subset M$ e o conjunto de treino $Tr = \{d_3, d_4, d_5, d_6\} \subset M$. Os documentos do conjunto Tr foram classificados manualmente em duas categorias: SPAM e EMAIL.

doc	t_1	t_2	t_3	t_4	categoria
1	2	1	0	0	EMAIL
2	0	1	1	2	SPAM
3	3	0	7	0	EMAIL
4	5	0	3	2	SPAM
5	2	1	0	1	SPAM
6	3	2	2	0	EMAIL

Tabela 4.4: *Representação BOW da coleção M*

A Tabela 4.5 mostra a matriz de similaridade S' dos documentos da coleção M resultante do critério de seleção kNN com o valor de $k = 3$.

doc	*doc*	*sim*	*doc*	*sim*	*doc*	*sim*
1	5	0,91	6	0,86	4	0,72
2	5	0,50	3	0,46	4	0,46
3	6	0,83	4	0,74	2	0,46
4	6	0,82	5	0,79	3	0,74
5	1	0,91	4	0,79	6	0,79
6	1	0,86	3	0,83	4	0,82

Tabela 4.5: *Matriz de similaridade da coleção M utilizando o critério de seleção kNN com o valor de $k = 3$*

Após executar o algoritmo $GerarCaracteristicas(M, S')$, foram acrescentadas novas características na representação BOW da coleção M. A Tabela 4.6 mostra a representação SBOW que representa a coleção M após o processo de geração de características utilizando a estratégia *peso 1*.

doc	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	categoria
1	2	1	0	0	0	0	0	1	1	1	EMAIL
2	0	1	1	2	0	0	1	1	1	0	EMAIL
3	3	0	7	0	0	1	0	1	0	1	EMAIL
4	5	0	3	2	0	0	1	0	1	1	SPAM
5	2	1	0	1	1	0	0	1	0	1	SPAM
6	3	2	2	0	1	0	1	1	0	0	EMAIL

Tabela 4.6: *Representação SBOW da coleção M*

De modo semelhante, é possível obter matrizes SBOWs resultantes dos critérios de seleção kINN e kSNN.

CAPÍTULO 5

Metodologia Experimental

Este Capítulo apresenta a metodologia empregada nos experimentos realizados neste trabalho. Na Seção 5.1, é mostrada uma visão geral do método adotado, na Seção 5.2, são descritas as coleções utilizadas nos experimentos, na Seção 5.3, são descritas as atividades relacionadas à preparação de documentos, na Seção 5.4, é descrito o método adotado para avaliar o desempenho dos classificadores e, na Seção 5.5, são descritos os métodos de classificação utilizados nos experimentos. A leitura do Capítulo 4 é fundamental para compreender as abordagens propostas neste trabalho que foram utilizadas em algumas atividades descritas neste capítulo.

5.1 Visão geral da metodologia

A Figura 5.1 mostra o esquema do método utilizado para alcançar os objetivos definidos neste trabalho.

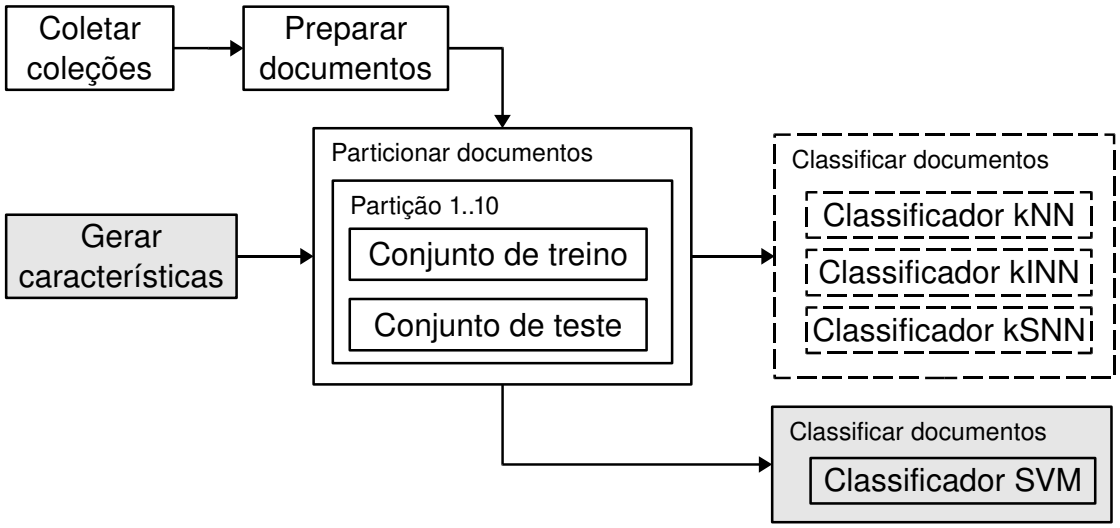

Figura 5.1: *Esquema do método adotado nos experimentos*

Conforme esquema apresentado na Figura 5.1, o método adotado nos experimentos possui seis atividades e pode ser dividido em duas etapas distintas. A atividade com a

borda pontilhada foi realizada somente na primeira etapa, as atividades com o fundo cinza foram realizadas somente na segunda etapa e as demais atividades foram realizadas em ambas etapas.

Na primeira etapa do método, os passos realizados foram os seguintes: inicialmente, as coleções de documentos foram coletadas, em seguida essas coleções foram preparadas para serem utilizadas pelos classificadores, após prepará-las, cada uma dessas coleções foi dividida em 10 partições e, por fim, os documentos de teste foram classificados pelos métodos kNN, kINN e kSNN. Na segunda etapa, as coleções foram coletadas, preparadas e particionadas, após essas atividades, a abordagem proposta para gerar características em documentos (Seção 4.2) foi aplicada nos conjuntos de treino e teste das partições, os documentos de treino foram classificados pelo método SVM e para avaliar a eficácia desse classificador, os documentos de teste também foram classificados por esse método.

As próximas seções detalham cada uma das atividades realizadas nas duas etapas do esquema apresentado na Figura 5.1, com exceção da atividade "gerar características", detalhada na Seção 4.2. Na próxima seção, são descritas as coleções de documentos utilizadas nos experimentos (Seção 5.2).

5.2 Coleção de documentos

Esta Seção descreve as coleções Reuters, 20 *Newsgroups* e Ohsumed, utilizadas nos experimentos realizados neste trabalho. Essas coleções são previamente classificadas, possuem natureza e características distintas, não exigem grande poder computacional e são bastante utilizadas em experimentos de classificação de documentos. A seguir, cada uma dessas coleções é detalhada.

5.2.1 Reuters

A Reuters [72] é uma coleção constituída originalmente por 21.578 notícias que foram divulgadas em 1987 pela agência de notícias Reuters. Essas notícias foram divididas em 5 grupos sobre economia (tópicos, instituições financeiras, organizações, pessoas e lugares) e classificadas em 135 categorias temáticas.

Por volta de 1990, a coleção Reuters foi disponibilizada à comunidade científica com o nome de Reuters-22173, pois era constituída por 22.173 documentos. Em 1996, a coleção passou por uma série de modificações, 595 documentos foram excluídos e alguns problemas corrigidos. Desde então, a coleção é denominada Reuters-21578, em referência à nova quantidade de documentos (21.578 documentos).

A Reuters-21578 possui algumas partições clássicas que diferem em relação à constituição de documentos de treino e de teste. As partições mais conhecidas e utilizadas são: Lewis Split (modLewis), Apté Split (modApté) e Hayes Split (ModHayes) [72].

A versão da Reuters utilizada neste trabalho é uma subcoleção da partição modApté denominada R10. Os documentos da R10 que possuíam nenhuma ou várias categorias relacionadas foram eliminados, resultando em uma coleção com 8 categorias, denominada Reuters-21578 R8[1].

A Reuters-21578 R8 possui um total de 7674 documentos, sendo que a quantidade de documentos por categoria varia entre 51 e 3923 documentos. A Figura 5.2 ilustra a distribuição por categoria dos documentos nessa coleção.

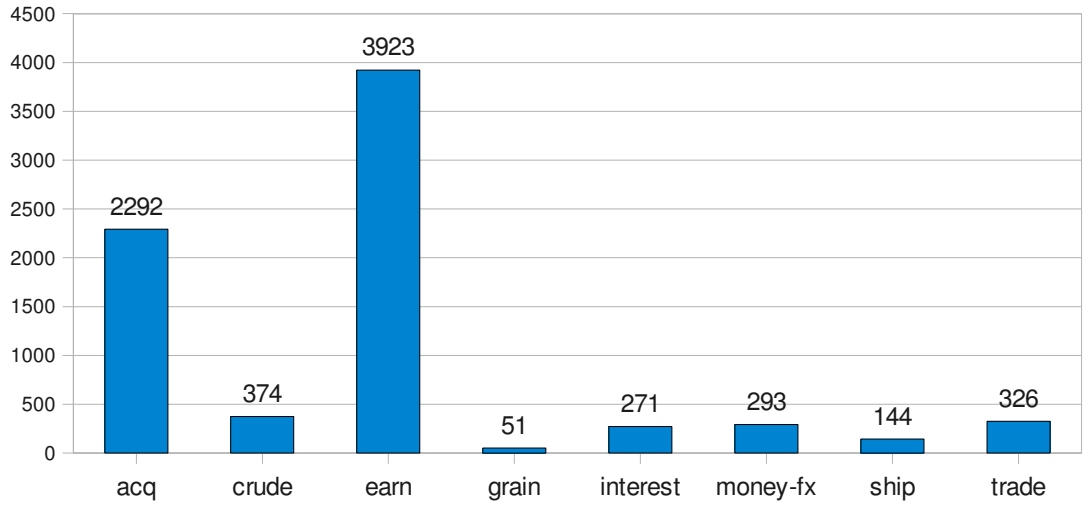

Figura 5.2: *Distribuição de documentos da coleção Reuters-21578 R8 (categoria x quant. de documentos)*

A Reuters-21578 R8 apresenta a seguinte característica: a distribuição dos documentos entre as categorias é bastante irregular, ou seja, algumas categorias possuem poucos documentos, enquanto outras possuem vários.

5.2.2 20 Newsgroups

A 20 *Newsgroups* é uma coleção com aproximadamente 20.000 documentos extraídos do fórum de discussões *UseNet*. Os documentos dessa coleção estão divididos em 20 categorias relacionadas aos seguintes assuntos: computador, negócio, religião, política, ciência e diversão.

Na versão da 20 *Newsgroups* utilizada neste trabalho, os documentos duplicados e que possuíam anexos foram excluídos, resultando em uma coleção com 18.821

[1] http://web.ist.utl.pt/ acardoso/datasets/

documentos, denominada 20 *Newsgroups*-18821[2]. Nessa coleção, a distribuição de documentos por categoria varia entre 628 e 999 documentos, conforme ilustrado na Figura 5.3.

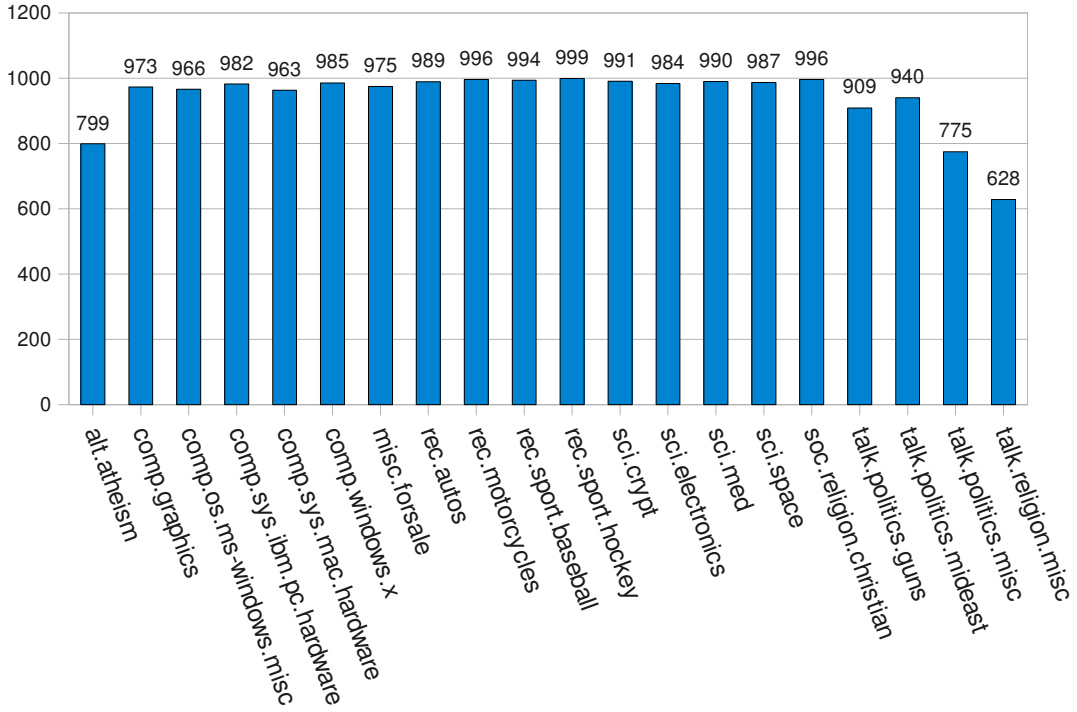

Figura 5.3: *Distribuição de documentos da coleção 20 Newsgroups (categoria x quant. de documentos)*

A 20 *Newsgroups* apresenta a seguinte característica: entre as suas categorias existem categorias muito correlacionadas e outras bastante distintas. Além disso, a distribuição dos documentos entre as categorias é regular, ou seja, as categorias possuem aproximadamente o mesmo número de documentos em cada uma delas, e cada documento pertence a apenas uma categoria.

5.2.3 Ohsumed

A coleção Ohsumed é uma subcoleção da MEDLINE, uma base de dados *online* de títulos e resumos na área de ciências médicas. A Ohsumed é constituída por 348.566 referências publicadas em 270 conferências médicas realizadas entre os anos de 1987 e 1991. Todas as referências possuem títulos e 233.445 referências possuem resumo.

A versão da Ohsumed utilizada neste trabalho é constituída por 50.216 documentos (resumos) coletados no ano de 1991. Entretanto, a eliminação de documentos que

[2]http://web.ist.utl.pt/ acardoso/datasets/

possuíam nenhuma ou várias categorias relacionadas reduziu essa coleção para 18.302 documentos, sendo denominada Ohsumed-18302[3].

A Ohsumed-18302 possui 23 categorias relacionadas a doenças cardiovasculares e a quantidade de documentos por categoria nessa coleção varia entre 56 e 2.876 documentos. A Figura 5.4 mostra a distribuição por categoria dos documentos nessa coleção.

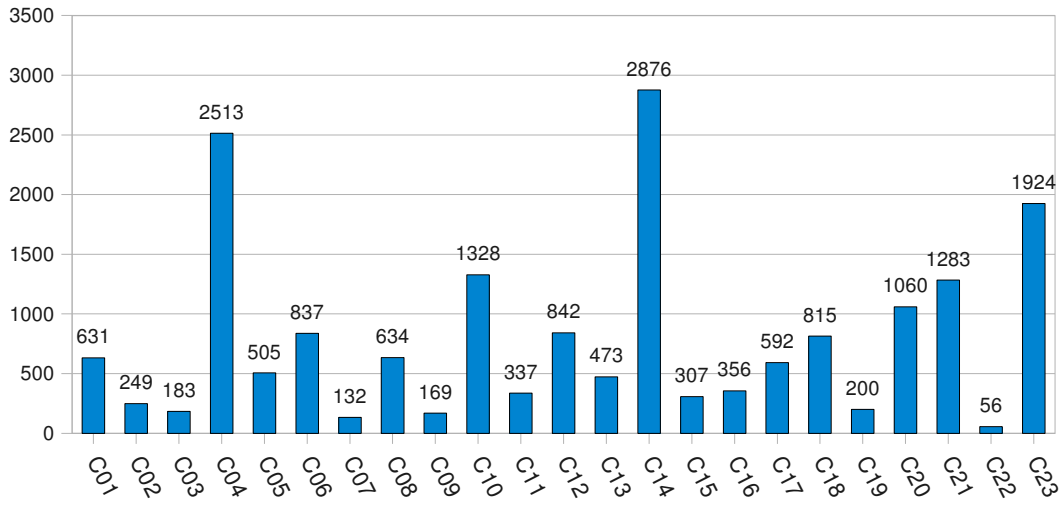

Figura 5.4: *Distribuição de documentos da coleção Ohsumed-18302 (categoria x quant. de documentos)*

A Ohsumed apresenta as seguintes características: a relação entre termos e categorias não está bem definida e a distribuição dos documentos entre as categorias é bastante irregular.

5.3 Preparação dos documentos

Os documentos das coleções utilizadas neste trabalho (Reuters, 20NG e Ohsumed) foram preparados adequadamente para serem acessados pelo classificador automático de documentos. Para indexá-los foi escolhida a representação conjunto de palavras (BOW, do inglês *bag of words*) e para atribuir o grau de importância dos termos nessa representação foi utilizada a medida TF-IDF (do inglês *Term Frequency - Inverse Document Frequency* [60]). Além disso, nesses documentos não havia marcas de pontuação, todos os termos estavam em minúsculo e todos os caracteres eram alfanuméricos.

Após indexar os documentos das coleções Reuters, 20NG e Ohsumed, quatro versões distintas foram criadas para cada uma dessas coleções:

[3] http://dit.unitn.it/moschitt/corpora.htm

- Versão AT: os documentos dessa versão foram constituídos pelos termos originais de determinada coleção. As coleções Reuters-AT, 20NG-AT e Ohsumed-AT pertencem a essa versão.
- Versão NS: as *stopwords* da *stoplist* SMART[4] foram removidas dos documentos da versão AT. Os termos que não foram removidos foram utilizados na constituição dos documentos da versão NS. As coleções Reuters-NS, 20NG-NS e Ohsumed-NS pertencem a essa versão.
- Versão ST: além da remoção das *stopwords*, o algoritmo Porter (Seção 3.2.2) foi aplicado nos documentos da versão AT. Os termos resultantes dessas operações foram utilizados na constituição dos documentos da versão ST. As coleções Reuters-ST, 20NG-ST e Ohsumed-ST pertencem a essa versão.
- Versão FS: os documentos dessa versão foram constituídos pelos melhores termos selecionados nos documentos da versão AT. Para escolher os melhores termos, foi aplicado o algoritmo ganho de informação (Seção 3.2.3) nas coleções da versão AT. As coleções Reuters-FS, 20NG-FS e Ohsumed-FS pertencem a essa versão.

A definição da quantidade de termos a serem selecionados em cada coleção foi um dos problemas enfrentados na constituição dos documentos da versão FS. Para determinar essa quantidade, cada uma das coleções da versão AT (Reuters-AT, 20NG-AT e Ohsumed-AT) foi dividida em três subcoleções distintas, constituídas pelos 2000, 7500 e 13000 melhores termos, conforme o algoritmo ganho de informação. Essas três faixas de valores foram definidas a partir da observação do comportamento da curva no gráfico que mostra o ganho de informação de acordo com o *ranking* do termo nas coleções Reuters-AT, 20NG-AT e Ohsumed-AT.

A Figura 5.5 mostra o gráfico do ganho de informação de acordo com o *ranking* do termo para o conjunto de treino de uma das partições da coleção Reuters-AT - conforme o método validação cruzada com 10 partições (Seção 3.3.5). Nas coleções 20NG-AT e Ohsumed-AT ocorreu comportamento similar e os valores 2000, 7500 e 13000 também foram definidos para essas coleções.

Os métodos kNN, kINN e kSNN, com os valores de $k = \{2,4,30,50,100\}$, foram aplicados nas três subcoleções (2000, 7500 e 13000 melhores termos) das coleções Reuters-AT, 20NG-AT e Ohsumed-AT para verificar a diferença de desempenho entre as subcoleções.

A Figura 5.6 mostra os resultados obtidos em $microF_1$ por cada um dos métodos (kNN, kINN e kSNN) ao aplicá-los com $k = \{2,4,30,50,100\}$ nas três subcoleções da coleção de documentos Reuters-AT.

[4] http://terral.lsi.uned.es/ ircourse/examples/stoplist.html

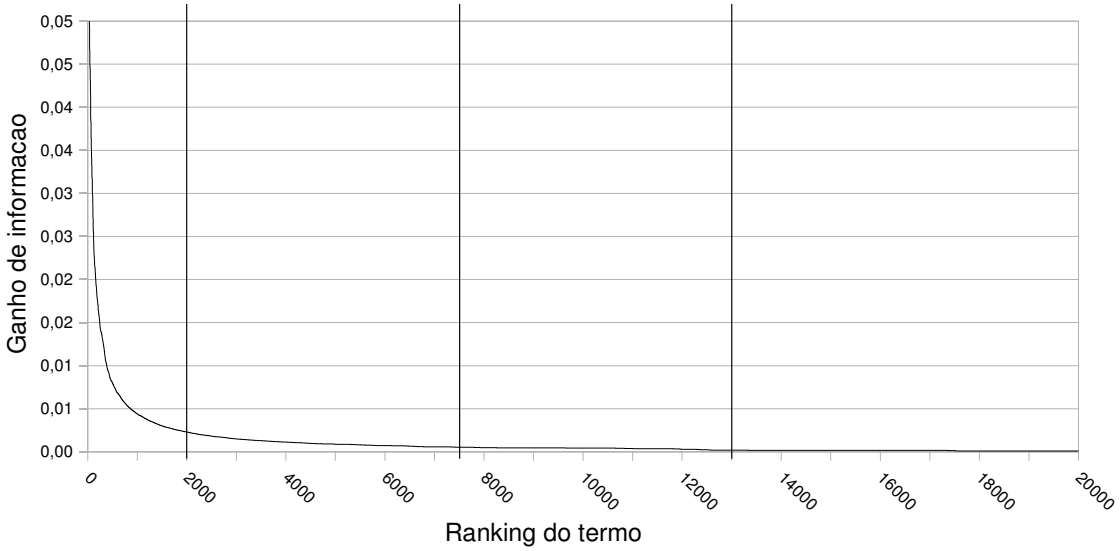

Figura 5.5: *Ganho de informação no conjunto de treino da coleção Reuters-AT*

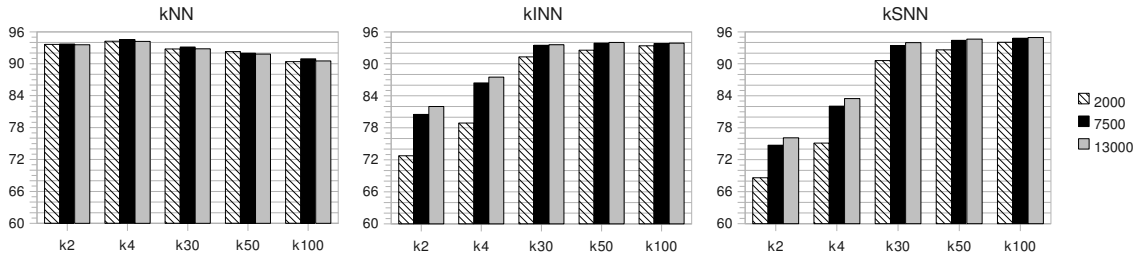

Figura 5.6: *Valores obtidos em microF$_1$ ao aplicar os métodos kNN, kINN e kSNN nas subcoleções de documentos da coleção Reuters-AT*

Conforme mostrado na Figura 5.6, ao aplicar os métodos kNN, kINN e kSNN, os resultados alcançados em *microF*$_1$ por determinado método nas três subcoleções de documentos da coleção Reuters-AT tiveram pequena diferença (menos de 1%). Entre as subcoleções das coleções 20NG-AT e Ohsumed-AT também ocorreu comportamento similar. Dessa forma, para evitar a realização de experimentos desnecessários, as coleções de documentos FS foram constituídas pelo conjunto de treino com os 13000 melhores termos, além do conjunto de teste.

5.4 Método de avaliação

Os experimentos realizados neste trabalho foram avaliados experimentalmente utilizando o método de validação cruzada com 10 partições (Seção 3.3.5). Esse método foi escolhido por ser amplamente adotado na avaliação de atividades de classificação de documentos [34].

De acordo com o método de validação cruzada com 10 partições, após criar as quatro versões (AT, NS, ST e FS) das coleções Reuters, 20NG e Ohsumed, os documentos de cada uma dessas versões foram divididos em 10 conjuntos de teste distintos ($Te_1, Te_2, \ldots, Te_{10}$) escolhidos aleatoriamente com aproximadamente $|D|/10$ documentos em cada conjunto, onde D é uma coleção de documentos. O classificador Ψ foi construído utilizando o conjunto de treino $D - Te_i$ e avaliado utilizando o conjunto de teste Te_i. Ao final da avaliação dos 10 classificadores, a média dos 10 resultados obtidos, para $macroF_1$ e $microF_1$, foi calculada para a avaliação final da classificação. Para validar esses resultados estatisticamente foi aplicado o teste Wilcoxon [127].

5.5 Métodos de classificação

Os métodos de classificação foram utilizados em duas etapas diferentes, conforme ilustrado na Figura 5.1. Na primeira etapa, foram utilizados o método kNN (Seção 3.3.3) e os métodos propostos, kINN e kSNN (Seção 4.2), e na segunda etapa foi utilizado o método SVM.

5.5.1 Métodos kNN, kINN e kSNN

O método kNN alcança o melhor desempenho quando o valor de k é relativamente alto (entre 30 e 200) [131] [132] [133]. Na prática, o valor de k normalmente é obtido após vários testes sobre o conjunto de treino [11]. Tendo em vista essas observações, nos experimentos realizados neste trabalho os métodos kNN, kINN e kSNN foram aplicados com os seguintes valores para o parâmetro k, tal que $2 \leq k \leq 200$: quando $2 \leq k \leq 5$, o valor de k variou de 1 em 1 unidade e quando $10 \leq k \leq 200$, o valor de k variou de 10 em 10 unidades.

A medida de similaridade adotada pelos métodos kNN, kINN e kSNN nos cálculos da similaridade entre documentos foi o cosseno (Seção 3.1.3) e algumas matrizes de similaridade, resultantes desses cálculos, foram posteriormente utilizadas pela abordagem de geração de características proposta neste trabalho (Seção 4.2).

5.5.2 Método SVM

Neste trabalho foi utilizada a implementação SVM light com *kernel* linear. De acordo com Joachims [59], a maioria dos problemas de classificação de documentos são problemas linearmente separáveis e, consequentemente, a maioria dos estudos na área utilizam um *kernel* SVM linear [14] [20] [40].

CAPÍTULO 6

Resultados Experimentais

Este capítulo apresenta os resultados obtidos nos experimentos realizados neste trabalho. Na Seção 6.1, são descritos os resultados experimentais relacionados às variações do método kNN propostas e, na Seção 6.2, são descritos os resultados experimentais relacionados à abordagem de geração de características em documentos proposta. A leitura do Capítulo 5 é fundamental para compreender a metodologia adotada para alcançar os resultados apresentados neste capítulo.

6.1 Variações do método kNN

Nesta seção, são apresentados os resultados dos experimentos realizados com o objetivo de responder ao Problema de Pesquisa 1 definido na Seção 1.2.1. Especificamente, foram realizados experimentos aplicando os métodos kNN, kINN e kSNN nas versões AT, NS, ST e FS das coleções Reuters, 20NG e Ohsumed. O método de validação cruzada foi utilizado em cada versão dessas três coleções, sendo que os mesmos conjuntos de teste e treino foram utilizados em cada um dos métodos de classificação empregados.

As Tabelas 6.1, 6.2, 6.3 e 6.4 apresentam, respectivamente, os resultados obtidos para as versões AT, NS, ST e FS das coleções Reuters, 20NG e Ohsumed. Em cada tabela, a coluna $macroF_1$ mostra o maior valor médio de $macroF_1$ para um dado método e o valor de k em que o valor de $macroF_1$ foi alcançado. Na coluna $microF_1$ de cada tabela mostra informações análogas. Os valores apresentados nessas colunas correspondem à media dos valores de $macroF_1$ ou $microF_1$ obtidos para as 10 partições utilizadas na validação cruzada.

As Tabelas 6.1, 6.2, 6.3 e 6.4 mostram os ganhos ou perdas em $macroF_1$ e $microF_1$ alcançados pelos métodos propostos em diferentes versões. Com o objetivo de obter uma visão global sobre o desempenho dos métodos, os melhores resultados de cada método foram agrupados na Tabela 6.5. Essa tabela apresenta um resumo dos melhores resultados alcançados em $macroF_1$ e $microF_1$ pelos métodos kNN, kINN e kSNN entre as versões AT, NS, FS e ST das coleções de documentos Reuters, 20NG, e Ohsumed.

Coleção	Método	$macroF_1$		$microF_1$		Ganho sobre o kNN	
		valor	k	valor	k	$macroF_1$	$microF_1$
Reuters-AT	kNN	88,29	30	94,41	110	–	–
	kINN	**90,24**	90	95,01	200	2,21 ▲	0,64 ▲
	kSNN	90,12	80	**95,54**	170	2,07 ▲	1,20 ▲
20NG-AT	kNN	90,89	5	91,05	10	–	–
	kINN	90,94	30	91,09	30	0,06 ●	0,04 ●
	kSNN	**90,97**	40	**91,14**	50	0,09 ●	0,10 ●
Ohsumed-AT	kNN	67,95	10	73,09	20	–	–
	kINN	**68,71**	40	**74,34**	50	1,12 ■	1,71 ▲
	kSNN	68,60	40	74,12	60	0,96 ■	1,41 ▲

Tabela 6.1: *Ganhos obtidos em $macroF_1$ e $microF_1$ sobre o método kNN ao aplicar o método kINN ou kSNN nas coleções Reuters-AT, 20NG-AT e Ohsumed-AT*

Coleção	Método	$macroF_1$		$microF_1$		Ganho sobre o kNN	
		valor	k	valor	k	$macroF_1$	$microF_1$
Reuters-NS	kNN	88,61	40	93,93	150	–	–
	kINN	90,21	140	94,90	200	1,81 ■	1,03 ▲
	kSNN	**90,44**	100	**95,56**	200	2,07 ▲	1,74 ▲
20NG-NS	kNN	90,72	10	90,85	10	–	–
	kINN	90,74	40	90,98	40	0,02 ●	0,14 ●
	kSNN	**90,80**	50	**91,03**	50	0,09 ●	0,20 ●
Ohsumed-NS	kNN	68,12	10	73,07	20	–	–
	kINN	**68,90**	20	**74,48**	50	1,15 ●	1,93 ▲
	kSNN	68,71	40	74,10	60	0,87 ●	1,41 ▲

Tabela 6.2: *Ganhos obtidos em $macroF_1$ e $microF_1$ sobre o método kNN ao aplicar os métodos kINN ou kSNN nas coleções Reuters-NS, 20NG-NS e Ohsumed-NS.*

Coleção	Método	$macroF_1$		$microF_1$		Ganho sobre o kNN	
		valor	k	valor	k	$macroF_1$	$microF_1$
Reuters-ST	kNN	88,21	70	93,18	200	–	–
	kINN	89,90	180	94,18	190	1,92 ▲	1,07 ▲
	kSNN	**90,11**	130	**95,11**	200	2,15 ▲	0,99 ▲
20NG-ST	kNN	**90,47**	5	90,59	5	–	–
	kINN	90,39	30	90,65	40	-0,09 ●	0,07 ●
	kSNN	90,44	40	**90,67**	40	-0,03 ●	0,09 ●
Ohsumed-ST	kNN	68,08	10	73,18	20	–	–
	kINN	**69,16**	30	**74,71**	50	1,59 ▲	2,09 ▲
	kSNN	68,78	60	74,28	60	1,03 ▲	1,50 ▲

Tabela 6.3: *Ganhos obtidos em $macroF_1$ e $microF_1$ sobre o método kNN ao aplicar o método kINN ou kSNN nas coleções Reuters-ST, 20NG-ST e Ohsumed-ST.*

Coleção	Método	$macroF_1$		$microF_1$		Ganho sobre o kNN	
		valor	k	valor	k	$macroF_1$	$microF_1$
Reuters-FS	kNN	88,51	20	**95,09**	80	–	–
	kINN	**90,24**	90	94,94	130	1,95 ▲	-0,16 ●
	kSNN	88,24	180	93,45	180	-0,31 ●	-1,72 ▲
20NG-FS	kNN	88,42	10	88,81	10	–	–
	kINN	**90,94**	30	**91,09**	30	2,85 ▲	2,57 ▲
	kSNN	88,53	60	89,04	160	0,12 ●	0,26 ●
Ohsumed-FS	kNN	67,62	10	72,92	20	–	–
	kINN	**68,71**	40	**74,34**	50	1,61 ▲	1,95 ▲
	kSNN	68,17	40	73,36	80	0,81 ●	0,60 ■

Tabela 6.4: *Ganhos obtidos em $macroF_1$ e $microF_1$ sobre o método kNN ao aplicar o método kINN ou kSNN nas coleções Reuters-FS, 20NG-FS e Ohsumed-FS.*

Coleção	Método	$macroF_1$				$microF_1$			
		valor	k	versão	ganho%	valor	k	versão	ganho%
Reuters	kNN	88,61	40	NS	–	95,09	80,90	FS	–
	kINN	90,24	90,120	AT,FS	1,83 ■	95,01	200	AT	-0,08 ●
	kSNN	**90,44**	100	NS	2,07 ▲	**95,56**	200	NS	0,49 ■
20NG	kNN	90,89	5	AT	–	91,05	10	AT	–
	kINN	90,94	30	AT,FS	0,06 ●	91,09	30	AT,FS	0,04 ●
	kSNN	**90,97**	40	AT	0,09 ●	**91,14**	50	AT	0,10 ●
Ohsumed	kNN	68,12	10	NS	–	73,18	20	ST	–
	kINN	**69,16**	30	ST	1,52 ●	**74,71**	50	ST	2,09 ▲
	kSNN	68,78	60	ST	0,96 ●	74,28	60	ST	1,50 ▲

Tabela 6.5: *Maiores valores obtidos em $macroF_1$ e $microF_1$ na aplicação dos método kNN, kINN e kSNN nas versões AT, NS, ST e FS das coleções de documentos Reuters, 20NG e Ohsumed.*

Os valores em negrito, apresentados nas colunas $macroF_1$ e $microF_1$ das Tabelas 6.1, 6.2, 6.3, 6.4 e 6.5, correspondem aos maiores valores alcançados nos experimentos. Sendo assim, o melhor valor médio em $macroF_1$ para a coleção Reuters na versão AT (Tabela 6.1) foi obtido com o método kINN com $k = 90$ e o melhor valor em $microF_1$ para a mesma coleção foi obtido pelo método kSNN com $k = 170$. Em cada tabela, a coluna *Ganho sobre o kNN* mostra a diferença de desempenho entre um método e o método kNN. Nessa coluna, valores negativos representam perdas e valores positivos representam ganhos percentuais em relação ao método kNN e as figuras ▲, ■ e ● significam, respectivamente, que o ganho ou perda apresentado foi fortemente significativo ($\geq 98\%$), significativo ($90\% \leq x < 98\%$) ou não significativo, conforme o teste Wilcoxon [127].

Conforme pode ser observado nos resultados apresentados nas Tabelas 6.1, 6.2,

6.3 e 6.4, a hipótese que os métodos propostos (kINN e kSNN) poderiam apresentar eficácia superior[1] ao método kNN se confirmou em algumas versões das coleções Reuters, 20NG e Ohsumed. Nas versões AT, NS e ST da coleção Reuters, os métodos propostos apresentaram ganhos estatisticamente significativos em $macroF_1$ e $microF_1$ sobre todos os resultados obtidos pelo método kNN e nas versões AT, NS, ST e FS da coleção Ohsumed, os métodos propostos apresentaram ganhos estatisticamente significativos em $macroF_1$ e $microF_1$ sobre a maioria dos resultados obtidos pelo método kNN.

De acordo com os resultados apresentados na Tabela 6.5, os métodos propostos apresentaram ganhos estatisticamente significativos notadamente em $macroF_1$, na coleção Reuters, e em $microF_1$, na coleção Ohsumed. O método kINN obteve o melhor desempenho em $microF_1$ na coleção Ohsumed. Entretanto, esse método obteve perda insignificante em relação ao método kNN na coleção Reuters. Pôde-se observar também que o método kSNN apresentou pequenos ganhos, porém estatisticamente significativos, em $macroF_1$ na coleção Reuters e em $microF_1$ na coleção Ohsumed. Além disto, o método kSNN não apresentou perda em $macroF_1$ ou $microF_1$ sobre o método kNN em nenhuma das coleções.

Ao comparar a eficácia de determinado método entre diferentes versões das coleções Reuters, 20NG e Ohsumed, os métodos kNN, kINN e kSNN tiveram eficácias semelhantes entre as versões AT, NS e ST (Tabelas 6.1, 6.2 e 6.3). Esses resultados podem ser justificados pela medida da importância dos termos adotada nos experimentos. A remoção de *stopwords* e a aplicação da técnica de *stemming* na versão AT das coleções Reuters, 20NG e Ohsumed tiveram pequena influência na eficácia da classificação das coleções resultantes (versões NS e ST). Essa pequena influência ocorreu devido à medida TF-IDF já considerar um baixo peso para termos que ocorrem muito em muitos documentos, este é o caso das *stopwords* e de alguns termos resultantes da técnica de *stemming*.

A Tabela 6.4 mostra que a versão FS (com seleção de características) das três coleções apresentaram resultados de classificação inferiores à versão AT quando os métodos kNN e kSNN foram aplicados nessas coleções. Entretanto, os resultados do método kINN foram praticamente inalterados para ambas as versões nas três coleções, o que sugere que esse método pode ser menos influenciado à escolha de características utilizadas nas coleções que os métodos kNN e kSNN. Dessa forma, se faz necessário conduzir experimentos em outras coleções de texto e utilizar diferentes critérios de seleção de características para se comprovar essa hipótese.

As Tabelas 6.1, 6.2 e 6.3 mostram que em todas as versões da coleção 20NG, os ganhos dos métodos propostos sobre o método kNN não foram estatisticamente significativos. O mesmo não ocorre com as coleções Reuters e Ohsumed nas quais os

[1] A eficácia de determinado método é superior ao método KNN quando existe ganhos estatisticamente significativos em $macroF_1$ ou $microF_1$.

métodos propostos apresentaram ganhos significativos em $macroF_1$ e $microF_1$ na maioria das versões experimentadas. Devido a distribuição de documentos por categoria em cada uma das coleções, pode-se imaginar que há uma relação entre a distribuição dos documentos nas coleções e o desempenho dos métodos. Na coleção 20NG, a distribuição dos documentos entre as categorias é uniforme (veja Figura 5.3). Contudo, nas coleções Reuters e Ohsumed, ao contrário, há discrepância na distribuição de documentos entre as categorias (veja as Figuras 5.2 e 5.4).

Para verificar como a irregularidade na distribuição dos documentos sobre as categorias influenciou na decisão do método kNN e dos métodos propostos (kINN e kSNN), as matrizes de confusão resultantes dos melhores resultados obtidos em $macroF_1$ na aplicação de cada um desses métodos nas coleções Reuters-NS e Ohsumed-NS foram analisadas[2]. Na coleção Reuters-NS, o método kNN obteve o melhor resultado com $k = 40$, o método kINN com $k = 140$ e o método kSNN com $k = 100$ e, na coleção Ohsumed-NS, o método kNN obteve o melhor resultado com $k = 10$, o método kINN com $k = 20$ e o método kSNN com $k = 40$ (Tabela 6.2).

A matriz de confusão possibilita, dado um conjunto de teste Te, visualizar a quantidade de classificações corretas sobre as classificações preditas para cada categoria. A quantidade de acertos em cada categoria se localiza na diagonal principal $M(C_i, C_i)$ da matriz e os demais elementos $M(C_i, C_j)$, para $i \neq j$, representam erros na classificação. A matriz de confusão de um classificador ideal possui todos esses elementos iguais a zero, uma vez que ele não comete erros. Nos experimentos realizados, utilizou-se o método de validação cruzada com 10 partições. Dessa forma, o método de classificação é aplicado em cada uma das 10 partições, resultando em 10 matrizes de confusão.

A Tabela 6.6 apresenta a matriz de confusão da primeira partição resultante da aplicação do método kNN com $k = 40$ na coleção Reuters-NS. Nessa matriz, $pr(1)$ apresenta a quantidade de documentos classificados em determinada categoria, $rc(1)$ apresenta a quantidade de documentos que pertence à determinada categoria, $pr(\%)$ apresenta a precisão na classificação de determinada categoria e $pr(\%)$ apresenta a cobertura na classificação de determinada categoria.

Ao analisar as matrizes de confusão dos melhores resultados obtidos em $macroF_1$ pelos métodos kNN, kINN e kSNN, observou-se que a categoria dominante (categoria com a maior quantidade de documentos em relação ao total de documentos) foi a categoria que apresentou maior variação em precisão e cobertura entre as categorias das coleções Reuters-NS e Ohsumed-NS.

[2] A versão NS foi escolhida na coleção Reuters por apresentar o melhor resultado em $macroF_1$ e $microF_1$ entre todas as versões e essa versão foi escolhida na coleção Ohsumed por apresentar resultado apenas marginalmente diferente das outras versões.

p1	categoria	0	1	2	3	4	5	6	7	rc(1)	rc(%)
0	acq	195	.	30	.	1	.	.	1	227	85,90
1	crude	.	26	.	.	.	.	.	.	26	100,00
2	earn	1	.	400	.	1	.	.	.	402	99,50
3	grain	.	.	.	3	.	.	.	1	4	75,00
4	interest	.	.	1	.	18	2	.	.	21	85,71
5	money-fx	.	1	1	.	6	25	.	1	34	73,53
6	ship	.	.	.	.	.	.	13	.	13	100,00
7	trade	.	.	.	.	1	.	.	39	40	97,50
	pr(1)	196	27	432	3	27	27	13	42		
	pr(%)	99,49	96,30	92,59	100	66,67	92,59	100	92,86		

Tabela 6.6: *Matriz de confusão da primeira partição resultante da aplicação do método kNN com $k = 40$ na coleção Reuters-NS*

A Tabela 6.7 apresenta as médias de precisão e de cobertura da categoria dominante das coleções Reuters-NS e Ohsumed-NS, obtidas ao aplicar os métodos kNN, kINN e kSNN nas 10 partições de cada coleção.

Coleção	Categoria	Método	k	Precisão	Cobertura
Reuters-NS	*earn*	kNN	40	91,90	**99,11**
		kINN	140	**98,92**	94,96
		kSNN	100	98,19	96,35
Ohsumed-NS	*C*14	kNN	10	75,08	**91,66**
		kINN	20	**80,80**	87,82
		kSNN	40	79,16	90,97

Tabela 6.7: *Precisão média e cobertura média da categoria dominante das coleções Reuters-NS e Ohsumed-NS.*

Conforme os resultados apresentados na Tabela 6.7, os métodos propostos foram mais precisos que o método kNN na classificação de documentos de teste da categoria dominante (*earn*, na coleção Reuters, e *C14*, na coleção Ohsumed). Entretanto, o método kNN possui maior cobertura que os métodos propostos na categoria dominante. Em outras palavras, quando um método proposto afirma que um documento de teste pertence à categoria dominante, a probabilidade dessa decisão ser correta é maior que ao ser classificado pelo método kNN. Entretanto, apesar do método kNN afirmar erroneamente que alguns documento de teste pertencem à categoria dominante, esse método acerta uma quantidade maior de documentos dessa categoria.

Para verificar se a categoria dominante foi a categoria que teve maior influência nos ganhos obtidos em $macroF_1$ pelos métodos propostos, os valores de F_1 obtidos ao aplicar o método kINN em cada uma das categorias da coleção Reuters-NS, foi comparado com os valores obtidos ao aplicar o método kNN. A Tabela 6.8 apresenta os

ganhos obtidos em F_1 pelo método kINN em cada uma das categorias da coleção Reuters-NS ao aplicar, respectivamente, os métodos kNN e kINN.

Categoria	F_1		Ganho sobre o kNN
	kNN	kINN	
acq	91,16	94,46	3,50
crude	94,60	94,35	-0,26
earn	95,37	96,90	1,58
grain	87,25	87,75	0,57
interest	83,39	84,48	1,29
money-fx	81,58	83,21	1,96
ship	87,43	90,85	3,77
trade	91,53	92,73	1,29

Tabela 6.8: *Precisão média, cobertura média e F_1 das categorias da coleção Reuters-NS ao aplicar os métodos kNN e kINN.*

Conforme os resultados apresentados na Tabela 6.8, a categoria dominante (*earn*) não foi a categoria que teve maior influência nos ganhos obtidos em *macro*F_1 pelos métodos propostos. A categoria que obteve a maior influência (*ship*) possui poucos documentos e a alteração na classificação de um documento dessa categoria tem grande impacto no valor de F_1 e, consequentemente, no valor de *macro*F_1 dos métodos kNN, kINN e kSNN. Dessa forma, devido à ausência de um comportamento padrão ao aplicar os métodos kNN, kINN e kSNN na classificação de documentos que não eram da categoria dominante, não foi possível justificar os ganhos obtidos em *macro*F_1 pelos métodos propostos. Contudo, como mostra a Tabela 6.8, os ganhos em F_1 ocorreram na maioria das categorias da coleção Reuters, mostrando que o comportamento superior do método kINN se manifestou em cada categoria. Situações semelhantes puderam ser observadas na coleção Ohsumed.

6.1.1 Análise da variação do valor de k

Um aspecto importante quanto à eficácia do método kNN e também de suas variações propostas (kINN e kSNN) é a escolha do parâmetro k. Essa escolha é consequência de um processo empírico e deve ser feita sobre o conjunto de treino de uma coleção.

O parâmetro k tem significados distintos para cada um dos critérios estudados e, portanto, exerce influências distintas em cada método. No critério kNN, o valor de k determina um ponto de corte na escolha dos documentos do conjunto de treino que são mais similares a um documento de teste d. Como consequência, são obtidos no máximo k vizinhos para d e a decisão quanto à categoria de d é tomada com base nas categorias desses vizinhos.

No critério kINN, k determina um ponto de corte na escolha de documentos de treino que possuem o documento d como um dos seus documentos mais similares. Como consequência, o números de vizinhos de d pode ser, menor, igual ou maior que k. Já no critério kSNN, k corresponde a uma interseção entre os dois critérios anteriores (kNN e kINN) e como consequência, o número de vizinhos de d é no máximo k.

Para analisar a variação do valor de k, foram realizados experimentos com o objetivo de identificar qual método (kNN, kINN ou kSNN) é menos sensível à variação do valor desse parâmetro. Nesses experimentos, utilizou-se o método de validação cruzada com 10 partições e o valor de k variou de 2 até 200. Os valores de k iniciais utilizados foram: 2,3,4,5 e 10. A partir do valor 10, variou-se o valor de k de 10 em 10 até 200. Além disso, foram utilizadas as versões das coleções que alcançaram os maiores resultados em $macroF_1$ e $microF_1$ (Tabela 6.5). Assim, foram utilizadas a versão NS para a coleção Reuters, AT para a coleção 20NG e ST para a coleção Ohsumed.

As Figuras 6.1 e 6.2, 6.3 e 6.4, 6.5 e 6.6 mostram, respectivamente, os gráficos com os valores obtidos em $macroF_1$ e $microF_1$ na aplicação dos métodos kNN, kINN e kSNN nas coleções Reuters-NS, 20NG-AT e Ohsumed-ST.

Os gráficos com os valores obtidos em $macroF_1$ e $microF_1$ não são suficientes para identificar qual método (kNN, kINN ou kSNN) é menos sensível à variação do valor de k, pois em alguns gráficos a diferença na variação desse valor entre os métodos é visualmente semelhante. Dessa forma, o desvio padrão foi escolhido como medida para comparar a variação do valor de k entre os métodos.

A Tabela 6.9 apresenta os desvios padrões obtidos em relação à media dos valores obtidos em $macroF_1$ e $microF_1$ ao aplicar os métodos kNN, kINN e kSNN nas coleções Reuters-NS, 20NG-AT e Ohsumed-ST. Nessa tabela, a coluna "Média" apresenta a média dos valores obtidos em $macroF_1$ ou $microF_1$ ao aplicar determinado método com o valor de k variando de 2 até 200.

Coleção	Método	Média		Desvio padrão	
		$macroF_1$	$microF_1$	$macroF_1$	$microF_1$
Reuters-NS	kNN	86,11	92,23	**1,91**	**2,09**
	kINN	87,35	91,30	3,52	4,08
	kSNN	86,86	91,64	4,46	4,69
20NG-AT	kNN	89,00	89,59	**1,14**	**0,85**
	kINN	89,03	89,07	1,54	2,19
	kSNN	88,81	88,52	2,22	3,31
Ohsumed-ST	kNN	62,89	70,11	2,81	**1,69**
	kINN	65,70	71,47	**2,57**	3,29
	kSNN	65,13	70,65	3,13	4,33

Tabela 6.9: *Desvio-padrão em $microF_1$ e $macroF_1$ na aplicação dos métodos kNN, kINN e kSNN nas coleções Reuters-NS, 20NG-AT e Ohsumed-ST.*

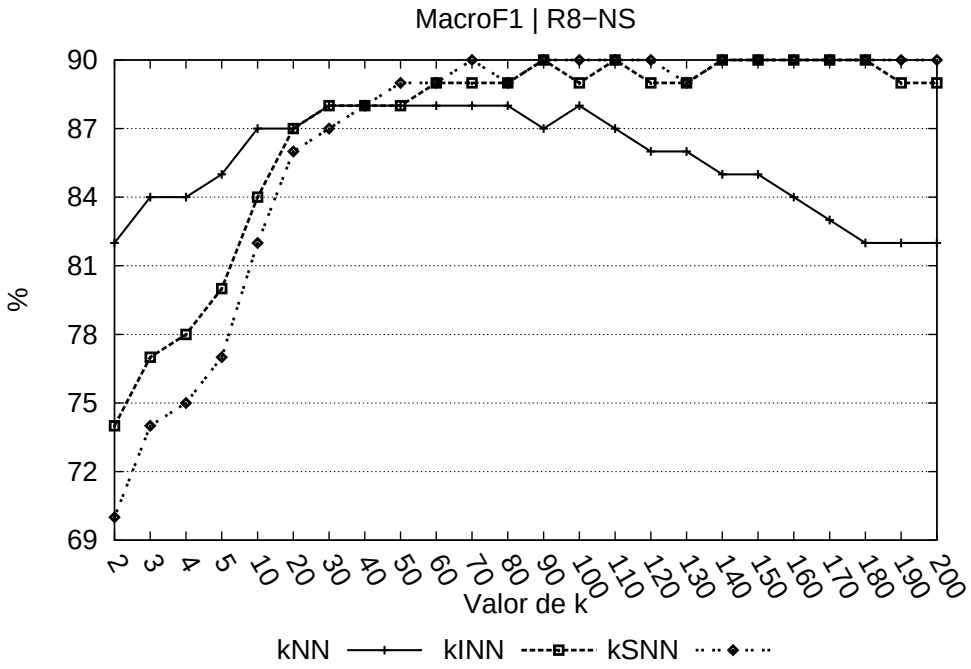

Figura 6.1: *Valores obtidos em macroF_1 na coleção Reuters-NS*

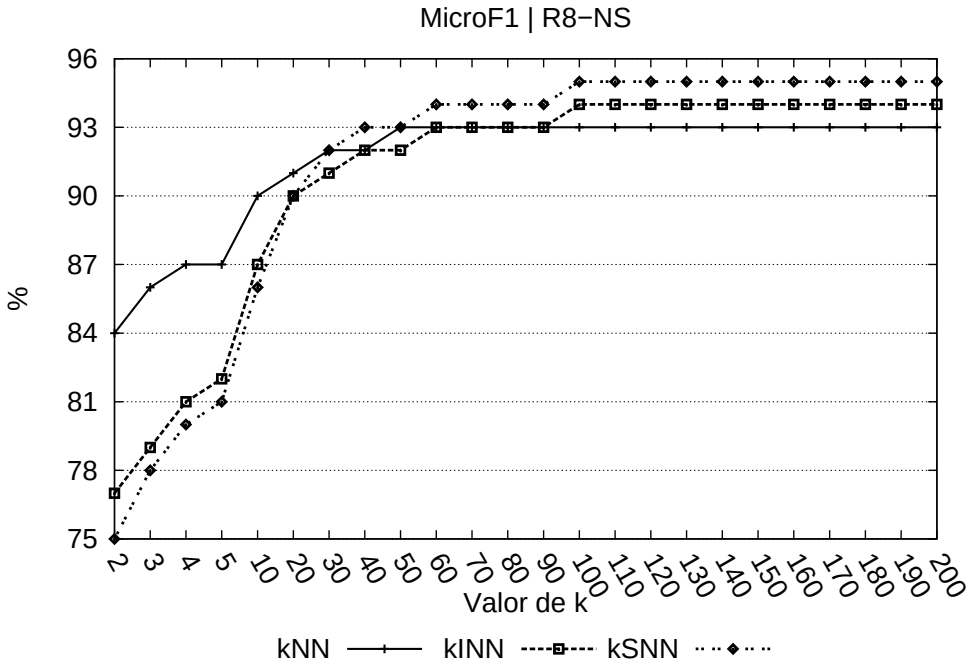

Figura 6.2: *Valores obtidos em microF_1 na coleção Reuters-NS*

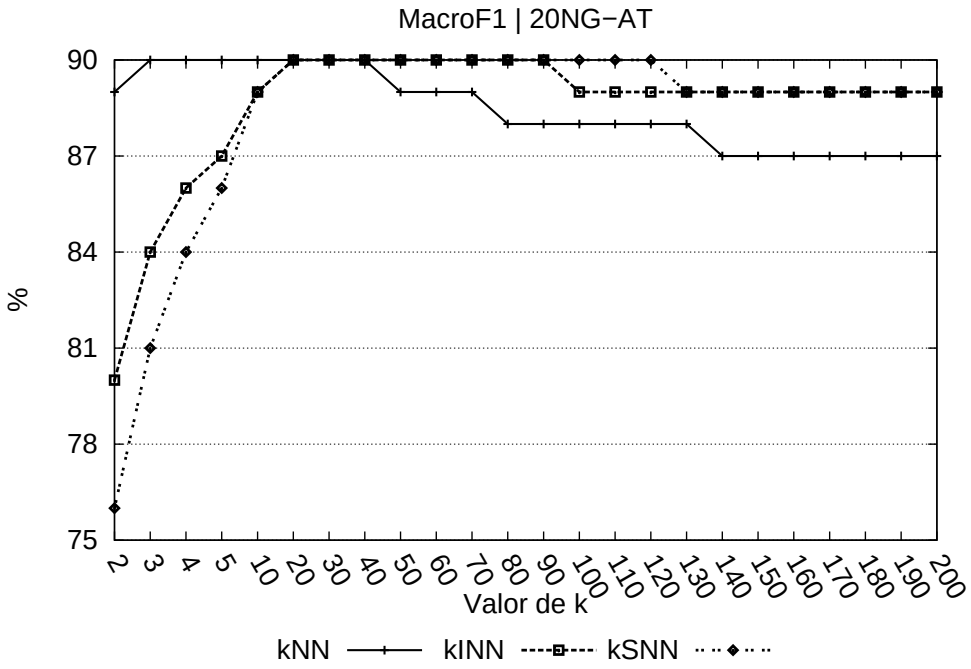

Figura 6.3: *Valores obtidos em macroF_1 na coleção 20NG-AT*

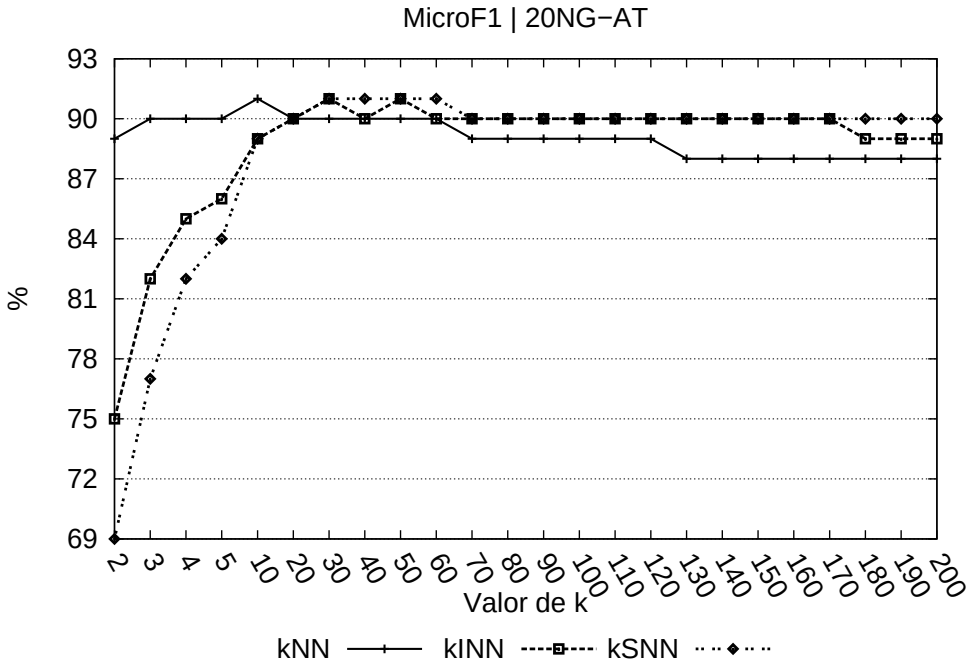

Figura 6.4: *Valores obtidos em microF_1 na coleção 20NG-AT*

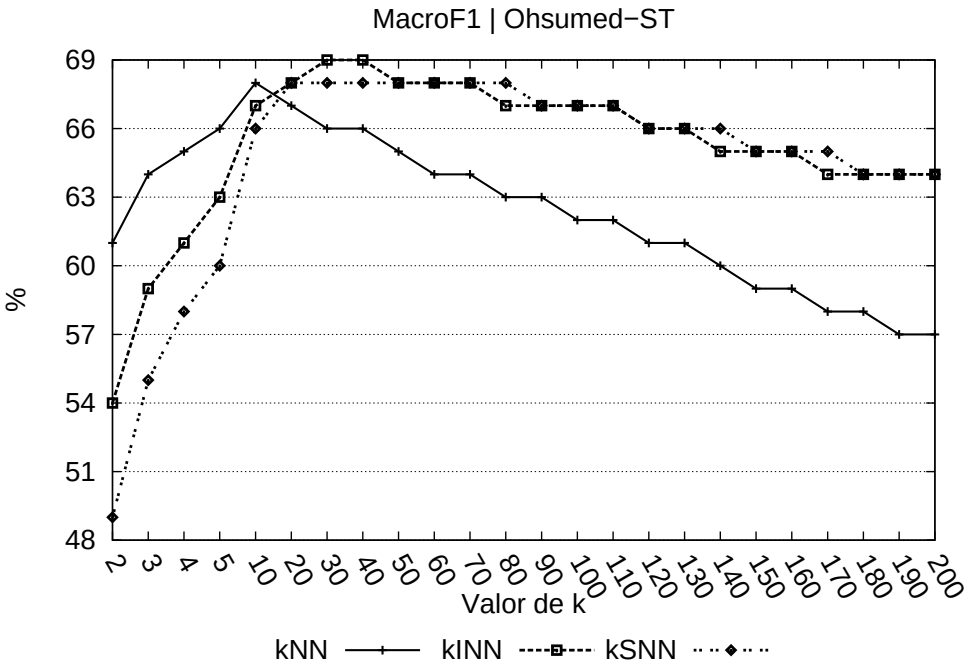

Figura 6.5: *Valores obtidos em macroF_1 na coleção Ohsumed-ST*

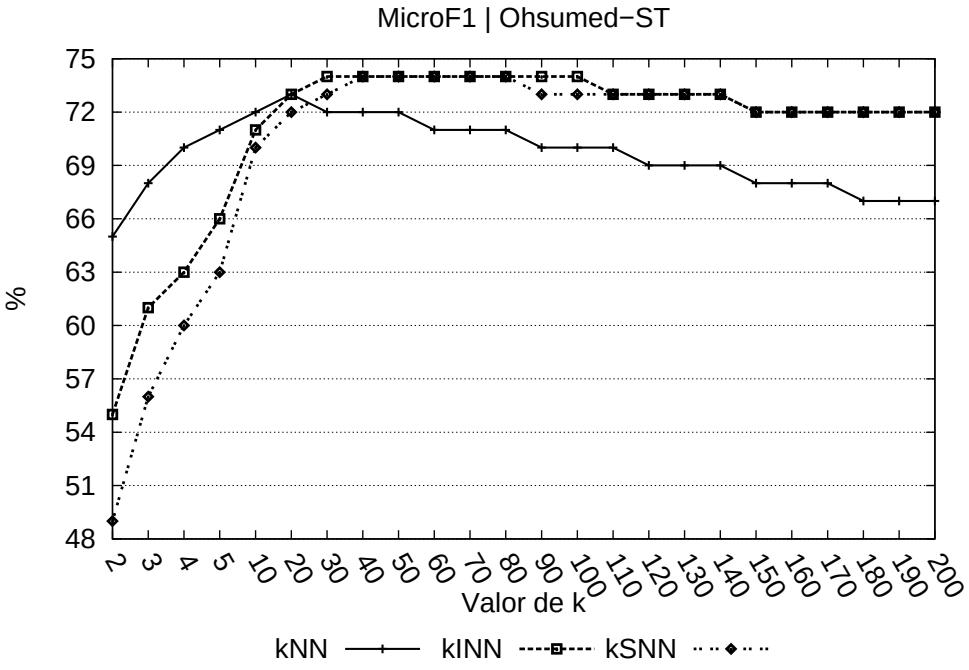

Figura 6.6: *Valores obtidos em microF_1 na coleção Ohsumed-ST*

Conforme resultados apresentados na Tabela 6.9, foi possível verificar que o método kNN foi menos sensível em $microF_1$ à variação do valor de k que os métodos kINN e kSNN. Em $macroF_1$, somente na coleção Ohsumed-ST que o método kINN foi menos sensível à variação do valor de k que os demais métodos (kNN e kSNN). Além disso, o método kSNN apresentou-se como o método mais sensível à variação do valor de k ao ser aplicado nas coleções Reuters-NS, 20NG-AT e Ohsumed-ST.

6.2 Geração de Características

Nesta seção, são apresentados os resultados obtidos na aplicação da abordagem de geração de características proposta neste trabalho (Seção 4.2). Em suma, essa abordagem consiste em expandir com novos termos a representação conjunto de palavras (BOW, do inglês *bag of words*) de uma coleção de documentos.

Conforme explicado na Seção 4.2, para expandir a matriz BOW de uma coleção com novos termos foram utilizados os identificadores dos documentos da matriz de similaridade (documentos 'vizinhos mais próximos') resultante do critério de seleção utilizado pelo método que alcançou o maior valor em $macroF_1$ ou $microF_1$ ao ser aplicado nas versões AT, NS, ST e FS dessa coleção.

A Tabela 6.10 mostra os maiores valores obtidos em $macroF_1$ e $microF_1$, o valor de k e a versão da coleção que obteve esses valores ao aplicar os métodos kNN, kINN e kSNN nas versões AT, NS, ST e FS das coleções de documentos Reuters, 20NG e Ohsumed.

Coleção	Método	$macroF_1$			$microF_1$		
		valor	k	versão	valor	k	versão
Reuters	kSNN	**90,44**	100	NS	**95,56**	200	NS
20NG	kSNN	**90,97**	40	AT	**91,14**	50	AT
Ohsumed	kINN	**69,16**	30	ST	**74,71**	50	ST

Tabela 6.10: *Maiores valores obtidos em $macroF_1$ e $microF_1$ na aplicação dos métodos kNN, kINN e kSNN nas coleções de documentos Reuters, 20NG e Ohsumed.*

As matrizes de similaridade resultantes dos critérios de seleção adotados pelos métodos indicados na Tabela 6.10 foram utilizadas para aplicar a abordagem de geração de características nas coleções correspondentes. Por exemplo, a versão NS da coleção Reuters obteve os maiores valores em $macroF_1$ e $microF_1$ utilizando o método kSNN com $k = 100$ e com $k = 200$, respectivamente. Conforme esses resultados, para gerar características na coleção Reuters foram utilizadas a matriz de similaridade resultante do critério de seleção kSNN (critério utilizado pelo método kSNN) com $k = 100$ e a matriz de similaridade resultante desse critério com $k = 200$.

A coleção expandida (SBOW) resultante da aplicação da abordagem de geração de características em determinada coleção foi denominada da seguinte forma: **[nome da coleção]-[versão da coleção]-[valor de k][método adotado]**. Por exemplo, ao utilizar a matriz de similaridade resultante do critério de seleção kSNN com $k = 200$ para gerar características na versão NS da coleção Reuters (ou coleção Reuters-NS), a coleção SBOW resultante desse processo foi denominada **Reuters-NS-200SNN**.

As Tabelas 6.11 e 6.12 apresentam os valores obtidos em $macroF_1$ e $microF_1$ ao aplicar o método SVM nas coleções Reuters-NS, 20NG-AT e Ohsumed-ST, antes de aplicar a abordagem de geração de características e após aplicar essa abordagem. A Tabela 6.11 mostra os resultados obtidos ao utilizar como peso dos novos termos o maior peso entre os termos originais na coleção utilizada conforme a medida TF (peso máximo) e a Tabela 6.12 mostra os resultados obtidos ao utilizar o peso 1 para os novos termos em cada coleção. Em cada tabela, a coluna *Ganho sobre a coleção de origem* mostra a diferença de desempenho entre o método SVM antes de aplicar a abordagem de geração de características e o método SVM após aplicar essa abordagem. Nessa coluna, valores negativos representam perdas e valores positivos representam ganhos percentuais em relação à coleção de origem e as figuras ▲, ■ e ● significam, respectivamente, que o ganho ou perda apresentado foi fortemente significativo ($\geq 98\%$), significativo ($90\% \leq x < 98\%$) ou não significativo, conforme o teste Wilcoxon [127].

Coleção	Método	$macroF_1$	$microF_1$	Ganho sobre a coleção de origem	
				$macroF_1$	$microF_1$
Reuters-NS	SVM	**92,46**	**96,89**	–	–
Reuters-NS-100SNN	SVM	91,79	95,55	-0,72 ●	-1,38 ▲
Reuters-NS-200SNN	SVM	92,10	96,30	-0,39 ●	-0,61 ■
20NG-AT	SVM	**90,41**	**90,63**	–	–
20NG-AT-40SNN	SVM	85,51	85,55	-5,42 ▲	-5,61 ▲
20NG-AT-50SNN	SVM	86,08	86,10	-4,79 ▲	-5,00 ▲
Ohsumed-ST	SVM	**68,80**	**75,32**	–	–
Ohsumed-ST-30INN	SVM	61,30	67,34	-10,90 ▲	-10,59 ▲
Ohsumed-ST-50INN	SVM	61,36	67,90	-10,81 ▲	-9,85 ▲

Tabela 6.11: *Ganhos obtidos em $macroF_1$ e $microF_1$ ao aplicar a abordagem de geração de características com peso máximo nas coleções Reuters-NS, 20NG-AT e Ohsumed-ST.*

De acordo com os resultados apresentados nas Tabelas 6.11 e 6.12, ao aplicar a abordagem de geração de características com peso máximo nas coleções de documentos Reuters-NS, 20NG-AT e Ohsumed-ST, essa abordagem obteve perdas em todos os cenários. Entretanto, ao aplicar a abordagem de geração de características com peso 1 nessas mesmas coleções, essa abordagem obteve ganhos estatisticamente significativos

Coleção	Método	$macroF_1$	$microF_1$	Ganho sobre a coleção de origem	
				$macroF_1$	$microF_1$
Reuters-NS	SVM	92,46	96,89	–	–
Reuters-NS-100SNN	SVM	93,04	97,34	0,63 ●	0,46 ▲
Reuters-NS-200SNN	SVM	**93,16**	**97,39**	0,76 ●	0,52 ▲
20NG-AT	SVM	90,41	90,63	–	–
20NG-AT-40SNN	SVM	91,68	91,88	1,40 ▲	1,38 ▲
20NG-AT-50SNN	SVM	**91,85**	**92,04**	1,59 ▲	1,56 ▲
Ohsumed-ST	SVM	68,80	75,32	–	–
Ohsumed-ST-30INN	SVM	70,17	76,07	1,99 ▲	1,00 ▲
Ohsumed-ST-50INN	SVM	**70,94**	**76,50**	3,11 ▲	1,57 ▲

Tabela 6.12: *Ganhos obtidos em $macroF_1$ e $microF_1$ ao aplicar a abordagem de geração de características com peso 1 nas coleções Reuters-NS, 20NG-AT e Ohsumed-ST.*

em quase todos os cenários. Nas coleções 20NG-AT e Ohsumed-ST, os ganhos alcançados em $macroF_1$ e $microF_1$ foram maiores ou iquais a 1%, com destaque para o ganho de $3,11\%$ em $macroF_1$ na coleção Ohsumed-ST. Já na coleção Reuters-NS, apesar do ganho em $microF_1$ ter ficado em torno de $0,5\%$, esse ganho é importante devido essa coleção já possuir um valor de $microF_1$ elevado antes de aplicar essa abordagem. Com o objetivo de analisar comparativamente a importância dos novos termos, foi conduzido um estudo descrito na próxima seção.

6.2.1 Análise dos melhores termos

A análise dos melhores termos teve como objetivo coletar os termos mais importantes de uma coleção, armazená-los em uma lista L, com elementos ordenados conforme o ganho de informação, dividir essa lista em dois conjuntos, o conjunto de termos novos e o conjunto de termos originais, e identificar qual desses conjuntos é mais importante para discriminar os documentos dessa coleção.

Para realizar essa análise foram utilizadas as coleções de documentos resultantes da aplicação da abordagem de geração de características com peso 1. A versão com peso 1 dessas coleções foi denominada FC1. Além disso, para cada coleção da versão FC1, os 1000 termos com os maiores ganhos de informação foram coletados e armazenados no conjunto Q_n, se o termo foi resultante da aplicação da abordagem de geração de características na coleção, ou no conjunto Q_o, se o termo for resultante da representação BOW.

A média mútua do *ranking* (MRR, do inglês *Mean Reciprocal Rank*) dos conjuntos Q_n e Q_o foi calculada para verificar qual desses conjuntos é mais importante para discriminar os documentos de uma coleção. Essa medida possibilita mostrar o conjunto

que possui, em média, maior ganho de informação levando em consideração a posição de um termo no *ranking* de termos. A MRR foi calculada pela Equação 6-1.

$$MRR = \frac{1}{min(Q_n, Q_o)} \sum_{i=1}^{min(Q_n,Q_o)} \frac{1}{rank_i} \tag{6-1}$$

onde Q_o é o conjunto de termos originais, Q_n é o conjunto de termos novos, $min(Q_n, Q_o)$ é o tamanho do conjunto Q_n ou Q_o com a menor quantidade de elementos, L é a lista de termos em ordem decrescente do ganho de informação de uma coleção e $rank_i$ é a posição do termo t_i na lista L.

A Tabela 6.13 mostra a MRR dos conjuntos Q_n e Q_o nas coleções de documentos da versão FC1.

Coleção	$\|Q_n\|$	$\|Q_o\|$	MRR	
			Q_n	Q_o
Reuters-NS-100SNN	6227	3773	$7{,}2237.10^{-4}$	$1{,}4019.10^{-3}$
Reuters-NS-200SNN	6940	3060	$8{,}7652.10^{-2}$	$1{,}2106.10^{-1}$
20NG-AT-40SNN	5897	4103	$2{,}1204.10^{-4}$	$2{,}0807.10^{-3}$
20NG-AT-50SNN	6512	3488	$2{,}3920.10^{-4}$	$2{,}3595.10^{-3}$
Ohsumed-ST-30INN	8079	1921	$2{,}8568.10^{-4}$	$3{,}8935.10^{-3}$
Ohsumed-ST-50INN	8717	1283	$3{,}4842.10^{-4}$	$5{,}2614.10^{-3}$

Tabela 6.13: *Valores da MRR dos conjuntos Q_n e Q_o nas coleções de documentos FC1.*

Conforme os valores mostrados na Tabela 6.13, é possível concluir que os termos mais discriminativos correspondem ao termos originais das coleções. Entretanto, os resultados mostrados na Tabela 6.12 e a quantidade de termos novos que aparecem em Q_n indicam que alguns termos novos, apesar de não serem os mais discriminativos, podem contribuir para aumentar o desempenho do classificador SVM, dependendo do valor de seus pesos. Um estudo mais aprofundado sobre como selecionar os melhores termos novos e de como atribuir pesos a eles pode ser importante para melhorar o desempenho do classificador SVM. Essa hipótese é justificada principalmente no caso da coleção Ohsumed em que é grande o número de termos novos com alto valor de ganho de informação. Isto pode ser verificado pelo grande número de termos novos em Q_n para essa coleção.

CAPÍTULO 7

Conclusão e Trabalhos Futuros

Neste trabalho realizou-se um estudo sobre duas abordagens na tentativa de aumentar a eficácia da atividade de classificação automática de textos (CAT). Na primeira abordagem foram propostas duas variações do método kNN (método kINN e método kSNN), ambas em relação ao critério de seleção adotado, e na segunda abordagem foi proposta uma técnica para a geração de características em documentos baseada nos critérios de seleção propostos.

Em relação às duas variações do método kNN propostas, as hipóteses formuladas foram as seguintes:

- Selecionar os documentos de treino que possuem o documento de teste d entre os seus k vizinhos mais próximos pode gerar mais vizinhos do documento d que o critério de seleção tradicionalmente utilizado pelo método kNN e, portanto, esse novo critério é mais confiável que o critério utilizado pelo kNN, dado que a decisão quanto à categoria do documento d se baseia em uma quantidade maior de documentos de treino.
- A combinação do critério sugerido na primeira hipótese com o critério tradicional utilizado pelo método kNN possibilita selecionar os vizinhos "mais similares" ao documento de teste d, proporcionando uma decisão mais confiável em relação à categoria desse documento.

Para verificar ou refutar essas duas hipóteses, os métodos de classificação propostos neste trabalho (método kINN e kSNN) foram aplicados nas versões AT, NS, ST e FS das coleções Reuters, 20NG e Ohsumed. Os resultados obtidos em $macroF_1$ e $microF_1$ pelos métodos propostos foram comparados com os resultados obtidos pelo método kNN, sendo possível verificar que os métodos kINN e kSNN (que utilizam, respectivamente, o critério de seleção kINN e o critério de seleção kSNN) foram mais confiáveis que o método kNN (que utiliza o critério de seleção kNN) nas versões AT, NS e ST da coleção Reuters e nas versões AT, NS, ST e FS da coleção Ohsumed. Além disso, o método kSNN não apresentou perda em $macroF_1$ ou $microF_1$ sobre o método kNN em nenhuma versão das coleções Reuters, 20NG e Ohsumed. Ao aplicar os métodos kNN, kINN e KSNN na

coleção 20NG, com exceção da versão FS, os resultados obtidos em $macroF_1$ e $microF_1$ foram semelhantes entre os métodos e a hipótese que os critérios de seleção kINN e kSNN são mais confiáveis que o critério kNN foi refutada. Os ganhos obtidos com os métodos propostos, apesar de pequenos, são interessantes pois os melhores resultados se aproximaram do método SVM, considerado estado da arte na classificação automática de textos. A Tabela 7.1 mostra os ganhos em $macroF_1$ e $microF_1$ dos métodos propostos sobre o método SVM.

Coleção	Método	Melhores resultados		Ganho sobre o SVM	
		$macroF_1$	$microF_1$	$macroF_1$	$microF_1$
Reuters	SVM	92,46	96,89	–	–
	kINN	90,24	95,01	-2,40	-1,94
	kSNN	90,44	95,56	-2,18	-1,37
20NG	SVM	90,41	90,63	–	–
	kINN	90,94	91,09	0,59	0,51
	kSNN	90,97	91,14	0,62	0,56
Ohsumed	SVM	68,80	75,32	–	–
	kINN	69,16	74,71	0,52	-0,81
	kSNN	68,78	74,28	-0,03	-1,38

Tabela 7.1: *Ganhos obtidos em $macroF_1$ e $microF_1$ sobre o método SVM ao aplicar os métodos kINN ou kSNN nas coleções Reuters, 20NG e Ohsumed.*

Em relação ao algoritmo de seleção de características adotado na versão FS das coleções Reuters, 20NG e Ohsumed, é possível que ele tenha removido, além de ruídos, características boas para discriminar documentos. Outras técnicas de seleção de características, tais como algoritmos que utilizam uma função critério dependente, poderiam ser exploradas na tentativa de aumentar a eficácia da classificação utilizando os métodos kNN, kINN e kSNN.

Um aspecto importante quanto à eficácia dos métodos kNN, kINN, kSNN é a escolha do parâmetro k. Com o objetivo de identificar qual método (kNN, kINN ou kSNN) foi menos sensível à variação do valor desse parâmetro, foram realizados experimentos que verificaram que o método kNN foi menos sensível em $microF_1$ à variação do valor de k que os métodos kINN e kSNN. Em $macroF_1$, somente na coleção Ohsumed-ST que o método kINN foi menos sensível à variação do valor de k que os demais métodos (kNN e kSNN). Além disso, o método kSNN apresentou-se o método mais sensível à variação do valor de k nas coleções Reuters-NS, 20NG-AT e Ohsumed-ST.

A segunda abordagem estudada neste trabalho consistiu em gerar novas características na representação conjunto de palavras (BOW, do inglês *bag of words*) das coleções Reuters, 20NG e Oshumed. As novas características corresponderam aos identificadores dos documentos da matriz de similaridade (documentos 'vizinhos mais próximos') resultante do critério de seleção utilizado pelo método que alcançou o maior valor em $macroF_1$

ou $microF_1$ ao ser aplicado nas versões AT, NS, ST e FS de determinada coleção. Os novos termos gerados tiveram peso 1 ou peso máximo, conforme explicado na descrição da abordagem proposta (Seção 6.2).

Em relação à abordagem de geração de características proposta neste trabalho, nos experimentos realizados utilizando essa abordagem, ao aplicar o peso 1 às novas características geradas nas coleções de documentos Reuters-NS, 20NG-AT e Ohsumed-ST, as coleções resultantes tiveram ganhos estatisticamente significativos. Dessa forma, a hipótese que a coleção resultante da aplicação dessa abordagem poderia aumentar a eficácia da CAT foi verificada nas três coleções. Entretanto, ao aplicar o peso máximo às novas características, essa hipótese foi refutada nessas três coleções. Esses resultados sugerem um estudo mais aprofundado sobre pesos apropriados para as novas características na tentativa de melhorar o desempenho do classificador.

Para verificar os termos que foram mais discriminativos nas coleções de documentos Reuters-NS, 20NG-AT e Ohsumed-ST após aplicar a abordagem de geração de característica proposta, foram realizados experimentos que indicaram que os termos originais das coleções foram mais discriminativos que às características geradas após aplicar essa abordagem. Apesar desses resultados, entre os 10000 termos mais discriminativos, grande parte desses termos eram características novas, o que levanta a hipótese que um estudo mais aprofundado sobre como selecionar os melhores termos novos e de como atribuir pesos a eles pode ser importante para melhorar o desempenho do classificador SVM.

7.1 Trabalhos Futuros

Além dos estudos realizados neste trabalho, como trabalho futuro pretende-se realizar os seguintes estudos:

1. Investigação das abordagens propostas em outros contextos de classificação de documentos, por exemplo, em coleções que tenham *links* entre documentos, tais como as coleções de páginas *Web*.
2. Investigação das abordagens propostas em outras aplicações de classificação que não seja classificação de texto, tais como bancos de dados de empresas e imagens de satélites ou médicas.
3. Investigação de métodos para selecionar as novas características criadas com a abordagem de geração de características proposta no trabalho, bem como estudar outras formas de atribuir peso às características, como por exemplo, atribuir o valor do ganho de informação às características.
4. Investigação da eficiência dos métodos propostos.

Referências Bibliográficas

[1] ACHTERT, E.; BÖHM, C.; KRÖGER, P.; KUNATH, P.; PRYAKHIN, A.; RENZ, M. **Efficient reverse k-nearest neighbor estimation**. *Informatik-Forschung und Entwicklung*, 21(3):179–195, 2007.

[2] AHA, D. **Lazy learning**. Kluwer Academic Publishers Norwell, MA, USA, 1997.

[3] ALMUALLIM, H.; DIETTERICH, T. **Learning with many irrelevant features**. In: *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, volume 2, p. 547–552. AAAI Press, 1991.

[4] AMALDI, E.; KANN, V. **On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems**. *Theoretical Computer Science*, 209(1-2):237–260, 1998.

[5] ANDO, R.; ZHANG, T. **A framework for learning predictive structures from multiple tasks and unlabeled data**. *The Journal of Machine Learning Research*, 6:1817–1853, 2005.

[6] ANDO, R.; ZHANG, T. **A high-performance semi-supervised learning method for text chunking**. In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, p. 1–9. Association for Computational Linguistics Morristown, NJ, USA, 2005.

[7] ANDROUTSOPOULOS, I.; KOUTSIAS, J.; CHANDRINOS, K.; SPYROPOULOS, C. **An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages**. In: *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 160–167. ACM New York, NY, USA, 2000.

[8] ANTONELLIS, I.; GALLOPOULOS, E. **Exploring term-document matrices from matrix models in text mining**. In: *SIAM Conference Data Mining*, 2006.

[9] BAEZA-YATES, R. A.; RIBEIRO-NETO, B. **Modern information retrieval**. Addison-Wesley Longman Publishing Co., Inc., 1999.

[10] BAGALLO, G.; HAUSSLER, D. **Boolean feature discovery in empirical learning**. *Machine Learning*, 5(1):71–99, 1990.

[11] BAOLI, L.; SHIWEN, Y. **An adaptive k-nearest neighbor text categorization strategy**. In: *ACM Transactions on Asian Language Information Processing*, volume 3, p. 215–226, 2004.

[12] BASILI, R.; MOSCHITTI, A.; PAZIENZA, M. **Language-sensitive text classification**. In: *Proceedings of RIAO-00, 6th International Conference Recherche d'Information Assistee par Ordinateur*, p. 331–343, 2000.

[13] BEKKERMAN, R.; EL-YANIV, R.; TISHBY, N.; WINTER, Y. **On feature distributional clustering for text categorization**. In: *Proceedings of the 24th Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, p. 146–153. ACM New York, NY, USA, 2001.

[14] BEKKERMAN, R.; EL-YANIV, R.; TISHBY, N.; WINTER, Y. **Distributional word clusters vs. words for text categorization**. *The Journal of Machine Learning Research*, 3:1183–1208, 2003.

[15] BELLMAN, R. **Adaptive control processes: a guided tour**. Princeton University Press, 1961.

[16] BELONGIE, S.; MALIK, J.; PUZICHA, J. **Shape matching and object recognition using shape contexts**. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 24, p. 509–522, 2002.

[17] BENNETT, P.; DUMAIS, S.; HORVITZ, E. **Inductive transfer for text classification using generalized reliability indicators**. In: *Proceedings of the ICML-2003 Workshop on*, 2003.

[18] BLAKE, C.; PRATT, W. **Better rules, fewer features: a semantic approach to selecting features from text**. In: *Proceedings of the IEEE Data Mining Conference*, p. 59–66, 2001.

[19] BLUM, A.; LANGLEY, P. **Selection of relevant features and examples in machine learning**. *Artificial Intelligence*, 97(1-2):245–271, 1997.

[20] BRANK, J.; GROBELNIK, M. **Interaction of feature selection methods and linear classification models**. In: *In Proceedings of the ICML-02 Workshop on Text Learning*, 2002.

[21] BURGES, C. **A tutorial on support vector machines for pattern recognition**. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[22] CAROPRESO, M. **A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization**. In: *Text Databases and Document Management: Theory and Practice*, p. 78. IGI Global, 2001.

[23] CHOI, B.; YAO, Z. **Web page classification**. *Foundations and Advances in Data Mining*, p. 221, 2005.

[24] CIMIANO, P.; HOTHO, A.; STAAB, S. **Learning concept hierarchies from text corpora using formal concept analysis**. *Journal of Artificial Intelligence Research*, 24:305–339, 2005.

[25] COHEN, W. **Automatically extracting features for concept learning from the web**. In: *Proceedings of the Seventeenth International Conference on Machine Learning*, p. 159–166. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2000.

[26] CORMEN, T. **Introduction to algorithms**. MIT press, 2001.

[27] COUTO, T.; CRISTO, M.; GONÇALVES, M.; CALADO, P.; ZIVIANI, N.; MOURA, E.; RIBEIRO-NETO, B. **A comparative study of citations and links in document classification**. In: *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, p. 75–84. ACM New York, NY, USA, 2006.

[28] COVER, T.; HART, P. **Nearest neighbor pattern classification**. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[29] CRISTIANINI, N.; SHAWE-TAYLOR, J. **An introduction to support Vector Machines: and other kernel-based learning methods**. Cambridge Univ Pr, 2000.

[30] DEERWESTER, S.; DUMAIS, S.; FURNAS, G.; LANDAUER, T.; HARSHMAN, R. **Indexing by latent semantic analysis**. *Journal of the American Society for Information Science and Technology*, 41(6):391–407, 1990.

[31] DO, C.; NG, A. **Transfer learning for text classification**. *Advances in Neural Information Processing Systems*, 18:299, 2006.

[32] DOMENICONI, C.; GUNOPULOS, D. **Adaptive nearest neighbor classification using support vector machines**. *Advances in Neural Information Processing Systems*, p. 665–672, 2002.

[33] DRUCKER, H.; VAPNIK, V.; WU, D. **Automatic text categorization and its applications to text retrieval**. *IEEE Transactions on Neural Network*, 10(5):1048–1054, 1999.

[34] DUDA, R.; HART, P.; STORK, D. **Pattern classification**. Wiley New York, 2001.

[35] DUMAIS, S.; CHEN, H. **Hierarchical classification of web content**. In: *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 256–263. ACM New York, NY, USA, 2000.

[36] DUMAIS, S.; PLATT, J.; HECKERMAN, D.; SAHAMI, M. **Inductive learning algorithms and representations for text categorization**. In: *Proceedings of the Seventh International Conference on Information and Knowledge Management*, p. 148–155, 1998.

[37] FAWCETT, C.; TOM, E. **Feature discovery for problem solving systems**. PhD thesis, Doctoral dissertation, Department of Computer Science, University of Massachusetts, Amherst, MA, 1993.

[38] FELLBAUM, C.; OTHERS. **WordNet: An electronic lexical database**. MIT press Cambridge, MA, 1998.

[39] FINKELSTEIN, L.; GABRILOVICH, E.; MATIAS, Y.; RIVLIN, E.; SOLAN, Z.; WOLFMAN, G.; RUPPIN, E. **Placing search in context: The concept revisited**. *ACM Transactions on Information Systems*, 20(1):116–131, 2002.

[40] FORMAN, G. **An extensive empirical study of feature selection metrics for text classification**. *The Journal of Machine Learning Research*, 3:1289–1305, 2003.

[41] FORMAN, G. **Feature selection for text classification**. *Computational Methods of Feature Selection*, p. 257, 2007.

[42] FRAKES, W.; BAEZA-YATES, R. **Information retrieval: data structures and algorithms**. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1992.

[43] FUHR, N. **A probabilistic model of dictionary based automatic indexing**. Techn. Hochsch. Darmstadt, Fachbereich Informatik, 1985.

[44] FUHR, N.; BUCKLEY, C. **A probabilistic learning approach for document indexing**. *ACM Transactions on Information Systems (TOIS)*, 9(3):223–24S, 1991.

[45] FURNKRANZ, J.; MITCHELL, T.; RILOFF, E. **A case study in using linguistic phrases for text categorization on the www**. In: *In Working Notes of the AAAI/ICML Workshop on Learning for Text Categorization*, 1998.

[46] GLOVER, E.; TSIOUTSIOULIKLIS, K.; LAWRENCE, S.; PENNOCK, D.; FLAKE, G. **Using web structure for classifying and describing web pages**. In: *Proceedings of the eleventh international conference on World Wide Web*, p. 562–569. ACM Press, 2002.

[47] GOLDBERG, A.; ZHU, X. **Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization**. In: *HLT-NAACL 2006 Workshop on Textgraphs: Graphbased Algorithms for Natural Language Processing*, 2006.

[48] GOLDBERGER, J.; ROWEIS, S.; HINTON, G.; SALAKHUTDINOV, R. **Neighbourhood components analysis**. *Advances in Neural Information Processing Systems*, 17:513–520, 2005.

[49] GOLUB, K. **Automated subject classification of textual web documents**. *Journal of Documentation*, 62(3):350–371, 2006.

[50] GOVERT, N.; LALMAS, M.; FUHR, N. **A probabilistic description-oriented approach for categorizing web documents**. In: *Proceedings of the Eighth International Conference on Information and Knowledge Management*, p. 475–482. ACM New York, NY, USA, 1999.

[51] GUYON, I.; ELISSEEFF, A. **An introduction to variable and feature selection**. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[52] HALL, M. **Correlation-based feature selection for machine learning**. PhD thesis, The University of Waikato, 1999.

[53] HALL, M. **Correlation-based feature selection for discrete and numeric class machine learning**. In: *Proceedings of the Seventeenth International Conference on Machine Learning*, p. 359–366. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 2000.

[54] HASTIE, T.; TIBSHIRANI, R. **Discriminant adaptive nearest neighbor classification**. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 18, p. 607–616, 1996.

[55] HAYKIN, S.; ENGEL, P. **Redes neurais: princípios e prática**. Bookman, 2001.

[56] HU, Y.; KIBLER, D. **A wrapper approach for constructive induction**. In: *The Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, 1996.

[57] JAIN, A.; DUIN, R.; MAO, J. **Statistical pattern recognition: A review**. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 4–37. IEEE Computer Society, 2000.

[58] JIANG, L.; ZHANG, H.; CAI, Z. **Dynamic k-nearest-neighbor naive bayes with attribute weighted**. *Lecture Notes in Computer Science*, 4223:365, 2006.

[59] JOACHIMS, T.; NEDELLEC, C.; ROUVEIROL, C. **Text categorization with support vector machines: learning with many relevant**. In: *Machine Learning: ECML-98 10th European Conference on Machine Learning, Chemnitz, Germany*. Springer, 1998.

[60] JONES, S. **A statistical interpretation of term specificity and its application in retrieval**. *Journal of Documentation*, 28(1):11–20, 1972.

[61] KEHAGIAS, A.; PETRIDIS, V.; KABURLASOS, V.; FRAGKOU, P. **A comparison of word-and sense-based text categorization using several classification algorithms**. *Journal of Intelligent Information Systems*, 21(3):227–247, 2003.

[62] KITTLER, J. **Une generalisation de quelques algorithms sous-optimaux de recherche d'ensembles d'attributs**. In: *Proceedings Congres Reconnaissance des Formes et Traitement des Images*, 1978.

[63] KOHAVI, R.; JOHN, G. **Wrappers for feature subset selection**. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[64] KONONENKO, I. **Comparison of inductive and naive bayesian learning approaches to automatic knowledge acquisition**. *Current Trends in Knowledge Acquisition*, p. 190–197, 1990.

[65] KORN, F.; MUTHUKRISHNAN, S. **Influence sets based on reverse nearest neighbor queries**. *ACM SIGMOD Record*, 29(2):201–212, 2000.

[66] KRISHNAIAH, P.; KANAL, L. **Classification, pattern recognition and reduction of dimensionality**. Amsterdam: North-Holland, 1982.

[67] KRUPKA, E.; NAVOT, A.; TISHBY, N. **Learning to select features using their properties**. *Journal of Machine Learning Research*, 9:2349–2376, 2008.

[68] KRUPKA, E.; TISHBY, N. **Incorporating prior knowledge on features into learning**. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.

[69] KUDENKO, D.; HIRSH, H. **Feature generation for sequence categorization**. In: *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, p. 733–738, 1998.

[70] KUDO, M.; SKLANSKY, J. **Comparison of algorithms that select features for pattern classifiers**. *Pattern Recognition*, 33(1):25–41, 2000.

[71] KUMARAN, G.; ALLAN, J. **Text classification and named entities for new event detection**. In: *Proceedings of the 27th Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, p. 297–304. ACM New York, NY, USA, 2004.

[72] LEWIS, D. D. **Reuters-21578 text categorization colletion**. http://www.daviddlewis.com/resources/testcollections/reuters21578/, último acesso em Set de 2009.

[73] LEWIS, D. **Evaluating text categorization**. In: *Proceedings of Speech and Natural Language Workshop*, p. 312–318. Morgan Kaufmann, 1991.

[74] LEWIS, D. **An evaluation of phrasal and clustered representations on a text categorization task**. In: *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 37–50. ACM New York, NY, USA, 1992.

[75] LEWIS, D.; CROFT, W. **Term clustering of syntactic phrases**. In: *Proceedings of the 13th Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, p. 385–404. ACM New York, NY, USA, 1989.

[76] LIU, H.; MOTODA, H. **Feature selection for knowledge discovery and data mining**. Springer, 1998.

[77] LIU, H.; YU, L. **Toward integrating feature selection algorithms for classification and clustering**. In: *IEEE Transactions on Knowledge and Data Engineering*, volume 17, p. 491–502, 2005.

[78] LIU, T.; CHEN, Z.; ZHANG, B.; MA, W.; WU, G. **Improving text classification using local latent semantic indexing**. In: *Proceedings of the Fourth IEEE International Conference on Data Mining*, p. 162–169. IEEE Computer Society Washington, DC, USA, 2004.

[79] LOVINS, J. **Development of a Stemming Algorithm**. Technical report, DTIC Research Report AD0735504, 1968.

[80] MANNING, C.; RAGHAVAN, P.; SCHÜTZE, H. **Introduction to information retrieval**. Cambridge University Press, 2008.

[81] MARILL, T.; GREEN, D. **On the effectiveness of receptors in recognition systems**. In: *IEEE transactions on Information Theory*, volume 9, p. 11–17, 1963.

[82] MARKOVITCH, S.; ROSENSTEIN, D. **Feature generation using general constructor functions**. *Machine Learning*, 49(1):59–98, 2002.

[83] MATHEUS, C. **The need for constructive induction**. In: *Machine Learning: Proceedings of the International Conference*, p. 173. Morgan Kaufmann Publishers, 1990.

[84] MATHEUS, C.; RENDELL, L. **Constructive induction on decision trees**. In: *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, volume 645650. Morgan Kaufmann, 1989.

[85] MIKHEEV, A. **Feature lattices for maximum entropy modelling**. In: *Proceedings of the 17th international conference on Computational linguistics*, p. 848–854. Association for Computational Linguistics Morristown, NJ, USA, 1998.

[86] MITCHEL, T. **Machine learning**, volume 48. WCB McGraw Hill, 1997.

[87] MITRA, M.; BUCKLEY, C.; SINGHAL, A.; CARDIE, C. **An analysis of statistical and syntactic phrases**. In: *Fifth RIAO Conference, Computer-Assisted Information Searching on the Internet*, 1997.

[88] MITRA, M.; SINGHAL, A.; BUCKLEY, C. **Improving automatic query expansion**. In: *Proceedings of the 21st Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, p. 206–214. ACM New York, NY, USA, 1998.

[89] MLADENIC, D. **Feature subset selection in text learning**. *Lecture Notes in Computer Science*, 1398:95–100, 1998.

[90] MLADENIC, D. **Turning Yahoo! into an automatic web-page classifier**. In: *13th Europearn Conference on Artificial Intelligence (ECAI98)*, volume 474, 1998.

[91] MLADENIC, D.; GROBELNIK, M. **Feature selection for classification based on text hierarchy**. In: *Proceedings of the Workshop on Learning from Text and the Web*, 1998.

[92] MLADENIC, D.; GROBELNIK, M. **Word sequences as features in text-learning**. In: *Proceedings of ERK-98, the Seventh Electrotechnical and Computer Science Conference*, p. 145–148, 1998.

[93] MURPHY, P.; PAZZANI, M. **ID2-of-3: Constructive induction of M-of-N concepts for discriminators in decision trees**. In: *Proceedings of the Eighth International Workshop on Machine Learning*, p. 183–187. Morgan Kaufmann, 1991.

[94] NARENDRA, P.; FUKUNAGA, K. **A branch and bound algorithm for feature subset selection**. *IEEE Transactions on Computers*, 100(26):917–922, 1977.

[95] ORENGO, V.; HUYCK, C. **A Stemming algorithm for the Portuguese Language**. In: *Proceedings of SPIRE-2001 Symposium on String Processing and Information Retrieval*, 2001.

[96] PENG, F.; SCHUURMANS, D. **Combining naive Bayes and n-gram language models for text classification**. *Lecture notes in computer science*, p. 335–350, 2003.

[97] PENG, F.; SCHUURMANS, D.; WANG, S. **Augmenting naive bayes classifiers with statistical language models**. *Information Retrieval*, 7(3):317–345, 2004.

[98] PENG, J.; HEISTERKAMP, D.; DAI, H. **Adaptive kernel metric nearest neighbor classification**. *International Conference on Pattern Recognition*, 16:33–36, 2002.

[99] PEREIRA, F.; TISHBY, N.; LEE, L. **Distributional clustering of English words**. In: *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, p. 183–190. Association for Computational Linguistics Morristown, NJ, USA, 1993.

[100] PORTER, M. **An algorithm for suffix stripping program**. *Program*, 14(3):130–137, 1980.

[101] QI, X.; DAVISON, B. **Web page classification: features and algorithms**. *ACM Computing Surveys (CSUR)*, 2009.

[102] QUINLAN, J. **Induction of decision trees**. *Machine learning*, 1(1):81–106, 1986.

[103] RASKUTTI, B.; FERRA, H.; KOWALCZYK, A. **Second order features for maximising text classification performance**. *Lecture notes in computer science*, p. 419–430, 2001.

[104] RUTHVEN, I.; LALMAS, M. **A survey on the use of relevance feedback for information access systems**. *The Knowledge Engineering Review*, 18(02):95–145, 2003.

[105] SABLE, C.; MCKEOWN, K.; CHURCH, K. **NLP found helpful (at least for one text categorization task)**. In: *Proceedings of the Acl-02 Conference on Empirical Methods in Natural Language Processing*, p. 172–179. Association for Computational Linguistics Morristown, NJ, USA, 2002.

[106] SAHAMI, M.; DUMAIS, S.; HECKERMAN, D.; HORVITZ, E. **A Bayesian approach to filtering junk e-mail**. In: *Learning for Text Categorization: Papers from the 1998 Workshop*, volume 62, p. 98–05. Madison, Wisconsin: AAAI Technical Report WS-98-05, 1998.

[107] SALTON, G. **Automatic text processing**. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1988.

[108] SALTON, G.; BUCKLEY, C. **Term-weighting approaches in automatic text retrieval**. *Information Processing and Management*, 24(5):513–523, 1988.

[109] SASSANO, M. **Virtual examples for text classification with support vector machines**. In: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language*, p. 208–215. Association for Computational Linguistics Morristown, NJ, USA, 2003.

[110] SCHOLKOPF, B.; SMOLA, A. **Learning with kernels**. MIT press Cambridge, Mass, 2002.

[111] SCULLEY, D.; WACHMAN, G. **Relaxed online SVMs for spam filtering**. In: *Proceedings of the 30th Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, p. 415–422. ACM New York, NY, USA, 2007.

[112] SEBASTIANI, F. **Machine learning in automated text categorization**. *ACM Computing Surveys (CSUR)*, 34(1):1–47, 2002.

[113] SHAKHNAROVICH, G.; DARRELL, T.; INDYK, P. **Nearest-neighbor methods in learning and vision: Theory and practice**. MIT Press, 2005.

[114] SHALEV-SHWARTZ, S.; SINGER, Y.; NG, A. **Online and batch learning of pseudo-metrics**. In: *Proceedings of the Twenty-first International Conference on Machine Learning*. ACM New York, NY, USA, 2004.

[115] SIMARD, P.; LECUN, Y.; DENKER, J. **Efficient pattern recognition using a new transformation distance**. In: *Advances in Neural Information Processing Systems*, p. 50–58. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1992.

[116] SUSSMANN, H.; KOKOTOVIC, P. **The Peaking Phenomenon And The Global Stabilization Of Nonlinearsystems**. *IEEE Transactions on Automatic Control*, 36(4):424–440, 1991.

[117] TASKAR, B.; WONG, M.; KOLLER, D. **Learning on the test data: Leveraging unseen features**. In: *International Conference on Machine Learning (ICML)*, volume 20, p. 744, 2003.

[118] TISHBY, N.; PEREIRA, F.; BIALEK, W. **The information bottleneck method**. In: *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing*, p. 368–377, 1999.

[119] VALIANT, L. **A theory of the learnable**. *Communications of the ACM*, 27(11):1134–1142, 1984.

[120] VAPNIK, V. **Statistical learning theory**. John Wiley & Sons, 1998.

[121] VAPNIK, V. **The nature of statistical learning theory**. Springer-Verlag, 1995.

[122] VAPNIK, V. **An overview of statistical learning theory**. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[123] VOORHEES, E. **Query expansion using lexical-semantic relations**. In: *Proceedings of the 17th Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, p. 61–69. Springer-Verlag New York, Inc. New York, NY, USA, 1994.

[124] VOORHEES, E. **Using wordnet for text retrieval**. *WordNet: an electronic lexical database*, p. 285–303, 1998.

[125] WATERMAN, D. **A guide to expert systems**. Addison Wesley, 1985.

[126] WEINBERGER, K.; BLITZER, J.; SAUL, L. **Distance metric learning for large margin nearest neighbor classification**. *Advances in neural information processing systems*, 18:1473, 2006.

[127] WILCOXON, F. **Individual comparisons by ranking methods**. *Biometrics Bulletin*, p. 80–83, 1945.

[128] WU, H.; GUNOPULOS, D. **Evaluating the utility of statistical phrases and latent semantic indexing for text classification**. In: *Proceedings of the IEEE International Conference on Data Mining*, p. 713–716, 2002.

[129] XIE, Z.; HSU, W.; LIU, Z.; LI LEE, M. **SNNB: A selective neighborhood based naive bayes for lazy learning**. *Lecture notes in computer science*, p. 104–114, 2002.

[130] XU, J.; CROFT, W. **Query expansion using local and global document analysis**. In: *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 4–11, 1996.

[131] YANG, Y. **Expert network: effective and efficient learning from human decisions in text categorization and retrieval**. In: *Proceedings of the 17th Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, p. 13–22. Springer-Verlag New York, Inc. New York, NY, USA, 1994.

[132] YANG, Y.; AULT, T.; PIERCE, T.; LATTIMER, C. **Improving text categorization methods for event tracking**. In: *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 65–72. ACM New York, NY, USA, 2000.

[133] YANG, Y.; CHUTE, C. **An example-based mapping method for text categorization and retrieval**. *ACM Transactions on Information Systems (TOIS)*, 12(3):252–277, 1994.

[134] YANG, Y.; LIU, X. **A re-examination of text categorization methods**. In: *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, p. 42–49. ACM New York, NY, USA, 1999.

[135] YANG, Y.; PEDERSEN, J. **A comparative study on feature selection in text categorization**. In: *Machine Learning-international Workshop Then Conference*, p. 412–420. MORGAN KAUFMANN PUBLISHERS, INC., 1997.

[136] ZELIKOVITZ, S.; HIRSH, H. **Improving short text classification using unlabeled background knowledge to assess document similarity**. In: *Proceedings of the Seventeenth International Conference on Machine Learning*, p. 1183–1190, 2000.

[137] ZELIKOVITZ, S.; HIRSH, H. **Using lsi for text classification in the presence of background text**. In: *Proceedings of the tenth international conference on Information and knowledge management*, p. 113–118. ACM New York, NY, USA, 2001.