PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

AGRUPAMENTO E CATEGORIZAÇÃO DE DOCUMENTOS JURÍDICOS

LUIS OTÁVIO DE COLLA FURQUIM

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

Orientadora: Profa. Vera Lúcia Strube de Lima

Porto Alegre
2011

Dados Internacionais de Catalogação na Publicação (CIP)

F989a Furquim, Luis Otávio de Colla
Agrupamento e categorização de documentos jurídicos / Luis Otávio de Colla Furquim. – Porto Alegre, 2011.
146 p.

Diss. (Mestrado) – Fac. de Informática, PUCRS.
Orientador: Profª. Drª. Vera Lúcia Strube de Lima.

1. Informática. 2. Categorização (Linguística).
3. Processamento de Textos (Computação).
4. Algoritmos (Programação). I. Lima, Vera Lúcia Strube de.
II. Título.

CDD 006.35

Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS

Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "**Agrupamento e Categorização de Documentos Jurídicos**", apresentada por Luis Otávio de Colla Furquim, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Inteligência Computacional , aprovada em 25/03/2011 pela Comissão Examinadora:

Vera Lúcia Strube de L...

Profa. Dra. Vera Lúcia Strube de Lima– Orientadora — PPGCC/PUCRS

Prof. Dr. Duncan Dubugras Alcoba Ruiz - — PPGCC/PUCRS

Prof. Dr. Leandro Krug Wives – — UFRGS

Homologada em 22/06/2012, conforme Ata No. 013 pela Comissão Coordenadora.

Prof. Dr. Fernando Luís Dotti
Coordenador.

PUCRS
Campus Central
Av. Ipiranga, 6681 – P32– sala 507 – CEP: 90619-900
Fone: (51) 3320-3611 – Fax (51) 3320–3621
E-mail: ppgcc@pucrs.br
www.pucrs.br/facin/pos

DEDICATÓRIA

À minha sogra, Ísis Palma,
mais uma brasileira que morreu sem ver seu pleito atendido.
E à minha mãe, Daisy de Colla Furquim,
que viveu lutando para ver atendidos os pleitos de inúmeros brasileiros.

AGRADECIMENTOS

A Deus, e todos que com ele estão, que me motiva, ilumina e me dá forças. A cujas bênçãos, tão injustamente, chamamos de “sorte” ou “destino”.

Aos meus amigos que com Ele deixei, que prometeram sustemtar-me a cada passo no projeto da minha vida e que nem por um só momento afastaram-se de sua promessa.

A minha amada esposa, Jussara, e às minhas enteadas Aninha e Mariana e aos meus filhos, Gabriel e Natália, que me apoiam e tanto se sacrificam para que eu possa empreender esta tarefa.

À minha mãe que sempre me incentivou e apoiou em meus estudos e que onde quer que esteja ainda deve estar se sacrificando para me ajudar. E novamente a Deus que está neste momento me ajudando a conter as minhas lágrimas por ela para que eu possa terminar de escrever estas linhas, tão mais difíceis de escrever que o próprio trabalho.

Ao meu pai, imparcial não somente quando julgou processos, mas, também, quando foi parte no processo, me mostrando que se deve fazer o que se acredita ser o correto, doa a quem doer, ainda que doa em si mesmo.

Ao meu amigo Alexandre Fernandes, sempre disposto a discutir cada “piração” que me ocorre à medida que vou estudando a bibliografia pertinente e voluntariamente pesquisando na Internet ferramentas, literatura e linguagens que me auxiliem neste trabalho e em trabalhos futuros.

Aos meus amigos Jorge Lengler e Marcelo Squeff, que nunca me deixam na mão, me socorrendo a qualquer dia e a qualquer hora.

Ao amigos Anderson Burger e Régis Coimbra que com sacrifício pessoal me socorreram e prestaram inestimável auxílio ao meu estudo.

Às amigas Karin Menoncin e Nalin Ferreira, que imensamente contribuíram para o sucesso deste estudo.

Aos meus amigos no Ministério Público Federal, sempre dispostos a me ajudar a compreender o mundo do Direito, e em especial, à Jânea Oliveira que não se contenta em responder minhas perguntas, mas empreende detalhadas pesquisas para me socorrer. Ao Dr. Roberto Thomé, que me oportunizou os primeiros contatos com o trabalho realizado em gabinete e que foram decisivos para que eu pudesse compreender o dia-a-dia de gabinete e o que busca o Jurista ao lidar com um processo. À Vânia Boklis e ao Renato Luft, que têm sido extremamente compreensivos, concedendo todas compensações de horário que pedi. Ao Dr. Antônio Welter, que confiou em meu trabalho e concedeu-me licença para completar meus estudos. À Marta Roberti que elucidou muitas dúvidas.

À minha cunhada Niara Palma e ao meu genro Rodrigo Jaroseski, sempre dispostos a ouvir minhas dúvidas matemáticas e buscar uma solução.

A todos que me apoiaram quando decidi iniciar o mestrado: Rafael Bordini, Renata Vieira, Paulo Ricardo Abraão, Letícia Leite, Marcelo Cohen, Márcio Pinho, Carlos Prolo, Sílvia Moraes e Doris Fridman.

À professora Vera Strube de Lima que sempre me atende com paciência e bom humor, sem deixar de exigir nada menos do que o máximo de mim.

Aos professores Duncan Ruiz, Paulo Fernandes, Renata Vieira, Alexandre Agustini e Márcio Pinho, que sempre me atenderam com toda presteza cada vez que os interpelei sem sequer marcar um horário e sem saber se tinham disponibilidade para me auxiliar.

Aos meus colegas Lucelene Lopes, Clarissa Xavier, Mírian Bruckschen, Larissa de Freitas e Igor Wendt que também me socorrem nos momentos de dúvida.

Aos amigos do Tribunal Regional Federal da 4ª Região, sempre prontos a me ajudar, em especial ao José Ribeiro, ao Marlon Silvestre e à Juliana dos Santos.

Aos amigos do DUN2003, sempre atenciosos e prestativos.

A todos que têm fome e sede de conhecimento e, em especial, aqueles cujas descobertas lançam luzes sobre este caminho que me proponho trilhar: Aristóteles, Alan Turing, James Pustejovsky, Philipp Cimiano e meu professor, César Saldanha.

À HP que paga minha bolsa.

Àqueles que por vergonhosa falta minha, tenha me esquecido de aqui mencionar, mas que sabem que também compartilharam uma parte destes momentos preciosos que culminaram neste trabalho.

Até quando, Yahweh, pedirei socorro
e não ouvirás,
gritarei a Ti: “Violência!”,
e não salvarás?

Por que me fazes ver a iniqüidade
e contemplas a opressão?
Rapina e violência estão diante de mim,
há disputa, levantam-se contendas!

Por isso a lei se enfraquece,
e o direito não aparece nunca mais!
Sim, o ímpio cerca o justo,
por isso o direito aparece torcido!

Habacuc 1.1-4

Bem-aventurados os que têm fome e sede de justiça,
porque serão saciados.

Mateus 5.6

AGRUPAMENTO E CATEGORIZAÇÃO DE DOCUMENTOS JURÍDICOS

RESUMO

Este trabalho estuda a aplicação de técnicas de aprendizado de máquina (agrupamento e classificação) à pesquisa de jurisprudência, no âmbito do processo judicial eletrônico. Discute e implementa alternativas para o agrupamento dos documentos da jurisprudência, gerando automaticamente classes que servem ao posterior processo de categorização dos documentos anexados ao processo jurídico. O algoritmo TClus de Aggarwal, Gates e Yu é selecionado para desenvolvimento de exemplo de uso, com propostas de alteração no descarte de documentos e grupos, e passando a incluir a divisão de grupos. A proposta ainda introduz um paradigma "bag of terms and law references"em lugar do "bag of words", quando utiliza, na geração dos atributos, os tesauros do Senado Federal e da Justiça Federal para detectar termos jurídicos nos documentos e expressões regulares para detectar referências legislativas. No exemplo de uso, empregam-se documentos oriundos da jurisprudência do Tribunal Regional Federal da 4ª Região. Os resultados dos agrupamentos foram avaliados pelas medidas *Relative Hardness* e $\bar{\rho}$ e submetidos aos testes de significância de Wilcoxon e contagem de vitórias e derrotas. Os resultados da categorização foram avaliados por avaliadores humanos. A discussão e análise desses resultados abrangeu a comparação do sucesso e falha na classificação em relação à similaridade do documento com o centróide no momento da categorização, à quantidade de documentos nos grupos, à quantidade e tipo de atributos nos centróides e à coesão dos grupos. Discute-se, ainda, a geração dos atributos e suas implicações nos resultados da classificação. Contribuições deste estudo: confirmação da possibilidade de uso do aprendizado de máquina na pesquisa jurisprudencial, evolução do algoritmo TClus ao eliminar os descartes de documentos e grupos e ao implementar a divisão de grupos, proposta de novo paradigma “bag of terms and law references”, através de prototipação do processo proposto com exemplo de uso e avaliações automáticas na fase de clustering, e por especialista humano na fase de categorização.

Palavras chave: categorização, agrupamento, *hard clustering*, *bag of terms and law references*, *Relative Hardness Measure*, $\bar{\rho}$-$Measure$, Wilcoxon *signed-ranks test*, teste de contagem de vitórias e derrotas, direito, jurisprudência.

CLUSTERING AND CATEGORIZATION OF LEGAL DOCUMENTS

ABSTRACT

In this work we study the use of machine learning (clustering and classification) in judicial decisions search under electronic legal proceedings. We discuss and develop alternatives for precedent clustering, automatically generating classes to use to categorize when a user attaches new documents to its electronic legal proceeding. A changed version of the algorithm TClus, authored by Aggarwal, Gates and Yu was selected to be the use example, we propose removing its document and cluster discarding features and adding a cluster division feature. We introduce here a new paradigm "bag of terms and law references" instead of "bag of words" by generating attributes using two thesauri from the Brazilian Federal Senate and the Brazilian Federal Justice to detect legal terms a regular expressions to detect law references. In our use example, we build a corpus with precedents of the 4^th Region's Federal Court. The clustering results were evaluated with the Relative Hardness Measure and the $\bar{\rho}$-Measure which were then tested with Wilcoxon's Signed-ranks Test and the Count of Wins and Losses Test to determine its significance. The categorization results were evaluated by human specialists. The analysis and discussion of these results covered comparations of true/false positives against document similarity with the centroid, quantity of documents in the clusters, quantity and type of the attributes in the centroids e cluster cohesion. We also discuss attribute generation and its implications in the classification results. Contributions in this work: we confirmed that it is possible to use machine learning techniques in judicial decisions search, we developed an evolution of the TClus algorithm by removing its document and group discarding features and creating a group division feature, we proposed a new paradigm called "bag of terms and law references" evaluated by a prototype of the proposed process in a use case and automatic evaluation in the clustering phase and a human specialist evaluation in the categorization phase.

Keywords: categorization, clustering, hard clustering, bag of terms and law references, Relative Hardness Measure, $\bar{\rho}$-Measure, Wilcoxon signed-ranks test, Count of Wins and Losses Test, law, judicial decisions.

LISTA DE FIGURAS

Figura 2.1 Delimitação do espaço de hipóteses 24
Figura 2.2 Delimitação do espaço de hipóteses - Ampliação (esquerda) e Redução (direita) da especificidade 24
Figura 2.3 Delimitação do espaço de hipóteses - Falha na determinação do resultado 25
Figura 2.4 Árvore de decisão para solucionar o exemplo da Tabela 2.2.2 27
Figura 2.5 Rede bayesiana 28
Figura 2.6 Algoritmo KNN. K=1 rotula como triângulo, K=3 rotula como quadrado, K=6 rotula como círculo 30
Figura 2.7 SVM: instâncias linearmente separáveis à esquerda e instâncias linearmente não separáveis à direita. Fontes: Cortes e Vapnik [CV95], Mangarasian e Musicant [MM01] 30
Figura 2.8 Mapeamento de dados não linearmente separáveis. Fontes: http://www.maxdama.com/2008/07/support-vector-machines-outline.html e http://www.imtech.res.in/raghava/rbpred/algorithm.html 31
Figura 2.9 *Simple Linkage* (a), considera a máxima similaridade, ou seja, os termos mais próximos. *Complete Linkage* (b), considera a mínima similaridade, ou seja, os termos mais distantes. *Average Linkage* (c), considera a média das similaridades entre todos os termos de cada *cluster* 33
Figura 2.10 O algoritmo K-Means pressupõe que as instâncias sejam resultado da superposição de distribuições gaussianas que compartilham mesma variância 35
Figura 2.11 O algoritmo EM pressupõe que as instâncias sejam resultado da superposição de distribuições gaussianas; as variâncias poderão ser distintas 35
Figura 3.1 Comportamento do Índice Normalizado Gini 65
Figura 3.2 Classificação de Documentos: Determinação da Dominância do Grupo 66
Figura 3.3 Seleção de atributos para Determinação de Dominância quando os Documentos estão em Região *Intercluster* 66
Figura 4.1 Jurisprudência do TRF/4ª 72
Figura 4.2 Processo de Agrupamento e Classificação 75
Figura 4.3 Arquitetura detalhada do agrupamento e da categorização 76
Figura 4.4 Exemplo de Estrutura de grafo presente no Tesauro da Justiça Federal 80
Figura 4.5 Programa para Mesclagem de Tesauros 81
Figura 4.6 Estrutura da Jurisprudência do TRF/4ª 82
Figura 4.7 Arquitetura do Pré-Processamento 83
Figura 4.8 Exemplo de vetor de atributos 85

Figura 4.9 Sucessivas iterações podem atrair documentos para o *cluster* recém-criado 89
Figura 5.1 Ferramenta de Validação da Categorização 99
Figura 5.2 Gráfico da Validação por especialista 100
Figura 5.3 Reconhecimento exclusivo dos termos mais específicos 105
Figura 5.4 Similaridade entre o documento e a classe - simcateg 113
Figura 5.5 Relação entre os indicadores **qtdoc**, **coesao**, **qtterm** e **qtrefleg** e a avaliação humana 115
Figura 5.6 Relação entre os indicadores **qtattseed**, **qtmerge** e **qtattdoc** e a avaliação humana 116
Figura A.1 Programa para seleção/descarte de documentos 134

LISTA DE TABELAS

Tabela 2.1 Dados de treino para determinação das situações em que se necessita colocar mais caixas em um banco 24
Tabela 2.2 Novos dados de treino para determinação das situações em que se necessita colocar mais caixas em um banco 26
Tabela 2.3 Probabilidades de rede bayesiana da Figura 2.5 28
Tabela 2.4 Cálculo de Coesão e Separação de agrupamentos 41
Tabela 3.1 Trabalhos Relacionados: Quadro Comparativo 70
Tabela 4.1 Excerto de Estrutura do Tesauro do Senado Federal 79
Tabela 4.2 Estrutura do Tesauro da Justiça Federal 80
Tabela 4.3 Sintaxe das Indicações de Equivalência no TJF 80
Tabela 4.4 Normalização de referências legislativas 84
Tabela 5.1 Variações empregadas em cada execução do agrupamento 93
Tabela 5.2 Medidas internas aferidas em cada agrupamento 96
Tabela 5.3 *Sign Test* para *Relative Hardness* 98
Tabela 5.4 *Sign Test* para $\bar{\rho}$-$Measure$ 98
Tabela 5.5 Atributos do Grupo “Crime” 102
Tabela 5.6 Temas do Grupo “Crime” 103
Tabela 5.7 Atributos do Grupo “estação de rádio” 104
Tabela 5.8 Atributos do Grupo “dano && indenização” 104
Tabela 5.9 Atributos do Grupo “crédito tributário && multa” 105
Tabela 5.10 Atributos do Grupo “dano && indenização” 106
Tabela 5.11 Novas Medidas internas aferidas em cada agrupamento 109
Tabela 5.12 *Sign Test* para *Relative Hardness* entre os algoritmos 3 e 6 110
Tabela 5.13 *Ranks* de *Relative Hardness* para o cálculo do *Wilcoxon Sign Test* entre os algoritmos 3 e 6 110
Tabela 5.14 *Sign Test* para $\bar{\rho}$-$Measure$ entre os algoritmos 3 e 6 111
Tabela 5.15 *Sign Test* para *Relative Hardness* 111
Tabela 5.16 *Sign Test* para $\bar{\rho}$-$Measure$ 111
Tabela 5.17 *Ranks* de $\bar{\rho}$-$Measure$ para o cálculo do *Wilcoxon Sign Test* 112
Tabela 5.18 Quantidade máxima de categorias usadas nos trabalhos relacionados 121
Tabela B.1 Quantidade de atributos não nulos nas Classes/Grupos Iniciais 135
Tabela B.2 Quantidade de Documentos nas Classes/Grupos Iniciais 135
Tabela C.1 Quantidade de atributos não nulos nas Classes/Grupos Finais 136
Tabela C.2 Quantidade de Documentos nas Classes/Grupos Finais 136
Tabela D.1 Atributos descartados 137

LISTA DE ALGORITMOS

Algoritmo 1 **Divide** 88
Algoritmo 2 **Assign** 89

LISTA DE EQUAÇÕES

2.1 Expressões Conjuntivas 26
2.2 Expressão Gerada por Ávore de Decisão 27
2.3 Teorema de Bayes - Fórmula Geral 29
2.4 TF - Term Frequency 38
2.5 IDF - Inverse Document Frequency 38
2.6 TF-IDF - Term Frequency-Inverse Document Frequency 38
2.7 Distância Euclidiana 38
2.8 Distância de Cosseno 39
2.9 Coeficiente de Silhueta Médio de um Grupo 41
2.10 Coeficiente de Silhueta Médio do Agrupamento 42
2.11 Índices Dunn - Fórmula Geral 42
2.12 Índice Davies-Bouldin 42
2.14 Medida Rho 43
2.15 Medida RH 43
2.16 Teste de Contagem de Vitórias e Derrotas 44
2.17 Teste de Wilcoxon 44
3.1 Classificador Bayesiano Parametrizado por EM Particional 51
3.2 Presença Fracional de uma Palavra em uma Classe 65
3.3 Índice Normalizado Gini 65
5.1 Teste de Sinal para RH entre os Algoritmos 3 e 6 110
5.2 Teste de Sinal para RH entre os Algoritmos 1 e 6 112

LISTA DE SIGLAS

AJG – *Assistência Judiciária Gratuita*
AMS-EM – *Alternate Model Selection EM*
BSEM – *Bayesian Structural EM*
CBC – *Cluster Based Categorization*
CDD – *Classificação Decimal de Dewey*
CF – *Constituição Federal*
CF – *Cluster Features*
CICLing – *Conference on Intelligent Text Processing and Computational Linguistics*
CID – *Classificação Internacional de Doenças*
CJF – *Conselho da Justiça Federal*
CNJ – *Conselho Nacional da Justiça*
Co-EM – *Co-Training EM*
CSS – *Cascading Style Sheet*
DIB – *Data do Início do Benefício*
DJU – *Diário da Justiça da União*
DMJ – *Departamento Médico Judiciário*
EIAC – *Embargos Infringentes em Apelação Cível*
ELAG – *Elimination of Lexical Ambiguities by Grammars*
EM – *Expectation Maximization*
FP – *Falso(s) Positivo(s)*
FREM – *Fast and Robust Expectation Maximization*
HTML – *Hypertext Markup Language*
IGP-DI – *Índice Geral de Preços Disponibilidade Interna*
INSS – *Instituto Nacional de Seguridade Social*
KBC – *Keyword Based Clustering*
KNN - *K Nearest Neighbour*
LSA – *Latent Semantic Analysis*
MM – *Meritíssimo*
M-EM – *EM with Multiple Mixture Components per Class*
MNINST – *Mixed NIST Database of Handwritten Digits*
MS-EM – *Model Selection EM*
NIST – *National Institue of Standards and Technology*
ODP – *Open Directory Project*
PDF – *Portable Document Format*
PLSA – *Probabilistic LSA*
POSIX – *Portable Operating System Interface*

RGPS – *Regime Geral de Previdência Social*
RH – *Relative Hardness*
SEM – *Structural EM*
SemEval – *Semantic Evaluations*
SSRjMC – *Erro de digitação no documento original, vide SSR/MC*
SSR/MC – *Secretaria de Serviços de Rádiodifusão do Ministério das Comunicações*
STF – *Supremo Tribunal Federal*
STJ – *Superior Tribunal de Justiça*
STM – *Superior Tribunal Militar*
SVC – *Support Vector Clustering*
SVM – *Support Vector Machine*
TF-IDF – *Term Frequency - Inverse Document Frequency*
TJF – *Tesauro da Justiça Federal*
TJRS – *Tribunal de Justiça do Estado do Rio Grande do Sul*
TOD – *Threshold Order dependent*
TRF – *Tribunal Regional Federal*
TRF4 ou TRF/4ª – *Tribunal Regional Federal da 4ª Região*
TRT4 ou TRT/4ª – *Tribunal Regional do Trabalho da 4ª Região*
TSE – *Tribunal Superior Eleitoral*
TST – *Tribunal Superior do Trabalho*
TSVM – *Transductive SVM*
UPGMA – *Unweighted Pair Group Method with Arithmetic Mean*
URL – *Universal Resource Locator*
VCB – *Vocabulário Controlado Básico*
VCJ – *Vocabulário Controlado da Justiça*
VP – *Verdadeiro(s) Positivo(s)*
WebKB – *Web Knowledge Base*
WP – *Word Presence*
WSI – *Word Sense Induction*

SUMÁRIO

1. Introdução 21

2. Fundamentação Teórica 23
2.1 Considerações Iniciais 23
2.2 Aprendizado supervisionado – Categorização 23
2.2.1 Busca em Espaço de Estados 23
2.2.2 Árvores de Decisão 26
2.2.3 Redes Bayesianas 28
2.2.4 *K-Nearest Neighbors* - KNN 29
2.2.5 SVM - *Support Vector Machine* 30
2.2.6 Metodologia 32
2.3 Aprendizado Não-Supervisionado – *Clustering* 32
2.3.1 O Algoritmo *K-Means* 35
2.3.2 Algoritmo *Expectation-Maximization* (EM) 36
2.3.3 Agrupamento Semi-Supervisionado 36
2.4 Pré-Processamento 37
2.5 Funções de Proximidade 38
2.6 Métodos de Validação de Classificação 39
2.7 Métodos de Validação de Grupos 40
2.7.1 Métodos de Validação de Grupos Supervisionados 40
2.7.2 Métodos de Validação de Grupos Não Supervisionados 41
2.7.2.1 Medidas de Coesão e Separação de Grupos 41
2.7.2.2 Coeficiente de Silhueta 41
2.7.2.3 A família de Índices Dunn 42
2.7.2.4 Índice Davies-Bouldin 42
2.7.2.5 Medida Λ 42
2.7.2.6 Medida $\bar{\rho}$ 43
2.7.2.7 Medida *Relative Hardness* 43
2.8 Métodos de Comparação de Algoritmos de Aprendizado de Máquina 44
2.9 Considerações Finais 45

3. Trabalhos Relacionados 46
3.1 Considerações Iniciais 46
3.2 Trabalhos Baseados em Classificadores Bayesianos 46
3.2.1 Gerando Redes Bayesianas usando o Algoritmo EM 46

3.2.2 Classificação de Texto num Modelo de Mistura Hierárquico para Pequenos Conjuntos de Treino . . . 48
3.2.3 Classificação de Textos Semi-Supervisionada Usando EM Particional 50
3.2.4 Analisando a Efetividade e Aplicabilidade do *Co-Training* . . . 52
3.3 Trabalhos Baseados em Classificadores SVM ou Derivados do SVM . . . 55
3.3.1 Combinando *Clustering* e *Co-Training* para Melhorar a Classificação de Textos Usando Dados Não Rotulados . . . 55
3.3.2 CBC: Classificação de Texto Baseada em *Clustering* Requerendo Mínimos Dados Rotulados . . . 57
3.3.3 *Support Cluster Machine* . . . 59
3.3.4 Classificação SVM Hierárquica Baseada em *Support Vector Clustering* e sua Aplicação na Categorização de Documentos . . . 60
3.3.5 Mineração de Textos de Decisões da Suprema Corte Administrativa Austríaca . . . 62
3.3.6 Aprendizagem Ativa Usando Pré-*Clustering* . . . 63
3.4 Usando Supervisão Parcial para Categorização de Textos . . . 64
3.5 Considerações Finais . . . 68

4. Classificação de Textos Jurídicos usando Classes Geradas por Agrupamento Parcialmente Supervisionado . . . 71
4.1 Considerações Iniciais . . . 71
4.2 Aporte Teórico Utilizado . . . 73
4.3 Visão Geral da Solução Adotada . . . 74
4.4 Detalhamento da Solução Adotada . . . 75
4.4.1 Composição do *Corpus* . . . 77
4.4.2 Pré-Processamento de Documentos . . . 78
4.4.2.1 Estruturas Terminológicas . . . 78
4.4.2.2 Base Lexical . . . 81
4.4.2.3 Arquitetura do Pré-Processamento . . . 82
4.4.3 *Parsing*, Lematização, Reconhecimento de Termos e Descarte de Atributos . . . 83
4.4.3.1 *Parser* . . . 84
4.4.3.2 Lematizador . . . 84
4.4.3.3 Reconhecimento de Termos . . . 85
4.4.3.4 Descarte de atributos . . . 85
4.5 Processo de Agrupamento e Classificação . . . 86
4.5.1 Agrupamento . . . 86
4.5.1.1 Algoritmo de Divisão . . . 86

4.5.1.2 Algoritmo de Divisão Implícita 87
4.5.2 Categorização 89
4.6 Considerações Finais 90

5. Avaliação 92
5.1 Considerações Iniciais 92
5.2 Parâmetros Adotados na Validação 92
5.3 Avaliações Realizadas 95
5.3.1 Análise dos Agrupamentos 95
5.3.2 Análise da Classificação 98
5.4 Informação Não Extraída dos Documentos 101
5.4.1 Falsos Positivos com Alta Similaridade 101
5.5 Verdadeiros Positivos com Baixa Similaridade 104
5.6 Possíveis Soluções 104
5.6.1 Problema dos Centróides com Poucos Atributos Não Nulos 104
5.6.2 Problema dos Atributos com Semântica Muito Genérica 105
5.6.2.1 Descarte de Nodos Não Terminais 105
5.6.2.2 Atribuição de Pesos aos Termos 106
5.6.2.3 Agrupamento Hierárquico 106
5.6.3 Atualização dos Tesauros 107
5.6.4 Agrupamento Semi-supervisionado por Referências Legislativas . . . 107
5.7 Atribuição de Pesos Semânticos aos Termos e referências Legislativas . . . 108
5.8 Nova Análise dos Agrupamentos 109
5.9 Nova Análise da Classificação 112
5.10 Impressões dos Especialistas Humanos 116
5.11 Considerações Finais 118

Referências Bibliográficas 126

Apêndice A. Programa de Seleção de Documentos 134

Apêndice B. Grupos Iniciais 135

Apêndice C. Grupos Finais 136

Apêndice D. Atributos Descartados Via Índice Normalizado Gini 137

Apêndice E. Sobre o Especialista Humano 1 138

Apêndice F. Sobre o Especialista Humano 2 139

Anexo A. Teor do documento Nº 50 140

Anexo B. Teor do documento Nº 17 144

1. Introdução

Por determinação do Conselho Nacional de Justiça, até o final do ano de 2010, todo o Poder Judiciário brasileiro teve que implantar o processo eletrônico, finalizando o trâmite de documentos em papel. Atividades de rotina em gabinetes dos magistrados incluem a pesquisa por decisões proferidas em casos julgados anteriormente, a jurisprudência. Quando o juiz encontra um caso semelhante ao que está estudando, tem a oportunidade de, concordando com os argumentos apresentados, aproveitar a fundamentação exposta, reduzindo drasticamente o tempo gasto elaborando a fundamentação de sua decisão. Para agilizar este trabalho, sistemas usando recursos de Processamento da Linguagem Natural e Mineração de Dados para classificação e recuperação de documentos podem representar uma melhoria nos procedimentos de pesquisa.

Este estudo propõe o uso de processos de agrupamento e categorização de textos jurídicos, descritos no Capítulo 4, Seções 4.5.1 e 4.5.2. O processo proposto é constituído de uma fase de agrupamento dos documentos que compõem a jurisprudência, gerando um conjunto de classes correspondentes aos grupos encontrados e outra fase que, quando os litigantes enviarem peças processuais em forma digital através de *upload* no sistema processual eletrônico, categoriza os documentos enviados e retorna aos usuários os documentos integrantes do grupo que gerou a respectiva classe.

Para tanto, nossa revisão dos fundamentos de aprendizado de máquina, constante do Capítulo 2, abrange algoritmos clássicos de agrupamento, abordados na Seção 2.3 e de categorização, Seção 2.2. O estudo de trabalhos correlatos, investigando a evolução do emprego de categorização precedida por agrupamento, encontra-se no Capítulo 3. Ressaltamos que somente um destes trabalhos, apresentado na Seção 3.3.5 utilizou documentos do domínio jurídico do conhecimento.

O pré-processamento dos documentos usa uma mescla de tesauros jurídicos mantidos pelo Senado Federal e pelo Conselho da Justiça Federal, descritos na Seção 4.4.2.1, para extrair termos dos documentos e compor vetores de atributos, abandonando o paradigma *bag of words* em prol do *bag of terms and law references*. Para aplicar o processo proposto em um exemplo de uso, foram construídos um *corpus*, descrito no Capítulo 4, Seção 4.4.1, uma base lexical, descrita no Capítulo 4, Seção 4.4.2.2, um *parser* e um *tagger*, descritos no Capítulo 4, Seção 4.4.3. Resultados são descritos no Capítulo 5.

A avaliação dos resultados, exposta no Capítulo 5, compreendeu o cálculo de medidas internas dos agrupamentos realizados, apresentados na Seção 5.3.1, para a fase de teste, e na Seção 5.8, para a fase de operação; e validação da categorização, na Seção 5.3.2, para a fase de teste, e Seção 5.9, para a fase de operação, através de especialista humano. Da análise destas avaliações emergem, então, nossas conclusões, expostas no Capítulo 5.11.

Nossas contribuições, não se limitam a confirmar a possibilidade de uso de técnicas

de aprendizado de máquina para realizar pesquisa de jurisprudência. Incluem, também, proposta de evolução deste algoritmo, avaliada mediante prototipação do algoritmo com variações onde se eliminaram os descartes de documentos e grupos e implementou-se a divisão de grupos. Além disto, propusemos novo paradigma “bag of terms and law references”, a ser melhor explorado em trabalhos futuros. Para tanto, além de construir parser reconhecedor de referências legislativas, mesclamos 3 dicionários, 2 lematizadores e 2 tesauros jurídicos, utilizados no pré-processamento do corpus jurídico, que construimos com a jurisprudência do Tribunal Regional Federal da 4ª Região.

2. Fundamentação Teórica

2.1 Considerações Iniciais

Desde os primórdios da informática, o computador vem sendo empregado na solução de problemas de complexidade crescente. Alguns destes problemas, são solucionáveis através de um método conhecido, determinado previamente. São exemplos disto, sistemas administrativos, cálculos, etc.

Outros problemas não têm sua solução pré-determinada. Conhece-se, apenas, uma certa quantidade de informações relacionadas ao problema. Neste caso, segundo Mitchell [Mit97] podemos empregar métodos de aprendizado de máquina. Em função das informações que dispomos a respeito do problema, temos duas possibilidades:

1. Entre as informações disponíveis, encontramos soluções para situações específicas. Neste caso, podem ser empregados métodos que buscam solucionar problemas novos, guiando-se pelas soluções já conhecidas. Este é o aprendizado supervisionado, também chamado de categorização ou, ainda, classificação. Faremos um breve estudo destes métodos na Seção 2.2;

2. Entre as informações disponíveis, não contamos com soluções prévias. Neste caso, é empregada a abordagem de aprendizado não supervisionado, ou *clustering*, ou, em português, agrupamento, que será estudado na Seção 2.3.

2.2 Aprendizado supervisionado – Categorização

Segundo Tan, Steinbach e Kumar [TSK09], este tipo de aprendizado consiste em analisar um conjunto de situações, denominadas “instâncias” ou exemplos, e suas características, denominadas “atributos”. Entre os atributos, aquele que apresenta a solução previamente conhecida é denominado “atributo alvo” ou “rótulo de classe”. Os valores do atributo alvo constituem as soluções previamente conhecidas e, portanto, compõem o conjunto de possíveis soluções.

2.2.1 Busca em Espaço de Estados

Quando estão presentes as soluções para casos específicos, formulam-se as hipóteses possíveis e, a partir delas, cria-se um solucionador, uma função booleana que, dadas novas informações, responde com o valor da característica faltante.

Suponha, por exemplo, que um banco deseja descobrir em que situações se precisa colocar caixas extras para atender o público num determinado dia. O conjunto de dados

Tabela 2.1 – Dados de treino para determinação das situações em que se necessita colocar mais caixas em um banco

Início de Mês	Feriadão	Verão	Dia da Semana	Mais Caixas
S	S	N	2ª	S
N	N	N	2ª	N
S	N	N	4ª	S
N	N	N	2ª	N
N	N	S	3ª	N

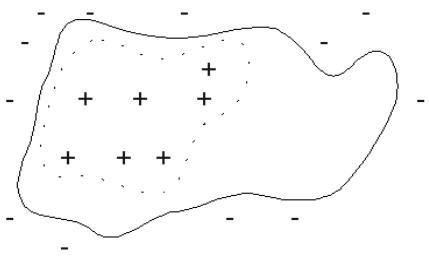


Figura 2.1 – Delimitação do espaço de hipóteses

disponíveis poderia ser o da Tabela 2.1, onde “Início de mês” é um atributo booleano que, quando verdadeiro, indica que é um dos primeiros 5 dias úteis do mês; “Feriadão” é verdadeiro quando é retorno ou véspera de feriadão e falso para as demais situações; “Verão”, um booleano que indica se é ou não verão; “Dia da semana”, os dias de expediente bancário e “Mais Caixas” é o atributo alvo, booleano.

A partir destes dados podem ser formuladas hipóteses para a função que determina situações de necessidade de se colocar caixas extras para atendimento. Estas hipóteses podem ser mais genéricas ou mais específicas. A hipótese mais específica para estes dados seria “Início de mês” = “S” e “Verão” = “N”, já a hipótese mais genérica seria “Início de mês” = “S”. Desta maneira, a busca da função solução pode ser definida como uma

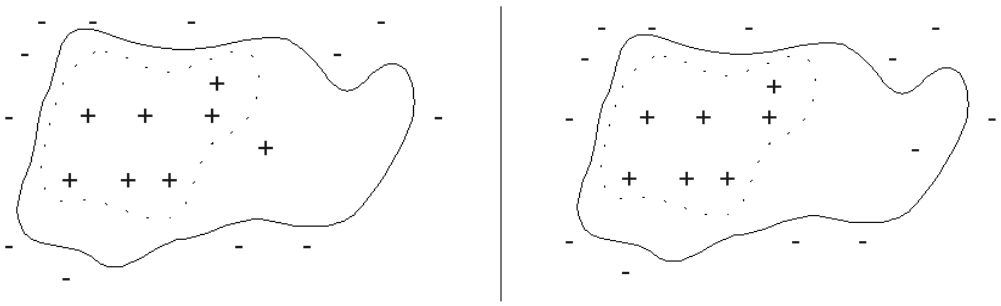


Figura 2.2 – Delimitação do espaço de hipóteses - Ampliação (esquerda) e Redução (direita) da especificidade

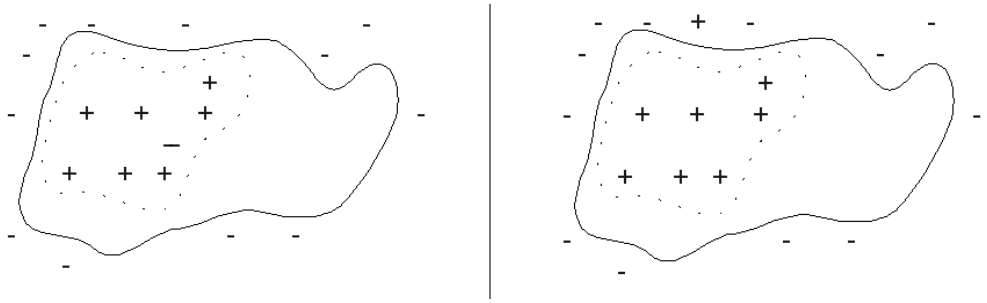


Figura 2.3 – Delimitação do espaço de hipóteses - Falha na determinação do resultado

busca, no espaço de hipóteses, pela hipótese mais adequada para solucionar o problema de encontrar o valor do atributo alvo [Mit97].

Duas abordagens clássicas propostas para solucionar este problema são:

1. O algoritmo *Find-S*, que busca todas as hipóteses que incluam todos os resultados positivos e excluam todos os resultados negativos, selecionando, dentre estas hipóteses, a **mais específica**;

2. O algoritmo *Candidate-Elimination*, que busca todas as hipóteses que incluam todos os resultados positivos e excluam todos os resultados negativos, selecionando, dentre estas hipóteses, a **mais genérica**.

A Figura 2.1 permite observar a diferença entre as soluções adotadas pelos dois algoritmos: a área delimitada pela linha pontilhada representa a hipótese mais específica do algoritmo *Find-S* e a área delimitada pela linha contínua, a hipótese mais genérica do algoritmo *Candidate-Elimination*. A área entre estas duas linhas compreende possíveis hipóteses intermediárias entre a mais específica e a mais genérica.

Se, conforme a Figura 2.2 (esquerda), uma nova instância, com atributos que incidam nesta área intermediária, apresentar um resultado positivo, dever-se-á descartar a hipótese mais específica do algoritmo *Find-S* e adotar, em seu lugar, a hipótese mais específica que englobe esta nova instância. Da mesma maneira, conforme a Figura 2.2 (direita), se uma nova instância, com atributos que incidam nesta área intermediária, apresentar um resultado negativo, dever-se-á descartar a hipótese mais genérica do algoritmo *Candidate-Elimination* e adotar, em seu lugar, a hipótese mais genérica que exclua esta instância. Suponha, agora, a ocorrência de uma nova instância com resultado negativo, dentro da área delimitada por uma hipótese mais específica, gerada pelo *Find-S*. Isto leva a uma situação na qual não será mais possível obter uma função que determine o resultado sem erros. Teremos a mesma situação, se uma instância com resultado positivo ocorrer além da área delimitada por uma hipótese mais genérica, gerada pelo algoritmo *Candidate-Elimination*. Vide Figura 2.3. Em outras palavras, é vazio o conjunto de hipóteses consistentes com a solução [Mit97].

Tabela 2.2 – Novos dados de treino para determinação das situações em que se necessita colocar mais caixas em um banco

Início de Mês	Feriado	Verão	Dia da Semana	Mais Caixas
N	V	N	5ª	S
S	V	N	2ª	S
N	N	S	2ª	S
N	N	N	2ª	N
S	N	N	4ª	S
N	R	S	5ª	N
N	R	S	3ª	S
N	N	N	2ª	N
N	N	S	2ª	S
S	N	S	3ª	S
N	N	S	3ª	N

2.2.2 Árvores de Decisão

Os algoritmos *Find-S* e *Candidate-Elimination* trabalham com a idéia da busca no espaço de hipóteses. No entanto, a quantidade de hipóteses mesmo para uma pequena quantidade de atributos e uma pequena quantidade de possíveis valores para estes atributos, facilmente atinge uma enorme gama de hipóteses. Suponha o exemplo dos caixas do banco. Considerando que em uma dada hipótese, cada atributo pode ter um de seus possíveis valores, caso específico, ou ter seu valor indeterminado, caso genérico. Desta maneira, voltando ao exemplo dado na Seção 2.2.1 (Tabela 2.1), teremos 3 * 3 * 3 * 6 = 162 possibilidades, 163 se contarmos a hipótese vazia. Se considerarmos o atributo “feriado” como tendo 3 valores possíveis (“véspera de feriado”, “retorno de feriado” e “nenhum feriado”), a quantidade de hipóteses sobe para 217. Se dobrarmos a quantidade de atributos, teremos 3 * 3 * 3 * 6 * 3 * 3 * 3 * 6 + 1 = 26.245 hipóteses! E a maioria dos problemas reais se representa com muito mais que 8 atributos. Assim, buscar no espaço de hipóteses facilmente se torna inviável [Mit97].

Além disto, estes algoritmos geram soluções na forma de expressões conjuntivas, ou seja:

$$\bigwedge_{i=1}^{n} A_i \; Rel_i \; V_i \tag{2.1}$$

onde A é um atributo, V é um valor e Rel é um relação, como ‘$=$’, ‘$<$’, ’$\leq$’ ou ‘$\geq$’.

Assim, supondo que, por exemplo, o atributo feriado passasse a ter 3 possíveis valores, *vespera de feriado*, *retorno de feriado* e *nenhum feriado*. Desta maneira, poderíamos ter instâncias positivas em que o valor deste atributo seria *vespera de feriado* e em outras instâncias positivas, o valor seria *retorno de feriado*. Neste caso, não encontraríamos uma hipótese consistente, pois teríamos que prever uma disjunção (*feriado=retorno de fe-*

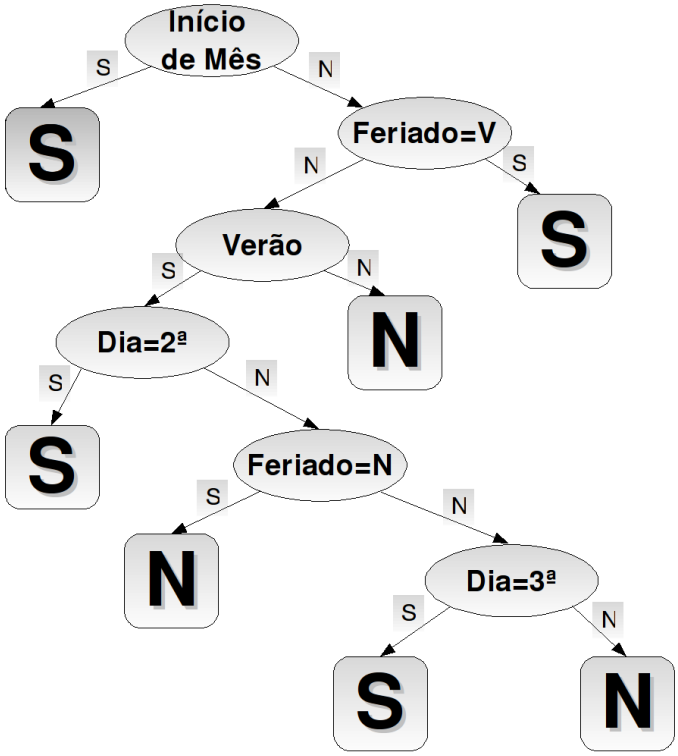


Figura 2.4 – Árvore de decisão para solucionar o exemplo da Tabela 2.2.2

riado **OU** *feriado=vespera de feriado*) e tal não é possível de se obter com estes algoritmos [Mit97].

As árvores de decisão são uma forma de se obter uma solução que preveja a disjunção. Suponha que o novo conjunto de treino seja o da Tabela 2.2.2. Uma possível árvore de decisão para classificar as instâncias poderia ser a da Figura 2.4. Os nodos folha com o valor “S” indicam as situações em que se precisa aumentar a quantidade de caixas no atendimento, os nodos folha com o valor “N” indicam que não há necessidade disto. Desta árvore geramos a expressão:

$$\begin{aligned} I \quad = \quad & s \vee (I = n \wedge F = v) \vee (I = n \wedge F \neq v \wedge V = s \wedge D = 2) \\ & \vee (I = n \wedge F \neq v \wedge V = s \wedge D \neq 2 \wedge F \neq N \wedge D = 3) \end{aligned}$$

que, ainda, pode ser simplificado para:

$$I = s \vee (I = n \wedge F = v) \vee (I = n \wedge F \neq v \wedge V = s \wedge D = 2) \vee (I = n \wedge F = r \wedge V = s \wedge D = 3) \quad (2.2)$$

onde

1. I é o início do mês;
2. F é o feriado;
3. V é o verão;

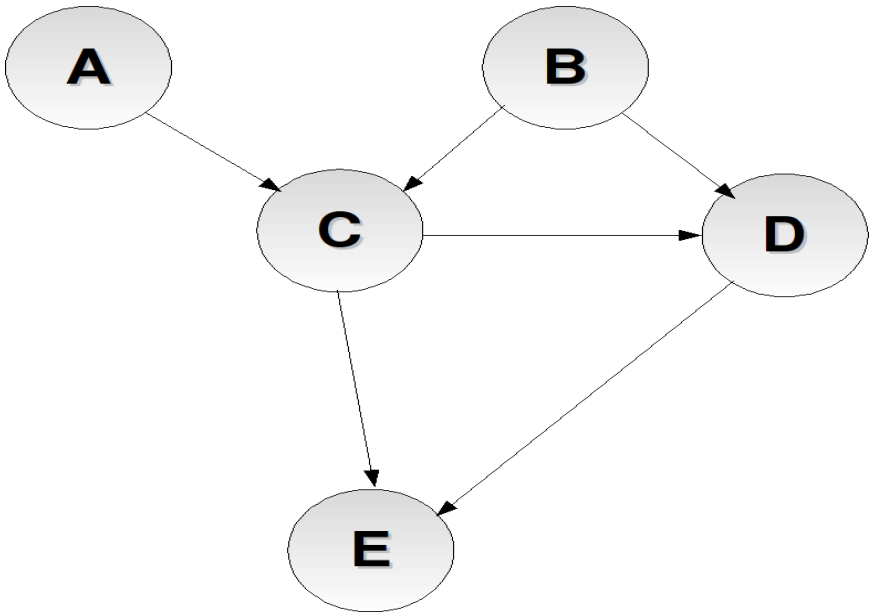


Figura 2.5 – Rede bayesiana

Tabela 2.3 – Probabilidades de rede bayesiana da Figura 2.5

	A
V	0,2
F	0,8

	B
V	0,77
F	0,23

		C	
A	B	V	F
V	V	0,08	0,17
V	F	0,14	0,11
F	V	0,1	0,25
F	F	0	0,15

		D	
B	C	V	F
V	V	0,02	0,23
V	F	0,01	0,24
F	V	0,25	0
F	F	0,19	0,06

		E	
C	D	V	F
V	V	0,18	0,07
V	F	0,11	0,12
F	V	0,25	0,02
F	F	0,2	0,05

4. D é o dia da semana;
5. s,n são os valores *booleanos* “sim” e “não”;
6. v,r são os valores “véspera de feriado” e “retorno de feriado”;
7. 2,3,4,5,6 são os dias da semana, de segunda a sexta-feira.

2.2.3 Redes Bayesianas

Uma outra abordagem para o aprendizado supervisionado, é o uso da teoria das probabilidades. Conforme Luger [Lug04] uma rede bayesiana [Pea85] é representada por um grafo acíclico dirigido, conforme exemplo da Figura 2.5, onde as relações indicam um certo grau de causalidade. Cada nodo representa um evento e tem uma probabilidade conhecida de ocorrência. Os nodos pais têm probabilidades de ocorrência independentes e os nodos filhos têm probabilidade de ocorrência influenciada pela ocorrência dos nodos pais. Conforme o exemplo da Tabela 2.3, o nodo pai A tem 20% de probabilidade de ocorrer, enquanto

o nodo B tem 77% de probabilidade de ocorrer. Já o nodo C tem 14% de probabilidade de ocorrer se somente A tiver ocorrido, 10% se somente B tiver ocorrido, 8% se ambos tiverem ocorrido e probabilidade 0%, ou seja, não ocorre, se nenhum deles tiver ocorrido. Note que a probabilidade de A e B ocorrerem simultaneamente é dada por

$$P(A \wedge B) = P(A) \times P(B)$$

e a probabilidade de ocorrência de "C" é dada por

$$P(C|A \wedge B) = \frac{P(C \wedge A \wedge B)}{P(A \wedge B)}$$

e a probabilidade de $P(A \wedge B|C)$ é dada por

$$P(A \wedge B|C) = \frac{P(C \wedge A \wedge B)}{P(C)}$$

isolando $P(C \wedge A \wedge B)$, obtemos

$$P(C \wedge A \wedge B) = P(A \wedge B|C) \times P(C)$$

substituindo este resultado na relação $P(C|A \wedge B)$, obtemos o teorema de Bayes:

$$P(C|A \wedge B) = \frac{P(A \wedge B|C) \times P(C)}{P(A \wedge B)}$$

A forma geral do teorema de Bayes é dada por:

$$P(H_i|E) = \frac{P(E|H_i) \times P(H_i)}{\sum_{k=1}^{n} P(E|H_k) \times P(H_k)} \tag{2.3}$$

onde

1. E é uma determinada evidência (nodo pai, ou causador);
2. H_i é uma dada hipótese.

2.2.4 *K-Nearest Neighbors* - KNN

O algoritmo KNN [CH67] adota o modelo de vetor de espaço n-dimensional para rotular novas instâncias baseando-se nos rótulos dos vizinhos mais próximos. Definido um valor para k, busca-se os k vizinhos mais próximos e atribui-se à nova instância o rótulo mais freqüente dentre eles [TSK09]. A Figura 2.6 apresenta exemplos de atribuição de rótulos de uma mesma instância

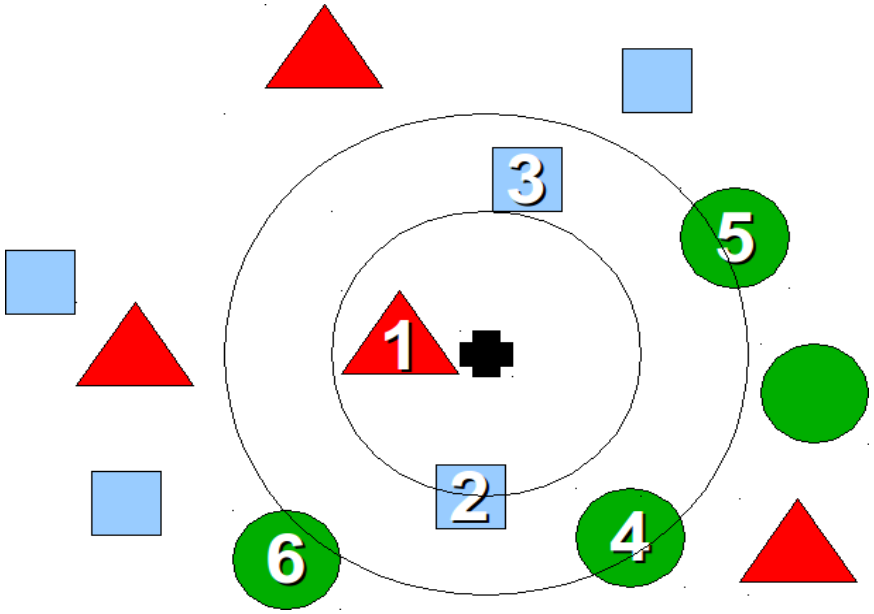


Figura 2.6 – Algoritmo KNN. K=1 rotula como triângulo, K=3 rotula como quadrado, K=6 rotula como círculo

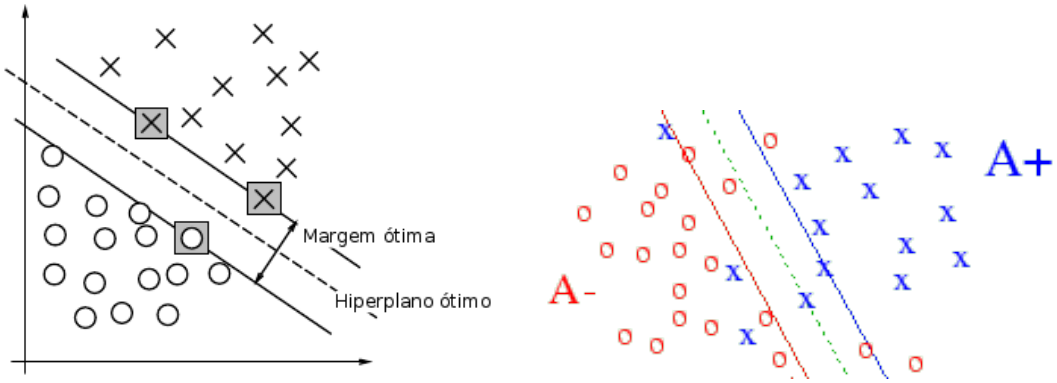


Figura 2.7 – SVM: instâncias linearmente separáveis à esquerda e instâncias linearmente não separáveis à direita. Fontes: Cortes e Vapnik [CV95], Mangarasian e Musicant [MM01]

2.2.5 SVM - *Support Vector Machine*

O SVM [BGV92] é um algoritmo que busca descobrir um hiperplano que (1) separe as classes alvo de forma que todas as instâncias de uma classe estejam de um lado do hiperplano e todas as instâncias da outra classe estejam do outro lado do hiperplano e (2) havendo múltiplos hiperplanos possíveis de satisfazer (1), selecione aquele em que haja a maior margem entre o hiperplano e os indivíduos mais próximos. A maximização desta margem reduz a chance de erros quando novas instâncias forem classificadas. Esta definição, no entanto, aplica-se a casos onde as instâncias são linearmente separáveis. O SVM pode ser reformulado para aceitar um certo limite de erros durante o treinamento e, assim, encontrar hiperplano em casos não separáveis linearmente. Este método é conhe-

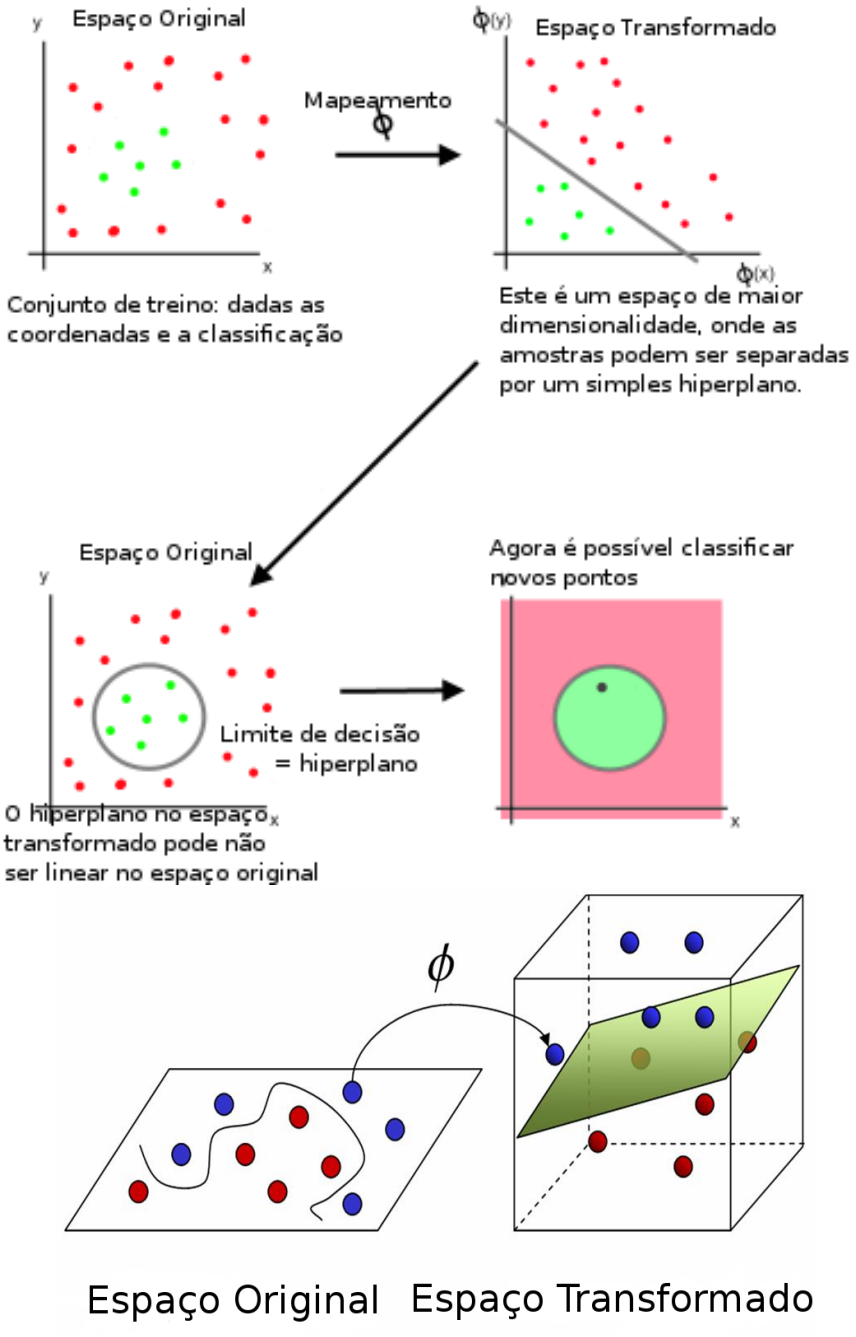


Figura 2.8 – Mapeamento de dados não linearmente separáveis. Fontes: http://www.maxdama.com/2008/07/support-vector-machines-outline.html e http://www.imtech.res.in/raghava/rbpred/algorithm.html

cido como abordagem da margem flexível e pode, também, ser aplicado a casos em que as instâncias sejam linearmente separáveis mas que a margem obtida seja muito pequena. Neste caso, é possível que exista um hiperplano com uma margem maior com erros dentro do limite de erros aceitáveis [TSK09]. A Figura 2.7 apresenta exemplos de casos de separabilidade de instâncias. Já a Figura 2.8 apresenta exemplos de dados não linearmente separáveis em que é preciso mapear o espaço original para um novo espaço onde seja possível separar as instâncias por meio de um hiperplano linear [TSK09].

2.2.6 Metodologia

De acordo com Mitchell [Mit97], no processo de categorização, temos 3 fases:

1. **Fase de treino**: na qual se analisam as instâncias conhecidas, procurando determinar formas de se chegar ao "atributo alvo", ou seja, a solução geral, aplicável quando fornecidas novas instâncias onde não se conheça previamente o valor deste atributo.

2. **Fase de teste**: na qual se avalia a qualidade do treino, seja em virtude do algoritmo, seja em face das instâncias utilizadas para o treinamento. Nesta fase, aplicamos a solução geral determinada na fase anterior a um conjunto de instâncias cujos valores dos atributos alvo são conhecidos. As soluções produzidas para estas instâncias são comparadas com as soluções que previamente dispúnhamos e nas quais confiamos que estejam corretas. Desta maneira, medimos a precisão do resultado, bem como as características específicas dos erros cometidos, tais como quantidade de "falsos positivos" e "falsos negativos";

3. **Fase de operação**: tendo determinado uma solução de qualidade aceitável, passa-se à fase de operação, quando não mais se conhecem as soluções para as situações. Avaliações serão feitas nesta fase, sendo comum descobrir, na prática, taxas de erros significativamente superiores aos encontrados na fase de teste. Isto pode ocorrer devido à solução ser demasiadamente específica para o conjunto de treino. Tal situação é denominada "overfitting";

2.3 Aprendizado Não-Supervisionado – *Clustering*

Todas as abordagens estudadas até aqui contam com o fato de se conhecer previamente, no conjunto de treino, para cada instância, o valor do atributo alvo, atuando, assim, como "professor" da máquina, ou seja, o aprendizado é supervisionado. Há casos, porém, em que não se conhece, com antecedência, o valor do atributo alvo. O processo de aprendizado, portanto, é não supervisionado por não dispor de um atributo alvo.

Na abordagem de *clustering*, as instâncias são comparadas entre si e organizadas em grupos. Os grupos resultantes deste processamento poderão atuar como o conjunto de

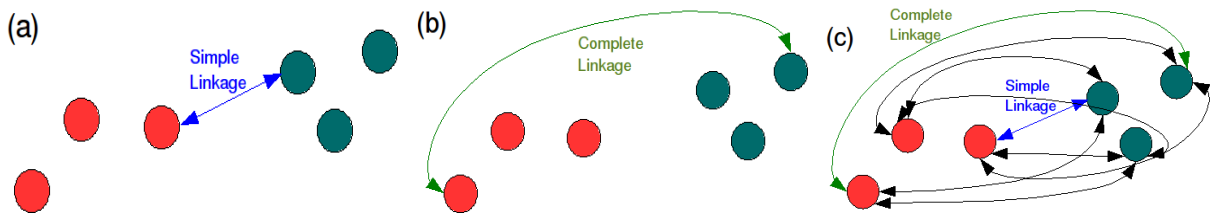


Figura 2.9 – *Simple Linkage* (a), considera a máxima similaridade, ou seja, os termos mais próximos. *Complete Linkage* (b), considera a mínima similaridade, ou seja, os termos mais distantes. *Average Linkage* (c), considera a média das similaridades entre todos os termos de cada *cluster*

atributos-alvo na categorização de textos. Desta maneira, aplicam-se critérios que determinem o grau de semelhança entre as instâncias. Em geral, os grupos resultantes serão compostos de instâncias com alto grau de semelhança entre si e as instâncias de grupos distintos deverão ter um baixo grau de semelhança entre si.

Deste problema emergem dois desafios:

1. **agrupar as instâncias** utilizando-se de critérios que determinem a semelhança entre as instâncias de um mesmo grupo e que as diferenciem dos demais grupos;

2. **rotular os grupos**, uma tarefa opcional[1], cuja necessidade depende da aplicação e, freqüentemente, realizada através de intervenção manual.

É importante notar que os métodos de agrupamento descritos a seguir têm sido utilizados para solucionar problemas nos mais diversos campos do conhecimento e a natureza dos dados, por esta razão, varia consideravelmente. Os dados podem ser as linhas de uma tabela em um banco de dados, os *pixels* de uma imagem, os *frames* de um vídeo, os documentos em um sistema de arquivos, entre outros. No âmbito deste estudo, passaremos a enfocar as situações em que o *data set* é um *corpus*, os documentos que compõem o *corpus* são as instâncias e as palavras que os compõem são os atributos destas instâncias ou, na maioria das vezes, se tornam a matéria-prima da qual obtemos os atributos.

Segundo Tan *et al.* [TSK09], várias são as estratégias empregadas nos algoritmos de *clustering*. Elas podem variar segundo:

1. a quantidade de grupos ao quais uma instância pode ser atribuída:

   (a) os algoritmos que atribuem cada instância a um único grupo são denominados algoritmos de *hard clustering*;

[1] Suponha, por exemplo, uma conversão de uma imagem *bitmap* com profundidade de *pixel* de 32 bits para uma imagem indexada com profundidade de 8 bits [Alp04]. As cores semelhantes serão agrupadas e o grupo será substituído por uma única cor, representada por um índice. Fica clara a absoluta inutilidade de se atribuir nomes a cada grupo.

   (b) os que atribuem a múltiplos grupos se denominam algoritmos de *soft clustering* que tendem a ser mais lentos que os de *hard clustering*, cuja complexidade geralmente aproxima-se de $O(n)$, ao passo que os de *soft clustering* ficam em torno de $O(n^2)$ [SKK00];

2. a relação entre os grupos:

   (a) os algoritmos hierárquicos organizam os grupos em forma de árvores de categorias, denominadas dendogramas, onde os nós folhas representam as instâncias, a raiz é um único *cluster* mais genérico que os demais e os nós intermediários representam *clusters* de variado grau de especificidade [HTF+05, TSK09]. Há três maneiras populares de medição da similaridade entre os *clusters*, conforme ilustrado na Figura 2.9:

      i. *Simple Linkage*: dados dois *clusters* P e Q, a similaridade entre os dois é definida como a maior similaridade entre duas instâncias $p \in P, q \in Q$ [TSK09], tende a definir grupos muito grandes [HTF+05];

      ii. *Complete Linkage*: dados dois *clusters* P e Q, a similaridade entre os dois é definida como a menor similaridade entre duas instâncias $p \in P, q \in Q$ [TSK09], tende a definir grupos compactos, mas pode haver instâncias em um grupo mais próximas de instâncias de grupos vizinhos que de instâncias de seu próprio grupo [HTF+05];

      iii. *Average Linkage*: dados dois *clusters* P e Q, a similaridade entre os dois é definida como a média das similaridades entre todas duas instâncias $p \in P, q \in Q$ [TSK09]. Equilíbrio entre prós e contras das medidas anteriores [HTF+05]. Observando-se a Figura 2.9, nota-se que a opção pelo *Average Linkage* eleva a complexidade do algoritmo.

   (b) os algoritmos particionais ou *flat* não organizam os grupos em hierarquias [TSK09];

3. a completude das atribuições:

   (a) os algoritmos que descartam instâncias são chamados parciais [TSK09];

   (b) aqueles que atribuem todas instâncias aos grupos resultantes são os completos [TSK09];

4. o critério de atribuição de instâncias aos grupos:

   (a) os algoritmos baseados em protótipo são aqueles que representam o grupo mediante um indivíduo ideal, que pode ser uma das instâncias (medóide) ou calculado a partir das instâncias integrantes do grupo (centróide) [TSK09];

   (b) os algoritmos baseados em grafos determinam a atribuição ao grupo pela existência de alguma relação de pertinência entre seus indivíduos [TSK09];

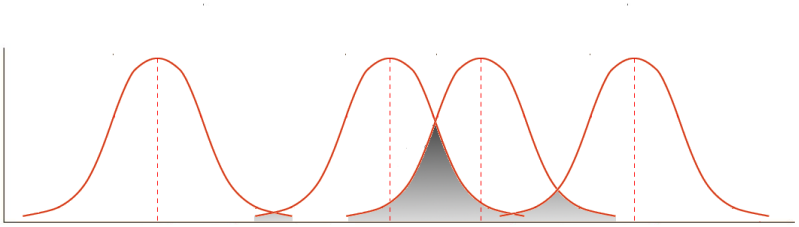


Figura 2.10 – O algoritmo K-Means pressupõe que as instâncias sejam resultado da superposição de distribuições gaussianas que compartilham mesma variância

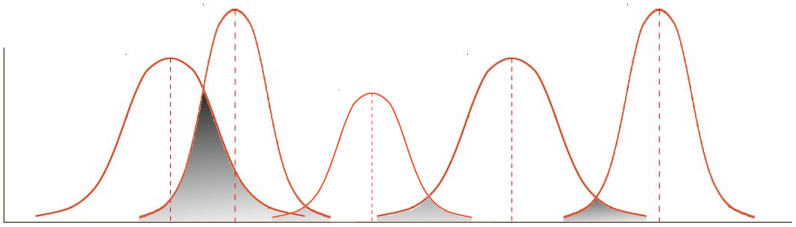


Figura 2.11 – O algoritmo EM pressupõe que as instâncias sejam resultado da superposição de distribuições gaussianas; as variâncias poderão ser distintas

(c) os algoritmos baseados em densidade delimitam os grupos detectando regiões com maior incidência de instâncias [TSK09].

2.3.1 O Algoritmo *K-Means*

Entre os algoritmos clássicos de *hard clustering* e, ainda, muito popular, está o *K-means*, proposto por MacQueen [Mac67], que pressupõe que a relação de pertinência entre as instâncias e grupos obedece a funções de distribuição de probabilidade [Mit97]. O K-means busca descobrir os centróides de cada grupo estimando as médias geradoras de cada grupo/distribuição e quais instâncias foram geradas por quais distribuições [Mit97, Alp04].

Para tanto, realiza-se iterações em 2 passos:

1. atribui cada instância ao centróide mais próximo/semelhante;

2. recalcula os centróides como o ponto médio das instâncias a ele atribuídas;

estas iterações encerram-se quando da convergência dos centróides [TSK09].

Visto que desejamos atuar no ramo do Direito, ressaltamos que o entrelaçamento de diferentes assuntos é reconhecidamente a maioria dos casos [CAKZ+05]. Por esta razão, espera-se que soluções de *hard clustering* tenham mais efetividade apenas ao se processar documentos que discorram acerca de um único tema.

2.3.2 Algoritmo *Expectation-Maximization* (EM)

O algoritmo EM, proposto por Dempster, Laird e Rubin [DLR77], assim como o K-Means, pressupõe que os grupos são determinados por distribuições de probabilidade gaussianas. No entanto, enquanto no K-Means as distribuições de probabilidade compartilham as mesmas variâncias, o EM admite a possibilidade de múltiplas variâncias [Mit97], conforme se pode verificar ao se comparar as Figuras 2.10 e 2.11. Na verdade, o algoritmo K-Means é um caso especial do algoritmo EM. Sendo um algoritmo de *soft clustering*, o EM admite que as instâncias possam estar vinculadas a mais de um grupo. Ele inicializa o processo usando o K-Means para estimar os grupos e suas médias iniciais e passa a calcular a probabilidade de que as instâncias estejam nos demais grupos [Mit97]. Para tanto, realiza iterações em 2 passos com a seguinte forma geral:

1. ***Expectation-Step***: Calcula-se a probabilidade $P(C_j|d_i,\Theta)$ da Classe C_j dado o documento d_i e $\Theta = (\mu, \sigma)$;

2. ***Maximization-Step***: Calcula-se novas médias, maximizando-se as probabilidades do *Expectation-Step*;

Pode se repetir as iterações até a convergência dos parâmetros ou até que a sua mudança seja inferior a um valor limite especificado.

Este algoritmo apresenta, no entanto, sérias dificuldades em convergir ou converge para uma solução inadequada quando o conjunto de dados é muito grande ou inicializado erroneamente. Suas variações FREM, on-line EM e *Scalable* EM também são altamente problemáticas em presença de grande volume de dados [CAKZ+05].

2.3.3 Agrupamento Semi-Supervisionado

De acordo com Grira, Crucianu e Boujemaa [GCB05], o agrupamento semi-supervisionado é uma forma de agrupamento na qual se impõe alguma restrição, normalmente nas formas *must-link* ou *cannot-link* que provê supervisão, embora limitada. Acrescentam ainda que o conhecimento representado por estas restrições é insuficiente para uso em aprendizado supervisionado. Assim, a combinação entre a aplicação da função de similaridade e alguma restrição guiam o procedimento de atribuição de instâncias aos *clusters*.

Ainda segundo Grira, Crucianu e Boujemaa [GCB05], os algoritmos de *clustering* semi-supervisionado se dividem em dois tipos: (1) aqueles que aplicam as restrições na função de similaridade e, (2) aqueles que aplicam as restrições no algoritmo do *clustering* propriamente dito. O algoritmo semi-supervisionado de Aggarwal, Gates e Yu [AGY04], estudado na Seção 3.4, aplica sua restrição no algoritmo de *clustering*. Mas, o faz apenas na inicialização dos dados, atribuindo cada instância a um grupo. As iterações subseqüentes não são influenciadas por qualquer restrição.

2.4 Pré-Processamento

Para que se possa classificar ou agrupar documentos, é preciso que as instâncias contenham as informações na forma adequada para a realização de operações de comparação. Se os atributos das instâncias se compõem de colunas em uma tabela de um banco de dados, é mais provável que não seja necessário nenhum processamento prévio. No entanto, é altamente provável que haja necessidade de formatação de dados em outros casos. Em nosso estudo, as instâncias se compõem de textos em linguagem natural e, portanto, deverão ser formatadas. Alguns destes procedimentos são muito conhecidos e usados largamente:

1. ***Parsing***, consiste em recortar o texto, dele extraindo as palavras que o compõem, etiquetando-as e realizando a análise sintática, identificando os grupos constituintes de acordo com uma gramática [MS00];

2. ***Stemming*** é o processo de normalização pelo qual buscamos reverter palavras, derivadas ou flexionadas para uma forma normal comum a todas as suas variações. Esta forma normal pode ser a raiz da palavra, "altamente" se reverterá para "alto", por exemplo; ou pode ser o seu *stem*, "alt", neste caso. A priori não importa se a reversão deverá remeter à raiz ou ao *stem*, o mais importante é que as várias flexões/derivações sejam mapeadas para uma mesma partícula [CDH+01];

3. **Lematização** é o processo de normalização em que se converte uma palavra inflexionada para uma forma não flexionada: o lema ou lexema correspondente [MS00]. Diferenças semânticas não são levadas em consideração. Os seguintes benefícios podem ser obtidos ao substituir as palavras dos documentos pelos seus lemas: eliminar ambigüidades léxicas, evitando contabilizar sob um mesmo atributo palavras com grafias iguais mas de sentidos diversos [MS00]; e contabilizar sob um mesmo atributo palavras com grafias diferentes que, por apresentarem sentidos muito próximos (por exemplo o mesmo verbo em diferentes flexões) compartilham o mesmo lema contabilizando-as sob um mesmo atributo e, assim, propiciando, não apenas a redução de sua dimensionalidade, mas, também, elevando a semelhança entre os documentos nos quais estas palavras ocorrem [BR04, HT06, Str05, KLJ+04, Gon05];

4. **Contabilização de freqüências** é um procedimento típico da abordagem *bag of words*, que considera que um documento é um "saco de palavras", desconsiderando a seqüência na qual elas ocorrem no texto, ou as relações sintáticas contidas nas orações. Esta abordagem reconhece apenas uma relação entre as palavras, que consiste no fato de se encontrarem no mesmo documento e limita-se a valorar a freqüência de suas ocorrências. A mera contabilização de freqüências pode, no entanto, induzir a erros de super/subvalorização de palavras. Considere, por exemplo, um texto de

10.000 palavras, onde encontramos 10 ocorrências da palavra “recorrer” e compare com um outro texto, composto de 200 palavras, onde detectamos 5 ocorrências da palavra “divisão”. Ora, o cálculo da freqüência absoluta indica que “recorrer” desempenha um papel mais preponderante no conjunto de dados, a despeito de representar 0,1% do texto em face dos 2,5% representados por “divisão”. Assim, o cálculo de freqüência de palavras, normalmente é acompanhado por algum método de normalização. O cálculo do percentual, aqui exposto é um método simples de se atingir tal objetivo. Outros métodos foram propostos, sendo o TF-IDF [LSZ04] largamente utilizado. Este método leva em consideração, não apenas a freqüência dos termos em cada documento, mas, também, a quantidade de documentos em que o termo ocorre. Desta maneira, temos o cálculo de freqüência de um dado termo k em um documento j:

$$TF(t_k, d_j) = \begin{cases} 1 + \log \#(t_k, d_j) & \text{se } \#(t_k, d_j) > 0 \\ 0 & \text{se } \#(t_k, d_j) \leq 0 \end{cases} \tag{2.4}$$

A freqüência inversa de documento, que exprime a relevância do termo, dada a quantidade de documentos em que ele ocorre, é dada por:

$$IDF = \log \frac{|D|}{\#D(t_k)} \tag{2.5}$$

Finalmente, temos que:

$$TF - IDF_{ij} = TF_{ij}.IDF_i \tag{2.6}$$

2.5 Funções de Proximidade

Durante o processo de agrupamento, ou em outras circunstâncias, é necessário realizar o cálculo de proximidade semântica entre pares de termos ou documentos. De acordo com Tan, Steinbach e Kumar [TSK09], a proximidade pode ser a diferença ou a similaridade entre as instâncias. Duas abordagens populares são:

1. **Distância euclidiana** é uma medida de diferença muito popular onde os termos são tratados como “vetores num espaço semântico” e, assim, aplica-se o cálculo de distância euclidiana entre eles, ou seja, para dois termos $\vec{t_1} = \{t_{1_0}, ..., t_{1_n}\}$ e $\vec{t_2} = \{t_{2_0}, ..., t_{2_n}\}$, a distância entre eles será dada por

$$\sqrt{\sum_{i=0}^{n} (t_{1_i} - t_{2_i})^2} \tag{2.7}$$

onde n é a quantidade de dimensões.

Quanto **menor** a distância entre duas instâncias, maior a probabilidade de que sejam atribuídas a um mesmo grupo.

2. **Cosseno do ângulo** dos vetores $\vec{t_1}$ e $\vec{t_2}$, é uma medida de similaridade dada por

$$\cos(\vec{t_1}, \vec{t_2}) = \frac{\vec{t_1}.\vec{t_2}}{\|\vec{t_1}\|.\|\vec{t_2}\|} \tag{2.8}$$

   Quanto **maior** a similaridade entre duas instâncias, maior a probabilidade de que sejam atribuídas a um mesmo grupo.

2.6 Métodos de Validação de Classificação

Para aferir a qualidade dos resultados, utilizam-se várias medidas que expressam o grau de qualidade destes métodos. Segundo Tan, Steinbach e Kumar [TSK09], as medidas mais comuns são compostas dos seguintes elementos básicos:

1. **Verdadeiros Positivos (VP)**: instâncias corretamente classificadas como pertencentes a uma classe específica;
2. **Verdadeiros negativos (VN)**: instâncias corretamente classificadas como não pertencentes a uma classe específica;
3. **Falsos positivos (FP)**: instâncias erroneamente classificadas como pertencentes a uma classe específica;
4. **Falsos negativos (FN)**: instâncias erroneamente classificadas como não pertencentes a uma classe específica.

Ainda conforme Tan, Steinbach e Kumar [TSK09], baseadas nas contagens destes elementos, destacamos as seguintes medidas:

1. a **acurácia** ou **precisão**: é o percentual de acertos, $\frac{VP+VN}{VP+FP+VN+FN}$, ou seja, a proporção das instâncias corretamente obtidas pelo total de instâncias;
2. a **abrangência** ou ***recall***: $\frac{VP}{VP+FP}$,ou seja, a quantidade das instâncias corretamente obtidas pela quantidade de instâncias realmente pertencentes à classe alvo;
3. a **medida F** ou ***F-measure***: é uma média entre precisão e abrangência, podendo atribuir pesos valorando uma ou outra medida, sendo que, mais freqüentemente, é usada a fórmula: $F = \frac{(1+\beta^2).P.R}{\beta^2.P+R}$, onde P é a precisão, R é o *recall* e β é o peso, definido por:

β	Peso
0,5	precisão é mais valorizada
1	-
2	*Recall* é mais valorizado

2.7 Métodos de Validação de Grupos

De acordo com Tan, Steinbach e Kumar [TSK09], os métodos de validação são taxonomicamente divididos em:

1. **Supervisionados**, também chamados de **índices externos**, quando se utilizam de informação adicional além da presente no conjunto de dados, são detalhados na Seção 2.7.1;

2. **Não supervisionados**, também chamados de **índices internos**, quando utilizam, exclusivamente, informação contida no conjunto de dados, são detalhados na Seção 2.7.2;

3. **Relativos**, métodos supervisionados ou não supervisionados quando usados para comparar diferentes experimentos.

2.7.1 Métodos de Validação de Grupos Supervisionados

Segundo Tan, Steinbach e Kumar [TSK09], dentre os métodos de validação supervisionados, baseados no pressuposto de que a um grupo corresponda uma classe, encontramos:

1. os **métodos orientados a classificação**

   (a) métodos que qualificam a presença/ausência das classes nos grupos através da acurácia, abrangência e *F-Measure*;

   (b) métodos que consideram a presença/ausência das classes nos grupos, sem qualificá-las. Seja $p_{ij} = \frac{m_{ij}}{m_i}$, m_i a quantidade de instâncias no grupo i e m_{ij} a quantidade de instâncias da classe j no grupo i, os métodos orientados a classificação seriam:

      i. a **entropia**: $\sum_{i=1}^{K} \frac{m_i.(-\sum_{j=1}^{L} p_{ij} log_2 p_{ij})}{m}$;

      ii. a **pureza**: $\sum_{i=1}^{K} \frac{m_i}{m} p_i$;

2. e os **métodos orientados a semelhança**, como

   (a) a **estatística Rand**: $\frac{f_{00}+f_{11}}{f_{00}+f_{01}+f_{10}+f_{11}}$

   (b) o **coeficiente de Jaccard**: $\frac{f_{11}}{f_{01}+f_{10}+f_{11}}$

   onde f_{ij} é a quantidade de pares de instâncias, para

$$i = \begin{cases} 1 & instancias\ de\ mesma\ classe \\ 0 & instancias\ de\ classes\ diferentes \end{cases} \quad j = \begin{cases} 1 & instancias\ no\ mesmo\ grupo \\ 0 & instancias\ em\ grupos\ diferentes \end{cases}$$

Tabela 2.4 – Cálculo de Coesão e Separação de agrupamentos

	Grafos	Protótipo
Coesão	$\sum_{\substack{x,y \in C_i \\ x \neq y}} f(x,y)$	$\sum_{x \in C_i} f(x,C_i)$
Separação	$\sum_{\substack{i=1 \\ j \neq i}} \sum_{\substack{x \in C_i \\ y \in C_j}} f(x,y)$	$f(C_i,C_j)$

2.7.2 Métodos de Validação de Grupos Não Supervisionados

2.7.2.1 Medidas de Coesão e Separação de Grupos

Dentre os métodos de validação não supervisionados, encontramos medidas baseadas em coesão interna do grupo, ou seja, a proximidade entre as instâncias de mesmo grupo e separação dos grupos, o grau de afastamento entre os grupos. Para grupos baseados em grafos, as medidas implicam comparações entre instâncias. Em grupos baseados em protótipo a coesão resulta de comparação entre as instâncias do grupo e o protótipo e a separação apenas entre protótipos [TSK09]. A Tabela 2.4 apresenta o cálculo de medidas de coesão e separação de agrupamentos, onde f é a função de proximidade.

2.7.2.2 Coeficiente de Silhueta

O coeficiente de silhueta [Rou87], baseado em grafos, combina coesão e separação para determinar se uma instância está bem inserta no grupo ou se está em região *inter-cluster*. Este coeficiente permite a visualização gráfica da qualidade dos grupos. O coeficiente de uma instância é dado por

$$s_i = \frac{(b_i - a_i)}{max(a_i, b_i)}$$

onde

$$a_i = \frac{\sum_{\substack{j \neq i \\ i,j \in C_k}} f(x_i, y_j)}{|C_k|}$$

e

$$b_i = min_{k \in C} \left(\frac{\sum_{\substack{j \in C_k \\ i \notin C_k}} f(x_i, y_j)}{|C_k|} \right)$$

o coeficiente de silhueta médio de um grupo é dado por

$$s_{C_j} = \frac{\sum_{i=1}^{|C_j|} s_i}{|C_j|} \tag{2.9}$$

e o coeficiente de silhueta médio de todo o agrupamento é dado por

$$s_C = \frac{\sum_{i=1}^{|C|} s_{C_i}}{|C|} \tag{2.10}$$

2.7.2.3 A família de Índices Dunn

A família de índices Dunn [Dun73] *apud* [BLA+97], também pondera coesão e separação dos grupos, apresentando a seguinte fórmula geral:

$$D_C = \frac{min_{i \neq j}(f(C_i, C_j))}{max_{1 \leq l \leq k}(\Delta(C_l))} \tag{2.11}$$

onde $\Delta(C_n)$ é o diâmetro do grupo n. Este índice pode, erroneamente reportar uma baixa qualidade do *clustering* quando houver ao menos um *cluster* com diâmetro relativamente grande e ao menos um par de *clusters* muito próximos [SEW03]. No índice original de Dunn, a distância entre *clusters* é a distância entre os vizinhos mais próximos (*single linkage*) e o diâmetro é a maior distância entre uma instância e o respectivo centróide. Bezdek [BLA+97] apresentou experimentos demonstrando resultados superiores ao usar *average linkage* para determinar a distância entre os *clusters* e calcular o diâmetro como o dobro da distância média entre as instâncias e seus respectivos centróides.

2.7.2.4 Índice Davies-Bouldin

O índice Davies-Bouldin [DB79] utiliza a razão entre a dispersão interna do grupo e a separação entre os grupos, dada por:

$$DB = \frac{\sum_{i=1}^{|C|} R_i}{|C|} \tag{2.12}$$

onde

$$R_i = max_{\substack{j \neq i \\ 1 \leq j \leq |C|}} \left(\frac{s(C_i) + s(C_j)}{f(C_i, C_j)} \right)$$

e

$$s(C_n) = \frac{\sum_{x \in C_n} f(x, c_n)}{|C_n|}$$

para C_n = n-ésimo *cluster* e c_n = centróide do n-ésimo *cluster*.

2.7.2.5 Medida Λ

Stein, Eissen e Wissbrock [SEW03] ressaltam que os índices da família Dunn e o índice Davies-Bouldin assumem o modelo baseado em protótipo, pressupondo *clusters* de forma

esférica e, assim, a aplicabilidade do índice torna-se questionável quando este modelo não se aplicar.

A medida Λ, proposta por Stein e Niggemann [SN99] adota modelo baseado em grafo e considera a densidade dos *clusters* no cálculo dado por

$$\sum_{i=1}^{|C|} |C_i|.\lambda_i \tag{2.13}$$

onde

$$\lambda_i = min \sum_{(u,v)\in E'} w(u,v)$$

E' é um conjunto de arcos tal que o C_i seja desconexo e $w(u,v)$ é o peso do arco que conecta u e v.

2.7.2.6 Medida $\bar{\rho}$

A medida $\bar{\rho}$ ou medida de densidade esperada, proposta por Stein, Eissen e Potthast [SEP06], também adota modelo baseado em grafo e pondera a densidade dos *clusters* em relação à densidade do agrupamento no seu cálculo, dada por

$$\sum_{i=1}^{k} \frac{|C_i|}{|C|} . \frac{w(C_i)}{|C_i|^{\theta}} \tag{2.14}$$

onde k é a quantidade de classes e

$$\theta = \frac{ln(w(C))}{ln(|C|)}$$

e

$$w(C) = |C| + \sum_{\substack{x \neq y \\ x,y \in C}} f(x,y)$$

Sendo a densidade baseada na soma das similaridades entre as instâncias, quanto maior for a densidade dos grupos em relação à densidade do agrupamento, maior será o valor desta medida e, portanto, melhor será a qualidade do agrupamento [SEP06].

2.7.2.7 Medida *Relative Hardness*

A medida *relative hardness* (RH), de Pinto e Rosso [PR07], concatena os documentos de cada categoria, obtendo um único vetor por categoria, ou seja, é baseada em protótipo, e é dada por

$$\frac{\sum_{i,j=1,i<j}^{n} f(CAT_i, CAT_j)}{n(n-1)/2} \tag{2.15}$$

onde n é a quantidade de categorias e CAT_i é a i-ésima categoria.

Esta medida é uma soma das similaridades *intra-cluster* e, portanto, como afirmado pelos seus autores, quanto menor o seu valor, melhor é a qualidade do agrupamento [PR07].

2.8 Métodos de Comparação de Algoritmos de Aprendizado de Máquina

Segundo Demšar [Dem06], ao realizar-se comparativos entre as performances de algoritmos de aprendizado de máquina é necessário certificar-se de que as diferenças de performance aferidas nos testes sejam realmente significativas e não pequenas diferenças aleatórias resultantes de características específicas do conjunto de dados utilizado. Em seu estudo, Demšar enfoca a comparação de algoritmos de aprendizado supervisionado. García *et al.* [GML$^+$09] usam teste de significância para comparar algoritmos genéticos e Cappelleri *et al.* [CCR$^+$02] o usam para avaliar análise de tratamento de diabetes. O teste de significância que aplicamos em nosso exemplo de uso segue a linha de Mukhopadhyay *et al.* [MM08]. Esses últimos autores calculam os índices de qualidade dos conjuntos de agrupamentos e realizam o teste de significância para determinar que os índices de qualidade dos agrupamentos obtidos pelos algoritmos por eles propostos são significativamente superiores.

Demšar [Dem06] apresenta vários métodos estatísticos de teste de significância. Por aplicarem-se à comparação de pares de algoritmos, não serem paramétricos e serem facilmente implementáveis, destacamos os seguintes testes:

1. **Teste de contagem de vitórias, derrotas e empates – Teste de Sinal** [She04, Sal97] *apud* [Dem06]: Este é o teste que menos rejeita a hipótese nula [Dem06]. Neste teste, calcula-se um escore E contabilizando-se a quantidade de conjuntos de dados nas quais o algoritmo proposto teve melhor performance. Os empates contam como 0,5 e o escore final é truncado para o inteiro imediatamente inferior. Se

$$E \geq \frac{N}{2} + \frac{1,96\sqrt{N}}{2} \tag{2.16}$$

   , para N = quantidade de conjuntos de dados, então a diferença entre os resultados sendo comparados é significativa, com 5% de confiança;

2. **Wilcoxon *signed-ranks test*** [Wil45]: Neste teste, para cada conjunto de dados, calcula-se a diferença de performance entre os algoritmos 1 e 2, $\Delta_i = a_{1i} - a_{2i}$. Atribui-se *ranks* aos valores absolutos destas diferenças. Em caso de empates, atribui-se a média dos *ranks*. Para cada algoritmo, somam-se os *ranks* nos quais sua performance superou a do outro algoritmo, truncando-se o resultado. A seguir, atribui-se a

T a menor das somas de *ranks*. Por fim, calcula-se

$$z = \frac{T - \frac{N(N+1)}{4}}{sqrt\frac{N(N+1)(2N+1)}{24}} \tag{2.17}$$

onde N é a quantidade de conjuntos de dados e, para $z < -1,96$, pode-se rejeitar a hipótese nula com 5% de confiança.

2.9 Considerações Finais

Revisamos a abordagem do Aprendizado Supervisionado, que só permite obter soluções se dispusermos de um atributo alvo e somente o conjunto de valores nele encontrados. Se, no conjunto de treino, o atributo alvo apresenta os valores $\{a, b, c\}$, jamais se poderá obter uma solução como d, ainda que esta seja a solução correta. O aprendizado consiste em determinar a **forma** de se chegar a uma solução específica, dado um conjunto de possíveis soluções.

O objetivo deste estudo é trabalhar com documentos jurídicos. Assim, nossas instâncias serão documentos contendo grande quantidade de termos, versando sobre ampla gama de assuntos. Uma vez que o trabalho não é determinar se um dado caso tem veredito favorável ou não, mas saber se o caso é semelhante a um ou mais casos já analisados e quais seriam estes, **não temos um conjunto fixo e conhecido de atributos alvo** e, portanto, necessitamos, primeiramente, determinar quais os possíveis valores que o atributo alvo poderá assumir.

Para tanto, estudamos, na seção anterior, os algoritmos de aprendizagem não supervisionada, que, através do agrupamento de instâncias semelhantes, buscam determinar quais são os possíveis valores que pode assumir o atributo alvo. Foram aqui apresentadas algumas abordagens para realização de agrupamento, já com foco no agrupamento de documentos.

No próximo capítulo, revisamos trabalhos relacionados, que apresentam processo em que as instâncias são agrupadas e os grupos obtidos determinam o conjunto de possíveis atributos-alvo para uso de categorizador.

3. Trabalhos Relacionados

3.1 Considerações Iniciais

Buscando a bibliografia pertinente, percebe-se que ainda se dispõe de poucos relatos de experimentos utilizando *clustering* como auxiliar de algoritmos de categorização de documentos não previamente rotulados. A Seção 3.3.5 apresenta um relato neste sentido. A maioria dos trabalhos relacionados foi agrupada na Seção 3.2 e na Seção 3.3, de acordo com o algoritmo categorizador utilizado. A Seção 3.4 apresenta um trabalho relacionado cujos algoritmos empregados distingüem-se dos anteriores, sendo, assim, apresentado separadamente. Alguns trabalhos não utilizaram documentos contendo texto em linguagem natural em seus experimentos. No entanto, uma vez que utilizam processo em que se realiza *clustering* para auxiliar algoritmo categorizador, considerou-se relevante revisá-los. Os experimentos relatados neste capítulo, bem como seus resultados, não foram reproduzidos, tendo sido empreendidos pelos autores dos respectivos artigos.

3.2 Trabalhos Baseados em Classificadores Bayesianos

Nesta seção estão agrupados os trabalhos que utilizam classificador Bayesiano. Além disto, a maioria deles usa o algoritmo EM na fase de *clustering*. Os primeiros trabalhos da Subseção 3.2.1 não apresentaram experimentos com documentos contendo textos em linguagem natural sem formatação específica. Somente o último destes trabalhos tratou de documentos textuais. No entanto, os trabalhos relatados nesta subseção foram selecionados tendo em vista o processo de *clustering* como auxiliar da *categorização* proposto pelos autores. Ainda que divirjam no tipo de dado tratado no próximo capítulo, propõem metodologias que podem ser, também, utilizadas com documentos textuais.

3.2.1 Gerando Redes Bayesianas usando o Algoritmo EM

Friedman [Fri97, Fri98], propõe o uso do algoritmo EM [DLR77] para alterar redes bayesianas, melhorando sua performance. Seu estudo continua em parceria com Elidan [EF01] e Lotner e Koller [ELF+00].

Em seu primeiro trabalho, Friedman [Fri97] apresenta o *Model Selection EM*, MS-EM e o *Alternate Model Selection EM*, AMS-EM, que difere daquele por evitar convergência prematura para máximos locais. Estes algoritmos aprendem novas redes bayesianas na presença de variáveis ocultas ou de valores faltantes. Além da descoberta de variáveis, promove, também, inserção e remoção de arcos. Friedman ressalta que trabalhos anteriores realizam estas operações fora do EM e, a cada alteração, chamam o EM para estimar

os parâmetros da rede. Por esta razão, o EM recalcula todos os parâmetros da rede a cada chamada. Na proposta de Friedman, uma vez que as modificações estruturais da rede são realizadas dentro do EM, o recálculo dos parâmetros é limitado aos nodos afetados pela alteração.

Em seu segundo trabalho, Friedman [Fri98] rebatiza o MS-EM para *Structural EM*, SEM. Sua nova proposta, o *Bayesian Structural EM*, BSEM, difere do SEM, que realiza uma busca no espaço de estruturas $\times$ parâmetros, por realizar uma busca apenas no espaço de estruturas. Além disto, o SEM busca valores aproximados e o BSEM busca valores exatos. Na busca por valores exatos, o BSEM precisaria repetir muitos cálculos intermediários. Assim, os cálculos intermediários são armazenados em cache e a quantidade efetiva de cálculos é significativamente reduzida.

No terceiro trabalho [ELF$^+$00], o estudo se concentra na descoberta de variáveis ocultas que interagem com variáveis observadas. O método proposto realiza uma busca na rede por subestruturas, chamadas de "semi-cliques" pelos autores, que podem indicar a presença de uma variável oculta. Segundo os autores, um semi-clique é um relaxamento no número de vizinhos, definido como um conjunto de variáveis tais que cada variável tenha arestas conectando-a com, pelo menos, metade das demais variáveis do conjunto. Ao encontrar um semi-clique, o algoritmo realiza uma inserção de uma nova variável, quebrando o semi-clique. Se, após um processo de aprendizado, a nova estrutura apresentar melhores resultados que a original, a nova variável é aceita na estrutura.

No quarto trabalho, Elidan e Friedman [EF01] propõem método para descobrir a dimensionalidade de variáveis ocultas. Para cada nova variável oculta H, seria necessário realizar muitas execuções do EM, variando a cardinalidade desta variável. Por isto, os autores consideraram que o EM teria um custo computacional muito elevado e propõem novo algoritmo inspirado por *clustering aglomerativo* e técnicas de fusão de modelos bayesianos. Além disto, os autores destacam que quando há muitas variáveis ocultas a serem determinadas, o custo de processamento escala rapidamente devido à influência da alteração de uma variável oculta sobre as demais variáveis ocultas. Por esta razão, o algoritmo trata primeiramente as variáveis ocultas com menor impacto sobre outras.

Os experimentos acima descritos foram realizados sobre vários conjuntos de dados. Nos dois primeiros, Friedman utilizou dados gerados artificialmente: 1) uma rede para decisão acerca de concessão de seguros de carro e 2) uma rede para monitoração de pacientes sob tratamento intensivo. No terceiro experimento, os autores acrescentaram conjuntos de dados reais de 1) bolsa de valores e 2) dados de pacientes de tuberculose. No quarto experimento, os autores não utilizaram o conjunto de dados de seguros de carro e acrescentaram o *corpus 20Newsgroups*, composto de 5.000 documentos. Para este último, um atributo continha o *newgroup* onde o documento foi postado (rótulo de classe) e os demais atributos representavam as 99 palavras mais freqüentes do vocabulário, exceto as *stopwords*, que foram removidas.

Nesta série de estudos vemos a evolução do trabalho de Friedman, buscando gerar redes bayesianas com o auxílio do EM. Em seus primeiros trabalhos não foram utilizados documentos em linguagem natural sem formatação específica. Porém no último trabalho foi utilizado o *corpus 20Newsgroups*. Os autores reportam apenas o ganho de performance tendo como *baseline* a rede bayesiana original.

3.2.2 Classificação de Texto num Modelo de Mistura Hierárquico para Pequenos Conjuntos de Treino

Toutanova *et al.* [TCP+01] propõem estender o categorizador *Naïve Bayes* utilizando um modelo de mistura hierárquica de tópicos, diferenciando termos de acordo com sua especificidade/generalidade. Os autores assumem uma hierarquia de tópicos pré-definida e buscam gerar automaticamente um modelo de probabilidade para documentos. Os parâmetros do modelo são aprendidos mediante o algoritmo EM [DLR77], que busca maximizar a verossimilhança nos dados de treino. Uma vez aprendidos os parâmetros, novos documentos são classificados, utilizando-se *Naïve Bayes*, computando e maximizando probabilidades de categorias em função das palavras contidas no documento, representadas como vetor de freqüências.

Este modelo hieráquico foi inspirado no modelo de Redução Hierárquica, *Hierarchy Shrinkage* [MRM+98]. Aqui, porém, os nodos intermediários da hierarquia representam níveis de abstração das palavras contidas nos documentos. Os autores assumem que cada palavra num documento é gerada por um nodo (nível de abstração) no caminho entre a classe do documento (nodo folha) e a raiz da hierarquia. Esta representação resulta numa mistura onde as probabilidades dos termos são compartilhadas por múltiplas classes (nodos folha). O nível de abstração de cada palavra é desconhecido e, assim, modelado como variável oculta.

A construção da árvore utiliza o modelo *Cluster-Abstraction* [Hof99] que gera modelos hierárquicos a partir de dados não rotulados utilizando o algoritmo EM [DLR77]. Diferentemente deste modelo, o modelo proposto pelos autores usa uma hierarquia pré-definida e dados rotulados para estimar os parâmetros do EM.

No *expectation step*, o algoritmo calcula a probabilidade $P(v|C,w_i)$ do nível v hieráquico, dadas a palavra w_i e a classe C, para cada par (C,v). No *maximization step*, o algoritmo calcula a probabilidade $P(w_i|v)$ da palavra w_i, dado o nível v da hierarquia e a probabilidade $P(v|C)$ do nível v dada a classe C. Como resultado, palavras mais genéricas têm probabilidades maiores em nodos mais próximos à raiz e palavras mais específicas de certas classes terão probabilidades maiores próximo aos respectivos nodos folha. Segundo os autores, dados empíricos mostram que basta realizar em torno de 2 a 5 iterações do EM para obter uma árvore com boas estimativas. Acima disto verificou-se a ocorrência de *overfitting*.

Obtido o modelo hieráquico, os autores apresentam duas maneiras de se utilizá-lo:

1. Para associar um único rótulo a cada documento, o algoritmo bayesiano seleciona a classe com a maior probabilidade dadas as palavras do documento, $P(c|d)$;

2. Para associar múltiplos rótulos sugerem o uso de valores de corte, recuperando as classes de maior probabilidade.

Para avaliar a performance do método proposto, os autores usaram dois *corpora*: *Reuters* 21578 e *20Newsgroups*. Os algoritmos comparados foram: *Naïve Bayes*, *Probabilistic Latent Semantic Analysis*, *Hierarchical Shrinkage*, KNN e SVMs.

O *corpus 20Newsgroups* possui, aproximadamente, 20.000 documentos, divididas em 20 grupos de, aproximadamente, 1.000 documentos. Foram selecionados 15 grupos para tornar o experimento semelhante ao relatado em [MRM+98]. Estes grupos possuem muitas semelhanças entre si e aproximadamente 4% dos documentos estão presentes em mais de um grupo. Os 15 grupos estão organizados em 5 categorias gerais e esta organização hieráquica foi adotada como modelo para o experimento (a raiz, 5 nodos intermediários e 15 nodos folha). O assunto da postagem foi incluído no documento. As letras foram convertidas para minúsculas. Não foi utilizado nenhum método de normalização. Foram removidas as *stopwords* e todas as palavras com freqüência inferior a 4 ocorrências no *corpus*. Vários treinos foram realizados, a cada treino variou-se a quantidade total de documentos, dividido igualmente por grupo.

O Modelo de Mistura Hierárquica proposto pelos autores obteve as melhores performances em 6 dos 8 experimentos, obtendo a segunda melhor performance nos demais. A diferença de performance entre os classificadores reduziu significativamente à medida que o conjunto de documentos aumentou de tamanho.

O segundo experimento utilizou o particionamento *ModApte*[1] do *corpus Reuters-21578* com as modificações usadas por Yang e Liu em [YL99]. Estas modificações consistem em selecionar somente documentos classificados em categorias que ocorrerem tanto no conjunto de treino quanto no de teste. As letras foram convertidas para minúsculas. Não foram removidas as *stopwords*. Foi realizada redução de dimensionalidade selecionando as N palavras com o menor impacto na incerteza das classes, onde $N \in \{1.000, 2.000, 10.000\}$.

Os autores consideraram que as categorias deste *corpus* não estavam apropriadas ao Modelo de Mistura Hierárquica proposto, pois 4 das 8 categorias previstas pertenciam ao domínio das finanças. Assim, decidiram usar um algoritmo aglomerativo para gerar novas categorias. Como resultado, obtiveram 4 categorias intermediárias e 90 categorias finais (nodos folha).

Usando este *corpus*, o classificador baseado em SVM apresentou a melhor performance em todas as medições. O Modelo de Mistura Hierárquica foi o segundo melhor em 4 das 7 aferições. Os autores passaram a estudar a hipótese de que a baixa performance decorra

[1] http://kdd.ics.uci.edu/databases/reuters21578/README.txt , VIII. B.

da fase de pré-processamento, pois não realizaram a normalização das palavras e acreditavam que houvesse diferenças nas listas de *stopwords* e nos esquemas de peso dos termos. Nota-se que os autores apresentam informação aparentemente contraditória em seu artigo, pois afirmam que não realizaram a remoção de *stopwords*. Em comparação com o Modelo de Redução Hierárquica, obteve melhor performance em 6 das 7 medições. Superou o *Naïve Bayes* em todas as medições realizadas.

3.2.3 Classificação de Textos Semi-Supervisionada Usando EM Particional

Cong, Lee, Wu e Liu [CLW+04], questionam o pressuposto comumentemente assumido em processos de aprendizado semi-supervisionado, no qual espera-se que os grupos gerados tenham correspondência um-para-um com as categorias constantes dos dados pré-rotulados. Desta maneira, propõem processo de agrupamento hierárquico usando *hard clustering* e, após, aplicam o EM em cada partição. Por fim, usam os dados rotulados para podar a árvore de maneira que os nodos restantes da árvore satisfaçam o pressuposto de correspondência um-para-um com as categorias.

Os autores afirmam que o EM tem má performance na presença de mais de 2 distribuições. Assim, empregam o particionamento recursivo dos dados de maneira a garantir que haja apenas duas distribuições em cada partição. Para tanto, usam um algoritmo de *hard clustering*, que consiste em:

1. biparticionar os dados randomicamente;
2. treinar um classificador bayesiano para corrigir a distribuição dos documentos entre as partições;
3. repetir recursivamente os passos acima para cada partição.

Os autores prevêem a iteração de passos do algoritmo acima até a convergência. A condição de parada da recursividade é a presença de, no máximo, 2 documentos rotulados.

Após o particionamento, o algoritmo proposto realiza a poda da árvore. Há 2 razões para a poda: (i) eliminar partições muito pequenas e (ii) eliminar problemas de *overfitting*. Para tanto, o algoritmo poda a árvore começando pelos nodos folha e caminhando em direção à raiz. Sempre que a soma dos erros de classificação dos nodos filhos for maior que os erros de classificação do nodo pai, poda-se os nodos filhos e retorna os erros de classificação do nodo pai. Em caso contrário, retorna-se a soma dos erros de classificação dos nodos filho.

Obtendo a árvore, o algoritmo inicia a execução do EM em cada partição, tendo, agora, como pressuposto, que exista uma correspondência um-para-um das distribuições com as categorias dos documentos. No entanto, os autores relatam que, ainda assim, em alguns casos, esta correspondência não ocorre e isto é detectado quando aumenta o número de iterações do EM. Por isto, a cada iteração, é realizada uma classificação com algoritmo

bayesiano usando *cross validation* e, verificando-se que a acurácia diminui, encerra-se o EM sem esperar por sua convergência.

A partir de então, é possível realizar a classificação de novos documentos. O processo de classificação é descrito pelos autores em dois passos:

1. O documento a ser classificado é agrupado hierarquicamente, iniciando pela raiz da árvore e seguindo até um nodo folha;

2. Ao atingir um nodo folha, o documento é classificado usando um classificador bayesiano, com os parâmetros obtidos pelo EM, usando a equação

$$P(C_j|d_i) = \frac{P(C_j)\prod_{k=1}^{|d_i|} P(w_{d_i.k}|C_j)}{\sum_{r=1}^{|C|} P(C_r)\prod_{k=1}^{|d_j|} P(w_{d_i.k}|C_r)} \tag{3.1}$$

   onde C_j é uma classes e d_i é um documento e $w_{d_i.k}$ é a palavra na posição k do documento d_i. A probabilidade $P(w_i|C_j)$ é substituída por $P'(w_i|C_j)$, conforme a equação

$$P'(w_i|C_j) = \frac{1+\sum_{i=1}^{|D|} N(w_i,d_i)P(C_j|d_i) - N(w_i,d_v)P(C_j|d_v)}{|V|+\sum_{g=1}^{|V|}(\sum_{i=1}^{|D|} N(w_g,d_i)P(C_j|d_i) - N(w_i,d_v)P(C_j|d_v))}$$

   cujos parâmetros foram obtidos através do EM.

Para os experimentos, os autores selecionaram os *corpora*:

1. **20Newsgroups**: contendo 19.997 artigos divididos quase igualmente entre 20 categorias. Deste *corpus* foram derivados 2 grupos:

   (a) **20A**: dividido em 20 *datasets*, cada um composto de duas categorias: (i) o conjunto dos positivos composto de uma das 20 categorias originais do *20Newsgroups* e o dos negativos, composto das 19 categorias restantes;

   (b) **20B**: dividido em 20 *datasets*, cada um composto de duas categorias: (i) o conjunto dos positivos composto de tópicos selecionados randomicamente do *20Newsgroups* e o dos negativos, composto dos tópicos restantes;

2. **Reuters 21578**: contendo 12.902 artigos divididos em 135 tópicos. Também foram derivados dois grupos deste *corpus*:

   (a) **RA**: dividido em 10 *datasets*, cada um composto de duas categorias: (i) o conjunto dos positivos composto de uma das 10 categorias mais populosas do *Reuters 21578* e o dos negativos, composto dos documentos restantes do *Reuters 21578*;

(b) **RB**: dividido em 20 *datasets*, cada um composto de duas categorias: (i) o conjunto dos positivos composto de tópicos selecionados randomicamente das 10 categorias mais populosas do *Reuters 21578* e o dos negativos, composto dos tópicos restantes;

O pré-processamento não envolveu *stemming*. Foram removidas as *stopwords* e as palavras que apareciam em menos de 3 documentos[2].

Para cada um dos grupos (20A, 20B, RA e RB), os autores realizaram 10 experimentos, selecionando randomicamente os documentos que compunham o conjunto de documentos rotulados. Todos os resultados apresentados são médias dos resultados dos 10 experimentos. Para os grupos 20A e 20B, o conjunto de teste continha 4.000 documentos, o conjunto de treino com documentos não rotulados continha 10.000 documentos e o conjunto de treino com documentos rotulados variou de tamanho entre 40 e 6.000 documentos. Para os grupos RA e RB, o conjunto de teste se compunha de 3.299 documentos, o conjunto de treino com documentos não rotulados continha 8.000 documentos e o conjunto de treino com documentos rotulados variou de tamanho entre 20 e 1.200 documentos.

Os algoritmos utilizados para comparação com o proposto foram: Naïve Bayes, EM, EM com interrupção das iterações[3] antes da convergência, M-EM, proposto por Nigam *et al.* [NMT+00].

Os autores utilizaram a medida F com $\beta = 1$ para avaliação de performance dos algoritmos. O algoritmo proposto apresentou performance superior aos demais algoritmos em todos os experimentos. A diferença de performance foi maior quando o conjunto de documentos rotulados era pequeno.

3.2.4 Analisando a Efetividade e Aplicabilidade do *Co-Training*

Nigam e Ghani [NG00] realizaram experimento analisando algoritmo de *Co-Training*, proposto por Blum e Mitchell [BM98] para conjuntos de dados disjuntos[4], como, por exemplo, um documento *web* e as palavras que ocorrem nos *hyperlinks* que o referem, para aumentar a performance de algoritmos de aprendizado quando se dispõe de dados rotulados e não rotulados. Os autores demonstraram que algoritmos que lidam com uma natural disjunção dos atributos do modelo de espaço vetorial obtêm melhor performance. Além disto, demonstram a possibilidade de tornar explícita uma disjunção dos atributos quando esta existir, mas for desconhecida.

Em seu artigo, os autores questionaram se realmente é possível pressupor esta disjunção dos atributos em dados reais e, para responder a essa questão, realizaram experimento comparando o *Co-Training* com o EM [DLR77], escolhido em vista de se dispor de

[2]Considerando-se o total de documentos em ambos os *corpora*, *20newsgroups* e *Reuters 21578*.
[3]Obedecendo ao mesmo critério utilizado no algoritmo proposto. Mas, sem realizar o particionamento.
[4]Conjuntos cuja intersecção é vazia

experimentos bem-sucedidos com classificação de textos rotulados e não rotulados e por entenderem haver simplicidade de utilização com o classificador *Naïve Bayes*.

Para o experimento com o EM, utilizaram, inicialmente, o *Naïve Bayes* para gerar os parâmetros do EM, usando, apenas, os dados rotulados. Durante o *expectation step* é calculada a probabilidade $P(C_j|d_i)$ de ocorrência da classe C_j dado o documento não rotulado d_i. O *maximization step* estima os novos parâmetros para o classificador.

Para o *Co-Training*, treinou-se dois classificadores *Naïve Bayes* com porções distintas dos atributos. Para tanto, faz-se uma iteração que inicia treinando os dois classificadores com o conjunto de dados rotulados e, para cada classe C_i, cada classificador rotula o documento com a maior confiança de que pertença à classe C_i. Espera-se, assim, que, no retreino, o novo documento rotulado pelo primeiro classificador forneça melhores dados de treino para o segundo classificador e, da mesma forma, o novo documento rotulado pelo segundo classificador forneça melhores dados de treino para o primeiro classificador.

Para os experimentos, foram utilizados dois *corpora*:

1. ***WebKB Course***[5]: uma coleção de documentos *web* dos departamentos de Ciência da Computação de 4 universidades[6]. O objetivo do experimento era descobrir quais documentos são páginas iniciais de cursos acadêmicos. Foram separados 25% dos documentos para a fase de teste. O classificador *Naïve Bayes* foi executado independentemente tanto com 100% dos documentos rotulados quanto com apenas 12, para que se pudesse aferir o ganho proporcionado pelos dois algoritmos. O EM e o *Co-Training* foram executados com 12 documentos rotulados. O *Co-Training* apresentou performance inferior ao EM. Os autores elencam algumas hipóteses para tanto:

   (a) A identificação de páginas iniciais pretendida era muito simples e, assim, o EM pôde ter uma boa performance;

   (b) A disjunção dos atributos do *corpus WebKB* não era tão independente quanto se pressupunha;

   (c) O *Co-Training* não é capaz de se beneficiar da disjunção dos atributos a ponto de superar o EM.

2. ***News 2x2***: Para garantir a disjunção dos dados os autores organizaram um *corpus* baseado no *20Newsgroups* com documentos dos grupos *comp.os.ms-windows.misc* e *talk.politics.misc* para compor o conjunto dos documentos rotulados como positivos e *comp.sys.ibm.pc.hardware* e *talk.politics.guns* para os documentos negativos. Assim, os vetores dos documentos eram dados por $\{c_1, c_2, ..., p_1, p_2, ...\}$, onde c_i é um atributo extraído de um documento *comp.** e p_i é um atributo extraído de um documento

[5] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/webkb-data.gtar.gz

[6] Universidades de Cornell, do Texas, de Washington e do Wisconsin.

*talk.politics.**. Foram removidas as *stopwords* mas não houve *stemmização* ou qualquer normalização das palavras. Foram, então, selecionados 4.000 atributos usando informação mútua. Foi realizada normalização do tamanho dos documentos. Três documentos por classe receberam rótulo, mil ficaram sem rótulo e 976 foram separados para teste. O *Co-Training* apresentou performance superior.

Da análise destes resultados, os autores consideraram ainda não estar provado que a disjunção dos atributos traga benefício para o aprendizado com dados rotulados e não rotulados. Restava, ainda, determinar se a boa performance se deve realmente à disjunção dos atributos ou se ao fato de que o *Co-Training* realiza um aprendizado incremental, incluindo um documento por classe a cada iteração, enquanto o EM trata todos os documentos a cada iteração. Assim, foi realizado novo experimento que demonstrou a efetividade de se tratar a disjunção dos atributos. Neste experimento, o EM e o *Co-Training* foram hibridizados, resultando no Co-EM e *self-training*, com as seguintes características:

1. **Co-EM**: Realiza as iterações rotulando todos os documentos. Mas, tratando separadamente os atributos disjuntos;

2. **Self-Training**: Manteve o aprendizado incremental, rotulando um documento por classe a cada iteração. Mas, usa apenas um classificador que atua sobre a totalidade do atributos.

Tendo, com o novo experimento, provado o benefício de se explorar a disjunção dos atributos, os autores buscaram descobrir se seria possível obter benefícios semelhantes em conjuntos de dados que não apresentem tal disjunção, ao menos não de forma conhecida. Para tanto, foi realizado novo experimento, ainda utilizando o *corpus News 2x2*. Porém, a divisão foi realizada de maneira aleatória. Os autores obtiveram resultados inferiores aos aferidos quando se conhecia a exata disjunção de atributos, mas superiores em relação ao tratamento dos atributos de forma indistinta. No entanto, ao se utilizar o *corpus News5* que contém os documentos dos grupos *comp.** do *corpus 20Newsgroups*, verificou-se que, apesar de não haver uma disjunção natural nos atributos, o *Co-Training* ainda teve uma taxa de erro 10% inferior ao EM.

Os autores ressaltam que o ganho ocorreu ao se realizar uma separação randômica de atributos e pretendem elaborar um algoritmo de separação baseado em informação mútua.

Cabe ressaltar que, neste trabalho, Nigam e Ghani [NG00] não propõem categorização auxiliada por *clustering*, objeto do presente estudo. No entanto, considerou-se relevante revisar este experimento tendo em vista que sua validação ocorreu por comparação com processo de categorização bayesiana auxiliada por *clustering* usando o algoritmo EM e que os resultados demonstraram que o *Co-Training* obteve performance superior.

3.3 Trabalhos Baseados em Classificadores SVM ou Derivados do SVM

Nesta seção estão agrupados os trabalhos que utilizam classificador SVM ou dele derivado. Como na seção anterior, o trabalho da Subseção 3.3.3 também não apresenta experimentos com textos em linguagem natural sem formatação específica. Da mesma maneira, sua inclusão neste estudo considera as metodologias propostas que podem ser utilizadas com documentos textuais.

3.3.1 Combinando *Clustering* e *Co-Training* para Melhorar a Classificação de Textos Usando Dados Não Rotulados

Raskutti, Ferrá e Kowalczyk [RFK02a], buscam solucionar uma limitação do *Co-Training*, que pressupõe que os atributos dos dados possam ser divididos em 2 grupos distintos, cada qual usado no treinamento de um classificador diferente. Eles apresentam uma proposta em que os atributos derivados do pré-processamento dos textos constituem o grupo usado no treinamento do primeiro classificador, denominado classificador WP e, para treinar o segundo classificador, denominado CF, propõem o uso de *Clustering* dos documentos para gerar novos atributos, contendo informações tais como medidas de similaridade.

O processo de *clustering* empregado pelos autores tem complexidade $O(r^2)$, para r sendo a quantidade de amostras de textos rotulados e não rotulados usadas no treinamento. Para evitar que o tempo de treino escale a níveis que exijam demasiados recursos computacionais, os autores dividem as amostras em S partições antes de executar o *clustering*. De cada partição, são selecionados, apenas, os N maiores *clusters*. Cada *cluster* C_i gera os seguintes novos atributos para os documentos:

1. Uma *flag* indicando se este é o *cluster* mais próximo do documento;
2. A similaridade com o centróide de C_i;
3. A similaridade com o centróide dos documentos não rotulados de C_i;
4. Para cada classe q, a similaridade com o centróide dos documentos de q presentes em C_i.

Assim, a quantidade de novos atributos é de $SN(q+3)$.
Os classificadores usados no *Co-Training* utilizam o algoritmo SVM da seguinte maneira:

1. Treinam-se os classificadores WP e CF;
2. Usa-se o CF para rotular o conjunto de treino e seleciona-se alguns destes documentos para integrar o novo conjunto de treino do WP;

3. Usa-se o WP para rotular o conjunto de treino e selecionam-se alguns destes documentos para integrar o novo conjunto de treino do CF;

4. Treinam-se novamente os classificadores WP e CF, agora denominados WP_{co} e CF_{co}.

Originalmente, os autores propuseram iterar os passos 2 a 4 acima. No entanto, após realizarem os primeiros testes, concluíram que os melhores resultados eram decorrentes de uma única etapa de *co-training* e, assim, descartaram tal iteração.

Para os experimentos, os autores dividiram o conjunto de treino em 5 partições e, de cada partição, selecionaram os 20 maiores *clusters* para a geração de novos atributos. Assim, $S = 5$ e $N = 20$.

Os autores utilizaram os seguintes *corpora*:

1. **WebKB**: as 4 categorias mais populosas, excluindo a categoria *others* e as páginas de redirecionamento de navegador, totalizando 4.108 páginas. Eles selecionaram, randomicamente, 225 documentos para o treino e 800 para o teste. Os demais documentos constituíram o conjunto de dados não rotulados. Após o pré-processamento, os autores obtiveram 87.601 atributos derivados das palavras extraídas dos documentos e 700 gerados pelo *clustering*.

2. **Reuters 21578**[7]: o *modApte split*, com 9.603 documentos de treino e 3.299 de teste. Os autores selecionaram as 10 categorias mais populosas e as dividiram em conjuntos rotulados e não rotulados. Embora tenham dividido as categorias em diferentes proporções, não foram informados o critério de divisão nem o tamanho de cada conjunto. Após o pré-processamento, os autores obtiveram 20.197 atributos derivados das palavras extraídas dos documentos e 1.300 gerados pelo *clustering*.

3. **20Newsgroups**: mensagens de 20 *newsgroups*, totalizando 18.828 documentos, sem os documentos redundantes, divididos homogeneamente entre os grupos [Lan95]. A maior parte dos cabeçalhos foi removida. Não foi informado quais cabeçalhos permaneceram. Os autores selecionaram, randomicamente, 2.000 documentos para o conjunto de dados rotulados de treino e 8.000 para o conjunto de dados não rotulados. O restante foi utilizado para teste. Após o pré-processamento, os autores obtiveram 26.362 atributos derivados das palavras extraídas dos documentos e 2.300 gerados pelo *clustering*.

Os autores reportam utilizar pré-processamento compatível com o relatado em [NMT+00], [Joa99] e [RFK02b]. Não se obteve acesso a este último artigo para conferência destes dados. Quanto aos dois primeiros, há concordância em relação ao pré-processamento realizado sobre o *WebKB*, que não passa por *stemming* ou remoção de *stopwords*. Porém, quanto ao *Reuters 21578*, [NMT+00] removem *stopwords* mas não fazem *stemming*. Já

[7] http://kdd.ics.uci.edu/databases/reuters21578/README.txt

[Joa99] faz o *stemming* e a remoção de *stopwords*. Da mesma maneira, quanto ao *20Newsgroups*, [NMT+00], removem *stopwords* mas não fazem *stemming*. [Joa99] não utiliza o *corpus* 20Newsgroups.

Neste experimento, os autores utilizaram a medida *micro-averaged breakeven point* μBP [MS00], comumentemente reportada em virtude de se obter diferentes *breakeven points* para cada categoria. No entanto, fazem ressalvas quanto ao uso desta medida tendo em vista que detalhes importantes podem ser suprimidos. Por esta razão, apresentaram, também, os *breakeven points* por categoria e a taxa de erros de classificação.

Os algoritmos comparados foram os utilizados no *Co-Training*, todos baseados em SVMs com *kernel* linear. Dois deles foram treinados com os atributos extraídos das ocorrências das palavras, com ou sem *Co-Training*, WP_{co} e WP, respectivamente. Os outros dois foram treinados com os atributos gerados pelo *clustering* dos documentos, com ou sem *Co-Training*, CF_{co} e CF, respectivamente.

Após os experimentos, os autores verificaram que o WP_{co} apresentou a melhor performance na maioria dos experimentos. Assim, consideraram este seu classificador final. O CF_{co} raramente apresentou melhor performance que o WP_{co} e, na maioria das vezes em que apresentou boa performance, o WP_{co} apresentou performance superior, demonstrando-se bastante sensível à influência do *Co-Training*. Os autores ainda fazem algumas considerações acerca do pequeno ganho apresentado pelo CF_{co} em função da baixa qualidade dos atributos gerados pelo *clustering* como, por exemplo, atributos binários. No entanto, percebe-se que estes atributos trazem um ganho significativo para o classificador WP_{co}. Este ganho, porém, ocorre somente na primeira iteração do *Co-Training* [RFK02a]. Analisando a natureza dos atributos gerados pelo *clustering*, verifica-se que, exceto pelos dois primeiros atributos de cada *cluster*, os demais, um para cada classe e um para os não rotulados, são dependentes de centróides das respectivas classes (ou dos documentos não rotulados). Ora, após a primeira iteração do *Co-Training*, alguns documentos são rotulados e, portanto, mudam os centróides citados. Não há menção a novas execuções do passo de *clustering*, nem a qualquer recálculo de distância de centróides. Desta maneira, supõe-se que não há alteração dos valores dos atributos utilizados no treinamento do CF_{co}, e, conseqüentemente, estes atributos tornam-se cada vez menos representativos comprometendo, assim, a performance do CF_{co}.

3.3.2 CBC: Classificação de Texto Baseada em *Clustering* Requerendo Mínimos Dados Rotulados

Zeng *et al.* [ZWC+03] propõem classificação de dados não rotulados utilizando algoritmo de *clustering* guiado por um pequeno conjunto de dados rotulados. Os dados não rotulados são, então, rotulados de acordo com o *cluster* ao qual foram associados. Dispondo, então, de um conjunto maior de dados rotulados, realiza-se o treinamento do classificador.

Os autores ressaltam que, embora a técnica descrita acima não seja uma proposta nova, o método por eles proposto objetiva solucionar dificuldades de classificação quando a quantidade de dados rotulados é extremamente pequena, como, por exemplo, menos de 10 amostras para cada rótulo. Enquanto outras propostas enfocam na classificação auxiliada por dados não rotulados, o enfoque, aqui, é no *clustering* auxiliado por dados rotulados.

Conforme os autores, os métodos de *clustering* são menos sensíveis a tendências causadas por dados esparsos iniciais que os de categorização. Além disto, o método de *clustering* proposto é, de fato, um classificador baseado numa distribuição de probabilidade e, assim, conforme demonstrado por Ng e Jordan [JN02], atinge sua performance assintótica mais rapidamente que os modelos discriminativos. Para a fase de *clustering* foi utilizada a versão *soft-constraint* do *K-Means* e a quantidade de grupos é determinada pelo número de classes existentes nos dados rotulados. Para a fase de categorização foi utilizado o *Transductive SVM*, TSVM [Joa99].

Os algoritmos de *clustering* e categorização são sucessivamente invocados através de sucessivas iterações auxiliando-se mutuamente no ato de rotular os dados:

1. No passo de *clustering*, calculam-se os centróides de cada classe considerando-se somente os dados rotulados. Estes centróides são usados como semente inicial do *K-Means*. Após a convergência do *K-Means*, somente um percentual p dos documentos não rotulados mais próximos dos centróides recebem o rótulo atribuído ao respectivo centróide. Os demais documentos permanecem sem rótulo.

2. No passo de categorização, realiza-se o treino do TSVM com todos os dados (rotulados e não rotulados) e, de cada classe, seleciona-se o mesmo percentual p de documentos não rotulados com a maior margem e aplica-se o rótulo da respectiva classe.

Esta iteração entre os algoritmos repete-se até que não restem documentos sem rótulo.

Os autores realizaram 3 experimentos com os seguintes *corpora*: 20Newsgroups, Reuters-21578 e páginas *web* do *Open Directory Project* (ODP). Para o *Cluster Based Categorization*, CBC, foram extraídos, de cada documento, um vetor de atributos para o título e outro para o corpo do documento. Para os demais algoritmos, foi extraído apenas um vetor de atributos para o corpo e o título de cada documento. Foi feita a *stemmização* das palavras e a remoção das *stopwords* e das palavras que ocorreram em no máximo 3 documentos. Também foram removidas as palavras que ocorriam apenas no conjunto de teste, mas não no conjunto de treino. Os atributos dos vetores receberam o *TF-IDF* das palavras restantes.

Foi necessário reduzir o número de classes em cada *corpus* devido ao tempo de treino do TSVM escalar em função da quantidade de classes. Assim, foram utilizadas:

1. **20Newsgroups**: as mesmas 5 classes comp.* utilizadas em Nigam e Ghani [NG00], contando, cada uma, com, aproximadamente, 1.000 documentos, 80% na fase de treino e 20% na fase de teste. Após o pré-processamento, restaram 14.171 palavras distintas, 14.059 no corpo dos documentos e 2.307 no título;

2. **Reuters-21578**: as 10 maiores classes, *earn, acq, money-fx, grain, crude, trade, interest, ship, wheat* e *corn*, particionadas de acordo com o *ModApte*, havendo 6.649 documentos de treino e 2.545 de teste. Após o pré-processamento, restaram 7.771 palavras distintas, 7.065 no corpo dos documentos e 6.947 no título;

3. **Open Directory Project**: as 6 maiores classes do segundo nível do diretório, *Business/Management* (858 documentos), *Computers/Software* (2.411), *Shopping/Crafts* (877), *Shopping/Home & Garden* (1.170), *Society/Religion & Spirituality* (886) e *Society/Holidays* (881). Foram utilizados 50% dos documentos para o treino e 50% para o teste. Após o pré-processamento, restaram 17.050 palavras distintas, 16.818 no corpo dos documentos e 3.729 no título;

Os autores utilizaram o pacote SVM-Light[8] para as categorizações SVM e TSVM, com *kernel* linear. O percentual de documentos não rotulados eleitos para recepção de rótulos tanto no passo de *clustering* quanto no de categorização do CBC foi de 1%. Como a classificação envolvia múltiplas classes, foi necessário treinar vários categorizadores SVM um-contra-todos.

A métrica de avaliação foi *micro-averaging* F1, que se constitui numa média ponderada de cada *F-Measure*, com $\beta = 1$. Em todos os *corpora* o CBC apresentou performance significativamente superior quando o conjunto de dados rotulados era pequeno, perdendo esta diferença e tornando-se aproximadamente equivalente ao SVM, TSVM e Co-Training à medida que aumentou o conjunto inicial de documentos rotulados.

Também foi avaliado o impacto de diferentes percentuais de seleção de documentos para aplicação de rótulo ao final das fases de *clustering* e categorização. Os autores perceberam que o percentual de 100%, ou seja, apenas uma iteração de *clustering* e categorização, foi claramente superior a percentuais inferiores. Pretendem estudar as razões de tal comportamento, que acreditam ocorrer em função do categorizador não conseguir acrescentar documentos que contribuam com informação significativa ao algoritmo de *clustering*.

3.3.3 *Support Cluster Machine*

Li, Chi, Fan e Xue [LCF+07] propõem o algoritmo *Support Cluster Machine*. Os autores argumentam que o SVM sofre de problemas de escalabilidade e acrescentam uma fase de pré-processamento ao SVM buscando reduzir a quantidade de instâncias usada para

[8] http://svmlight.joachims.org/

treinar o classificador. Diferentemente de outras propostas, o objetivo do *clustering* não é a seleção das instâncias mais representativas para treinamento do classificador. O classificador é treinado com os centróides obtidos pelo *clustering*. Este modelo parte do pressuposto que os dados seguem uma distribuição estatística e, assim, a escolha dos centróides para o treinamento é mais adequada que a de instâncias representativas por preservar com maior exatidão o perfil estatístico do conjunto de dados original. Desta forma, os autores reportam que obtiveram acurácia equivalente ao SVM com significativa redução de custo computacional.

Os autores ainda relatam que a mesma função de medida de distância usada para comparar os *clusters* entre si na fase de *clustering* é, também, usada na categorização, comparando documentos com os *clusters*. Não definem um algoritmo de *clustering* específico a ser empregado no pré-processamento: dentre as sugestões apresentadas encontram-se o K-Means, conforme implementado em Hartigan e Won [HW79], EM [DLR77] e *clustering hierárquico* [ZRL96].

Nos seus experimentos, os autores optaram pelo algoritmo Threshold Order Dependent, TOD [FK99] *apud* [LCF+07]. Este algoritmo, para cada documento, verifica se este está a uma distância acima de um valor de corte do ponto mais próximo. Se estiver, um novo *cluster* é formado com o documento como centro. Senão, associa-o ao cluster mais próximo. Os autores optaram por usar este algoritmo por sua complexidade linear e por ser capaz de lidar com dados seqüenciais com uma complexidade espacial insignificante. Para fins de comparação, o EM também foi utilizado.

Os conjuntos de dados utilizados pelos autores foram: 1) *Toydata*, gerado randomicamente pelos autores, 2) um banco de dados com imagens de números escritos a mão, divididos em 10 classes, obtido do MNIST[9] e 3) o banco de dados *Adult*[10], com informações sobre renda, construído a partir de dados de censo.

3.3.4 Classificação SVM Hierárquica Baseada em *Support Vector Clustering* e sua Aplicação na Categorização de Documentos

Hao, Chiang e Tu [HCT07] relatam experimento de categorização de documentos usando uma hierarquia de classes obtida através de *clustering*.

Os autores optaram por utilizar SVM, *Support Vector Machine* [CV95], como algoritmo de classificação por entenderem tratar-se do estado da arte na categorização de documentos. No entanto, preocupando-se com a dificuldade de se obter um bom classificador capaz de dintingüir entre múltiplas classes em face da facilidade de se obter um que trate apenas duas classes, decidiram gerar múltiplos classificadores binários.

Duas estratégias podem ser utilizadas para se reconhecer diversas classes usando

[9] http://yann.lecun.com/exdb/mnist/
[10] http://archive.ics.uci.edu/ml/

múltiplos classificadores binários:

1. A **estratégia um-contra-todos** consiste em gerar um classificador binário para cada classe, ou seja, um classificador que decide se o documento pertence a uma determinada classe ou não;

2. A **estratégia um-contra-um** consiste em gerar um classificador binário para cada **par** de classes. Desta forma, para decidir se um documento pertence à classe x_i, é necessário submetê-lo a cada um dos classificadores $C(x_i, x_j)$, onde $j = 1..n$ e $j \neq i$ e n é o número de classes.

Desta maneira, encontraram um novo problema: a escalabilidade. Além da grande quantidade de atributos típicos da categorização de documentos, ter de lidar com uma grande quantidade de classes eleva o custo computacional a patamares proibitivos. Para lidar com esta questão, adotaram um modelo hierárquico de classes. Conforme os autores, um dado documento poderá ser classificado em uma classe folha ou em uma classe mais genérica, representada por um nodo intermediário da árvore.

O processo de classificação consiste, então, em realizar categorizações *flat* iniciando na raiz da árvore de classes, descendo recursivamente através de um ou mais ramos. A cada nível, a categorização ocorre usando um dos seguintes métodos:

1. Se o nodo se dividir em dois ramos, é utilizada uma classificação binária SVM;

2. Se o nodo possuir mais de dois ramos, são utilizados múltiplos classificadores. A decisão entre estratégia um-contra-todos ou um-contra-um tem como base a acurácia obtida em cada estratégia no nodo corrente.

Devido à dificuldade de se obter documentos pré-rotulados por especialista humano para as fases de treino e teste, os autores utilizaram SVC, *Support Vector Clustering* [BHHS+02], para gerar a hierarquia automaticamente. Inicialmente, cria-se a raiz representando todos os documentos em um único grupo e vai se subdividindo os grupos através da variação de parâmetros do SVC[11], gerando os diversos níveis da hierarquia de classes. Para decidir quando parar a divisão dos grupos, os autores utilizaram a medida CS [XB91] para *fuzzy clustering*, que leva em conta tanto o grau de compactação dos documentos de cada grupo, quanto o grau de separação dos grupos entre si.

O pré-processamento dos documentos compreendeu a remoção de *stopwords* e palavras com menos de quatro ocorrências no *corpus*. Ainda assim, as instâncias contavam com, aproximadamente, 10.000 atributos. Desta maneira, os autores reduziram a dimensionalidade utilizando ganho de informação [YP97], *Learning Vector Quantization* [SK99] e *Latent Semantic Indexing* [Ben73].

[11] O limiar e o tratamento de ruído (*outliers*).

Foi utilizado o *corpus Reuters-21578*[12]. O treino utilizou 9.603 documentos e o teste utilizou 3.299 documentos. Foram eliminadas classes que não continham documentos para treino ou teste e documentos que não estavam ligados a nenhuma classe. O conjunto resultante tinha 90 categorias. Desta maneira, os autores realizaram seus experimentos com o mesmo conjunto de documentos utilizados nos experimentos descritos por Joachims [Joa98].

Os autores mediram a performance usando *F-Measure* com $\beta = 1$. Selecionaram as 10 categorias mais freqüentes e compararam os resultados obtidos com *Decision Tree C4.5*, KNN e SVM não hierárquico. O método proposto obteve melhor resultado em 6 categorias. Verificaram que nas 4 categorias em que não alcançou a melhor performance, possuíam grande número de subcategorias e os seus grupos geradores não tinham boa separação, sendo comum que se intercalassem.

3.3.5 Mineração de Textos de Decisões da Suprema Corte Administrativa Austríaca

Feinerer e Hornik [FH08], relatam experimento de *clustering* e classificação de documentos contendo jurisprudência da Suprema Corte Administrativa Austríaca, no subdomínio do Direito Tributário, no período de 2.000 a 2.004, tendo em vista a importância de seus efeitos no setor comercial. O objetivo deste estudo foi comparar os agrupamentos formados com estudos anteriores, da década de 1.980, acerca de jurisprudência no mesmo subdomínio, a fim de averiguar os efeitos das mudanças sociais do corpo normativo tributário Austríaco.

Os autores usaram um *corpus* de treino composto de 994 documentos textuais, cada um contendo uma decisão da corte em língua Alemã. As palavras foram *stemmizadas* e receberam 2 medidas de peso: TF e TF-IDF [LSZ04]. Especialistas do domínio realizaram, manualmente, a divisão dos textos em 3 grupos, classificando-os com os rótulos “VA Tax”, “Income Tax” e “outros”.

No primeiro experimento, os autores usaram o algoritmo *K-means*. No entanto, embora os testes tenham durado poucos minutos em seus equipamentos, os autores afirmam estarem cientes de que a escalabilidade dos dados demandaria demasiados recursos computacionais. Assim, um novo experimento de agrupamento foi realizado. Desta vez, cada cluster foi manualmente configurado par ter um conjunto específico de palavras-chave do subdomínio. Assim, cada documento foi analisado tendo em vista a similaridade com o conjunto de palavras-chave. Este método foi denominado *Keyword Based Clustering Method*, ou Método de Agrupamento Baseado em Palavras-Chave.

Ambos experimentos foram avaliados por meio dos índices Rand [Ran71] e cRand [HA85]. O *Keyword Based Clustering Method* superou o *K-Means*, aumentando o índice Rand de 0,52 para 0,66 e o cRand de 0,03 para 0,32. O grupo “Income Tax” teve 100% de precisão e *Recall*.

[12] http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

Posteriormente, os autores realizaram experimentos de classificação de documentos de jurisprudência em 2 grupos: documentos que tratam de matéria fiscal Austríaca e documentos que não tratam desta matéria. Foi utilizada a classificação "C-SVC" com *Support Vector Machines*. Foram utilizados 200 documentos para o treino e 50 para o teste. O treino da SVM levou um dia e a classificação de cada documento foi calculada em 15 minutos num computador com processador de 2.6GHz e 2Gb de memória principal.

Assim, os autores decidiram utilizar a abordagem de matriz termo-documento, com pesos baseados em TF e TF-IDF. Novamente utilizaram 200 documentos para o treino e 50 para o teste. Obtiveram índices Rand em torno de 0.6 e cRand em 0.2, tidos como altamente promissores pelos autores e indicativos de que o uso de SVM para a classificação de textos tem um grande potencial.

Neste trabalho, Feinerer e Hornik [FH08] não propõem novos algoritmos. Apenas o uso de algoritmos já conhecidos. Além disto, o processo de classificação após o *clustering* apenas determinava se o documento classificado fazia parte do domínio ou não. Tem-se como maior contribuição o uso do *Keyword Based Clustering Method* e, uma vez que, se comparado à situação brasileira, estão disponíveis, entre nós, o Vocabulário Controlado Básico, publicado pelo Senado Federal Brasileiro [JAS+07] e o Tesauro Jurídico da Justiça Federal Brasileira [SMS+07]. Pretende-se utilizá-los na fase de pré-processamento dos exemplos de uso propostos em nosso trabalho, tal como descrito no Capítulo 4.

3.3.6 Aprendizagem Ativa Usando Pré-*Clustering*

Nguyen e Smeulders [NS04] propõem a utilização de *clustering* para auxiliar algoritmo de *Active Learning*, proposto por Lewis e Gale [LG94]. O algoritmo original baseia-se em treinar um classificador com um conjunto de dados rotulados iniciais e, então, realizar iterações executando o classificador sobre dados não rotulados para selecionar os n documentos que o classificador tenha a menor certeza de qual rótulo aplicar. Estes documentos são, então, rotulados por especialista humano. O classificador realiza novo treino incluindo os novos documentos rotulados. Esta iteração se repete enquanto o especialista humano estiver disposto a realizar classificações. Nesta proposta, os autores usam um algoritmo de *clustering* para 1) realizar a seleção de documentos a rotular e 2) rotular os documentos sem a intervenção humana.

Para tanto, o algoritmo de *soft clustering* agrupa os documentos rotulados e não rotulados, selecionando, primeiramente, os documentos mais representativos dos *clusters* para rotular e aplica o rótulo do documento rotulado mais próximo no mesmo *cluster*. A partir de então, o algoritmo passa por cada *cluster*, iniciando pelos mais densos, selecionando dois tipo de amostras: 1) os documentos não rotulados mais representativos do *cluster* e 2) os documentos não rotulados mais próximos dos limites entre *clusters*. Os documentos que pertencem a um único *cluster* recebem o rótulo do documento rotulado mais próximo

e os que pertencem a mais de um *cluster* são atribuídos ao *cluster* de maior probabilidade e, então, recebem rótulo do documento rotulado mais próximo que esteja neste *cluster*. Quando a margem de classificação atinge a borda dos *clusters*, é executado um novo reagrupamento com um valor de limiar menor, a fim de obter mais *clusters* de menor tamanho, refinando, assim, a qualidade da classificação.

Foram realizados dois experimentos: o primeiro buscou detectar imagens que contivessem rostos humanos. Os autores utilizaram 2.500 imagens com tamanho 20×20, obtidas conforme experimentos referidos em artigo anterior de Pham, Worring e Smeulders [PWS01]. No entanto, destaca-se que o referido artigo relata a construção de um banco de dados contendo 33.360 faces e 360.000 padrões não faciais. O processo pelo qual os autores selecionaram 2.500 imagens deste conjunto anterior não foi informado. O segundo buscou identificar números escritos a mão, separando imagens de um determinado dígito das demais. As imagens foram obtidas do banco de dados MNIST[13].

Para comparações, foram implementados três outros algoritmos de *Active Learning*, todos usando SVM linear para classificação. O primeiro seleciona os dados de treino randomicamente. O segundo seleciona as instâncias mais próximas da borda de classificação. O terceiro usa os medóides dos *clusters* próximos à margem do SVM.

3.4 Usando Supervisão Parcial para Categorização de Textos

Aggarwal, Gates e Yu [AGY04] propõem método de categorização utilizando classes definidas por algoritmo de *clustering* parcialmente supervisionado. No experimento realizado, os autores usaram a taxonomia do Yahoo[14], para gerar as sementes iniciais para o algoritmo de *clustering*. O algoritmo de *clustering* proposto emprega junção de *clusters* cujos centróides sejam muito próximos, além de remoção de *clusters* com pequena quantidade de documentos. Por esta razão, após o *clustering*, as classes geradas divergem da taxonomia do Yahoo, embora mantenham coerência com estas, segundo avaliação humana realizada.

Uma vez obtidas as classes, a categorização de novos documentos é feita por algoritmo também proposto pelos autores, que emprega a mesma medida de distância utilizada no *clustering*. Por esta razão, os autores afirmam que a categorização pode, teoricamente, obter acurácia perfeita e, portanto, a qualidade da categorização passa a depender exclusivamente da qualidade do *clustering*.

De acordo com os autores, o *clustering* sem qualquer tipo de supervisão é capaz de gerar grupos de boa qualidade somente quando há uma pequena quantidade de grupos, aproximadamente 50. O experimento realizado gerou 1.167 *clusters*.

[13] http://yann.lecun.com/exdb/minist/
[14] http://www.yahoo.com

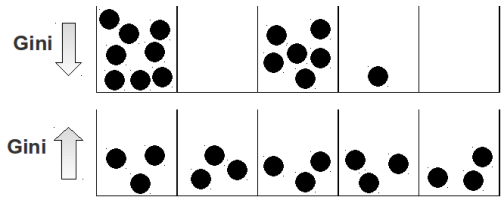


Figura 3.1 – Comportamento do Índice Normalizado Gini

Ambos os algoritmos utilizam o modelo de espaço vetorial para representação dos documentos e a distância de cosseno como medida de similaridade.

A fase de pré-processamento utiliza o cálculo do Índice Normalizado Gini [PG94] para o descarte de atributos. Para tanto, calcula-se a presença fracional de uma dada palavra numa classe i, dada por f_i/n_i, onde f_i é a freqüência da palavra na classe i e n_i a contagem de palavras na classe i, o desvio fracional p_i é definido por

$$p_i = \frac{f_i/n_i}{\sum_{j=1}^{K} f_j/n_j} \tag{3.2}$$

onde K é o número de classes.

O Índice Normalizado Gini normalizado de uma dada palavra é dado por:

$$g = 1 - \sqrt{\sum_{i=1}^{K} p_i^2} \tag{3.3}$$

Assim, conforme inlustrado na Figura 3.1, à medida que se equilibra a distribuição de uma palavra através de diferentes classes, o Índice Normalizado Gini se aproxima de $1 - 1/\sqrt{K}$. Por outro lado, conforme a palavra se demonstrar muito particular de uma dada classe, o Índice Normalizado Gini decresce significativamente.

O algoritmo de *clustering* parte de um conjunto inicial de centróides e realiza sucessivas iterações, divididas em 4 fases:

1. **Atribuição de Documentos**: cada documento é associado ao *cluster* cujo centróide esteja mais próximo. Documentos cuja distância até o centróide mais próximo esteja acima de um valor de corte são descartados como ruído. Ao final da fase é calculado um novo centróide para cada grupo;

2. **Seleção de Atributos**: As palavras com o menor peso na definição dos centróides são descartadas. Este descarte deve ocorrer gradualmente, a cada iteração, e não numa única vez para que não ocorra a perda de atributos importantes em função de centróides ainda não muito bem refinados;

3. **Aglomeração**: *Clusters* cujos centróides estejam muito próximos são agrupados;

4. **Eliminação**: *Clusters* que apresentem um conjunto muito pequeno de documentos

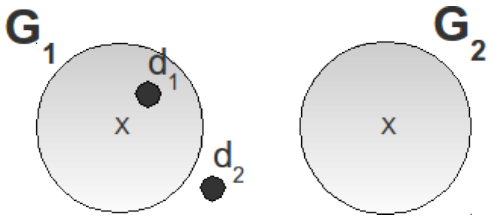


Figura 3.2 – Classificação de Documentos: Determinação da Dominância do Grupo

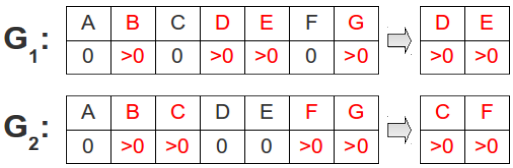


Figura 3.3 – Seleção de atributos para Determinação de Dominância quando os Documentos estão em Região *Intercluster*

são descartados. Seus documentos são atribuídos a outros *clusters* ou descartados como ruído conforme a sua distância aos demais centróides.

A condição de parada das iterações é baseada na quantidade de atributos. Ao atingir um valor de corte mínimo, as demais fases da iteração corrente são executadas e, então, encerram-se as iterações.

O algoritmo de categorização, a exemplo do algoritmo de *clustering*, poderia simplesmente classificar os novos documentos buscando o centróide mais próximo, utilizando a mesma função de similaridade. No entanto, os autores ressaltam que há a possibilidade de que um documento seja mal classificado quando houver um documento muito próximo a diferentes *clusters*. Para distingüir entre assuntos muito próximos, os autores empregam um método proposto originalmente por Chakrabarti *et al.* [CDA+98], adaptado a um modelo não-hierárquico.

O algoritmo de categorização seleciona os k *clusters* com centróides mais próximos e seleciona o *cluster* de maior dominação. Assim, sejam, por exemplo, dois grupos G_1 e G_2, dois documentos $d1$ e $d2$ a classificar, conforme a Figura 3.2, um valor de limiar l e uma função de similaridade sim(G_i,d_j). Se a similaridade $sim(G_1,d_1) > (sim(G_2,d_1)+l)$, ou seja d_1 está muito mais próximo de G_1 que de G_2, G_1 é dominante em relação a d_1, que é classificado como pertencente à classe representada por G_1. No entanto, embora $sim(G_1,d_2) > sim(G_2,d_2)$, $sim(G_1,d_2) <= (sim(G_2,d_2)+l)$, ou seja, d_2 encontra-se em uma região *intercluster*. Neste caso, a dominância é definida através de uma segunda função de similaridade $simdif(G_i - G_j, d_k)$, que calcula a similaridade desconsiderando, para tanto, os atributos não nulos de G_j, ou seja, descartam-se de G_i, todos os atributos zerados em seu centróide e todos os atributos diferentes de zero em G_j. Por exemplo, supondo que os vetores de atributos sejam compostos pelos atributos “A” a “G”, conforme apresentado na Figura 3.3, a função $simdif(G_1 - G_2, d_2)$ irá calcular a distância de d_2 ao centróide de G_1

considerando, somente, os atributos “D” e “E”. Já a função $simdif(G_2 - G_1, d_2)$ irá calcular a distância de d_2 ao centróide de G_2 considerando, somente, os atributos “C” e “F”.

No experimento realizado, os autores tentaram, primeiramente, realizar o *clustering* totalmente não supervisionado. No entanto, as classes obtidas eram muito genéricas fazendo a mistura de assuntos como “arte por computador”, “artesanato” e “museus”, resultando no assunto demasiadamente genérico "arte". Assim, decidiram pela utilização da taxonomia do Yahoo, na versão de novembro de 1.996, para gerar as sementes iniciais do algoritmo de *clustering*. Para tanto, a árvore de assuntos do Yahoo foi truncada, totalizando, então, 1.463 nodos folha. Através desta, obtiveram o *corpus* de, aproximadamente, 167 mil documentos.

Durante o pré-processamento dos documentos, foram utilizadas as seguintes reduções de dimensionalidade dos documentos:

1. descarte das palavras que ocorriam em, no máximo, 7 documentos, reduzindo de 700 mil palavras distintas no *corpus*, para 87 mil;
2. descarte das 10 mil palavras com o maior Índice Normalizado Gini, resultando em, aproximadamente, 77 mil palavras;

Os parâmetros utilizados para o *clustering* foram:

1. **Condição de Parada das Iterações**: redução da dimensionalidade para 200 palavras;
2. **Fator de Redução da Dimensionalidade**: 0,7;
3. **Limiar para Descarte de Grupo**: 8 documentos;
4. **Limiar para Agregar Grupos**: similaridade entre os centróides superior a 0,95.

No algoritmo de classificação, para detectar se C_1 é dominante em relação a C_2, a primeira condição testada verificava se a similaridade de um dado documento d tinha uma similaridade superior a de C_2 em relação a d em, no mínimo l. No experimento, foi utilizado um limiar l de 0,025.

Para avaliar a performance do processo, uma vez que as categorias não eram iguais às do Yahoo, não era possível usar estas classes para verificar a correta pertinência de documentos. Assim, procedeu-se a uma avaliação empírica que consistiu em separar uma amostra de 141 documentos dos *clusters* obtidos e entrevistar 10 pessoas que responderam, para cada documento, uma das 5 opções:

1. Categorização do Yahoo é melhor (8%);
2. Esta categorização é melhor (8%);
3. Ambas estão igualmente corretas (**78%**);
4. Nenhuma está correta (6%);

5. Não sabe (1%).

O trabalho de Aggarwal, Gates e Yu [AGY04] destaca-se não apenas por propor algoritmos de agrupamento e de categorização, mas, também, por diversas proposições, tais como o uso do Índice Normalizado Gini e avaliação do peso das palavras na definição dos centróides dos *clusters* na seleção de atributos e a função de similaridade do categorizador, que nem sempre decide pela mera verificação de distância simples, mas faz seleção de atributos quando um documento está próximo de mais de um *cluster*. Ressalte-se que esta função de similaridade é tida pelos autores como uma das contribuições do artigo. A outra contribuição, que determinou o título do artigo, é a conclusão dos autores que o *clustering* sem supervisão gera classes muito genéricas quando é grande o número de *clusters*. Verifica-se, no entanto, que o algoritmo de *clustering* proposto itera 4 fases e, na terceira fase, realiza a aglomeração de *clusters* que estejam muito próximos. Isto é um indicativo de que tal problema possa ser solucionado diminuindo-se o valor de limiar usado para definir a união de *clusters*.

Embora tenha apresentado bons resultados, não foi localizada, até o presente, uma continuidade para o mesmo. Foram encontradas 18 citações a este trabalho, mas nenhuma que aproveitasse as metodologias nele apresentadas. Alguns dos trabalhos da Seção 3.3 são mais recentes e, assim, constituem indicativo de que haja uma certa tendência atual em investigar o processo de categorização auxiliada por *clustering* usando categorizadores da "família" SVM. Uma das possíveis causas disto pode ter sido a escolha do *corpus* para os experimentos. O *corpus* usado pelos autores era composto de documentos obtidos através do Yahoo, diferentemente da maioria dos trabalhos aqui apresentados, que deram preferência a *corpora* mais comumentemente usados em pesquisas de aprendizado de máquina, tais como o *20Newsgroups* [Lan95], *Reuters21578*[15] e *WebKB Course*[16]. Desta maneira, torna-se mais difícil comparar diferentes propostas. Além disto, a forma como o método proposto foi validado também dificulta comparações: não foram utilizadas medidas de avaliação dos resultados tais como a medidas de coesão/separação de *clusters*, apenas avaliação humana sob critérios subjetivos tais como "esta classificação é melhor/pior/equivalente àquela classificação". O mesmo experimento, com os mesmos resultados, avaliado por outro grupo poderia obter, a nosso ver, avaliação bem distinta.

3.5 Considerações Finais

Foi revisada a literatura pertinente e, além de reunir-se conhecimento para empreender uma solução para problema nessa área, percebeu-se uma mudança de tendência no processo de categorização auxiliado por *clustering*, usando redes *bayesianas* e EM para categorizadores SVM ou derivados, não havendo preferência clara por qualquer algoritmo

[15] http://kdd.ics.uci.edu/databases/reuters21578/README.txt
[16] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/webkb-data.gtar.gz

de *clustering*. Detectou-se o trabalho da Seção 3.4, entitulado “Usando Supervisão Parcial para Categorização de Textos”, como alternativa isolada, indicativo, talvez, de possibilidades de aprofundamento de estudos. A Tabela 3.1 apresenta comparação dos trabalhos relacionados. A coluna “Processo Proposto” apresenta os algoritmos de agrupamento e categorização adotados pelos respectivos autores em suas proposições. A coluna “Validação” apresenta os algoritmos usados para comparar os resultados, validando seus respectivos processos. Toutanova *et al.* [TCP+01] e Hao, Chiang e Tu [HCT07], a compararam exclusivamente com experimentos com classificadores. Feinerer e Hornik [FH08] e Aggarwal, Gates e Yu [AGY04] não realizaram comparações, preferindo avaliação por especialista humano.

Quanto à redução de atributos, note-se que, apesar de a maioria dos trabalhos estudados usarem *stemming* para esta atividade, optou-se pela lematização, tendo em vista as perdas decorrentes de um processo de *stemming* em línguas mais ricas em inflexões que o inglês, conforme visto por Korenius [KLJ+04] em textos finlandeses e Gonzalez [Gon05] em textos em português.

Quanto ao objetivo de cada trabalho notamos que, embora haja um processo de *clustering* seguido de categorização, apenas os trabalhos de Feinerer e Hornik [FH08] e Aggarwal, Gates e Yu [AGY04] utilizam o clustering para descobrir as classes a serem utilizadas pelo categorizador. Hao, Chiang e Tu [HCT07] utilizam o clustering para descobrir a hierarquia das classes. A maioria dos trabalhos utiliza o clustering para melhorar a performance dos classificadores, mantendo o conjunto de classes original. São exemplos disto, os trabalhos de Zeng et al. [ZWC+03] e Nigam e Ghani [NG00] que buscam aumentar o conjunto de treino; Toutanova et al. [TCP+01], que buscam gerar os parâmetros bayesianos; Cong, Lee, Wu e Liu [CLW+04], que buscam garantir a relação um-para-um entre os grupos e as classes; e Raskutti, Ferrá e Kowalczyk [RFK02a], que buscam prover disjunção de atributos, requerida pelo Co-training.

Quanto à avaliação de suas propostas, novamente, Feinerer e Hornik [FH08] e Aggarwal, Gates e Yu [AGY04] se distingüem, realizando avaliação humana, em contraste com os demais que avaliam por comparação com estudos anteriores. Dois fato- res contribuem para tal ocorrência: a construção dos corpora utilizados e a proposta de geração de classes. Ao dispor de conjunto de documentos e classes distinto dos encontrados em outros estudos, ficam sem possibilidade de usá-los para comparação. Restam-lhes duas alternativas: realizar nova execução dos algoritmos a comparar ou avaliar através de especialista humano. Semelhante situação ocorre em nosso estudo. No entanto, devido às dificuldades de se conseguir realizar avaliação humana de uma grande quantidade de dados, optamos por reexecutar o algoritmo de Aggarwal, Gates e Yu [AGY04] utilizando os mesmos dados de nosso estudo, conforme veremos no Capítulo 3, para avaliar os resultados do clustering, restringindo a avaliação humana aos resultados da categorização.

Tabela 3.1 – Trabalhos Relacionados: Quadro Comparativo

Referência	Objetivo	Método	Processo Proposto		Validação	
			Clust.	Classif.	Clust.	Classif.
Toutanova [TCP+01]	Gerar parâmetros bayesianos	Parâmetros bayesianos = *hidden values* estimados e maximizados pelo EM.	EM	Naïve Bayes		Naïve Bayes, PLSA, HS, KNN, SVM
Nigam [NG00]	Aumentar o conjunto de treino.	Usa 2 classificadores, um para cada conjunto disjunto de atributos.	Co-Train.	Naïve Bayes	Co-EM, EM, Self-Train.	Naïve Bayes
Cong [CLW+04]	Prover relação 1x1 entre grupos e classes.	Classificadores podam árvore gerada por *clustering* hierárquico.	*hard clust.* hier. e EM	Naïve Bayes	EM, E-EM e M-EM	Naïve Bayes
Feinerer [FH08]	Descoberta de classes	Termos pré-definem centróides, descarta demais *tokens*.	K-Means KBC	C-SVC SVM	Aval. humana	
Hao [HCT07]	Geração de árvore de classificadores	Iterações do SVC dividem em grupos cada vez menores.	SVC	SVM		SVM, KNN
Zeng [ZWC+03]	Aumentar o conjunto de treino.	Centróides pré-definidos por taxonomia. Iterações rotulam *subset* dos documentos.	CBC	TSVM	Co-Train.	SVM, TSVM
Raskutti [RFK02a]	Provê disjunção de atribs.	Gera segundo grupo de atributos através de *clustering*	Flat 1-pass	SVM	Flat 1-pass	SVM
Aggarwal [AGY04]	Descoberta de classes	Centróides pré-definidos por taxonomia. Modifica a taxonomia. Descarta docs/grupos.	TClus	Assign	Aval. humana	

4. Classificação de Textos Jurídicos usando Classes Geradas por Agrupamento Parcialmente Supervisionado

4.1 Considerações Iniciais

Atualmente, o trabalho de pesquisa jurisprudencial realizado pelos Operadores do Direito demanda demasiado tempo em virtude das limitações das ferramentas de pesquisa disponíveis. Dentre as limitações encontradas neste sistema, destacamos:

1. **Escopo da Pesquisa**: A jurisprudência é composta de documentos cuja classificação é um preâmbulo, denominado ementa composto de uma seqüência de termos jurídicos e um resumo do tema abordado no texto. A Figura 4.1 apresenta uma visão geral da estrutura do texto jurisprudencial. As seções 1, 2, 3 e 6 do documento representam informação do caso específico, não do tema debatido no texto. A seção 4 é o *caput* da ementa, composto de seqüências de termos, simples ou compostos, separados por caracteres de ponto '.', destacados em vermelho. A seção 5 é o corpo da ementa, que apresenta um resumo dos temas abordados, sem termos específicos que os identifiquem. A seção 7 é o relatório dos fatos, seguida da cognição do juiz. Esta última seção é a que apresenta o conteúdo textual de interesse do usuário que realiza a pesquisa;

   Os sistemas de pesquisa oferecidos pelos tribunais, não realizam a busca do argumento de pesquisa no inteiro teor do documento, limitando seu escopo à ementa. Infelizmente, é notório no meio jurídico que, freqüentemente, a precisão da classificação encontrada nas ementas é deficitária. Por vezes estão incompletos os termos descritores constantes da ementa. Outras vezes encontram-se, ali, termos descritores referentes a assuntos que não são objeto do debate registrado no inteiro teor do documento;

2. **Argumento de Pesquisa**: Os sistemas de pesquisa oferecidos pelos tribunais recuperam documentos que contenham as palavras digitadas no argumento de pesquisa que, na melhor das hipóteses, faculta o uso de operadores *booleanos*. Embora tais operadores agreguem o benefício de permitir pesquisas mais específicas, muitos usuários não conseguem assimilar sua lógica e, por sentirem-se desconfortáveis com tal interface, não se beneficiam dos recursos disponíveis. Além disto, tal sistemática não propicia a possibilidade de encontrar-se documentos que não contenham algum dos argumentos de pesquisa mas que versem sobre assunto semelhante a documentos que contenham tal argumento. Analogamente, esta sistemática pode recuperar documentos que contenham o argumento de pesquisa, mas cujo tema seja diverso daquele buscado pelo usuário.

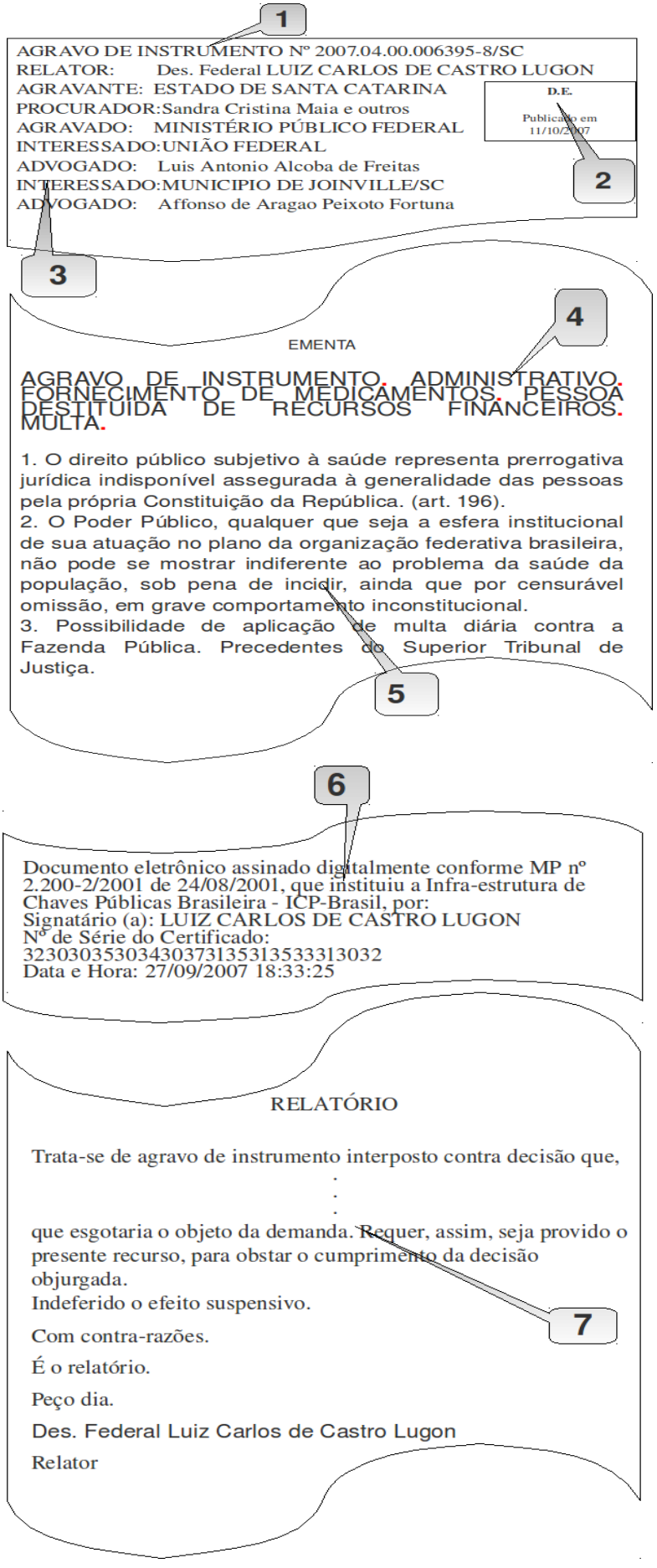


Figura 4.1 – Jurisprudência do TRF/4ª

Tendo em vista a implantação do processo eletrônico desde janeiro de 2010, pretende-se utilizar os documentos anexados ao processo pelas partes como argumento de pesquisa. Ou seja, submete-se o documento ao classificador treinado com a jurisprudência e recupera-se os documentos que compõem o conjunto de treino da respectiva classe. Além disto, o escopo da pesquisa não será limitado às ementas, mas ampliado, abrangendo o inteiro teor do documento.

A recuperação de documentos utilizando solução baseada em aprendizado de máquina aqui descrita, permitirá: a) recuperar documentos que versem sobre o tema pesquisado, ainda que não contenham as palavras constantes do argumento de pesquisa [MFBS+00]; b) desobrigar o usuário de assimilar conhecimentos de álgebra *booleana* como condição para usufruir dos benefícios de uma pesquisa mais específica.

Considerando as deficiências da classificação ementária, bem como a grande quantidade de documentos que compõem a jurisprudência de cada corte e, por sua vez, a grande quantidade de cortes em nosso país, não há como prover um conjunto de documentos devidamente rotulados para treinar um classificador. Optamos, então, a exemplo dos trabalhos de Feinerer e Hornik [FH08] e Aggarwal, Gates e Yu [AGY04], revisados no Capítulo 3, por experimentar um processo de agrupamento de documentos para prover as classes a serem utilizadas pelo categorizador.

No entanto, em nossa revisão bibliográfica, vimos que algoritmos de agrupamento clássicos, como o *K-Means* [Mac67], necessitam pré-configuração da quantidade de grupos a serem gerados. Uma vez que não se conhecem, *a priori*, nem os temas debatidos nem a sua quantidade, buscamos implementar um processo de geração de classes através de agrupamento de documentos de jurisprudência para treinar um categorizador, atendendo aos seguintes quesitos:

1. Reduzir os problemas advindos dos erros de classificação encontrados nas ementas dos documentos que compõem a jurisprudência;

2. Descobrir as classes a serem utilizadas pelo categorizador sem exigir que se configure previamente a sua quantidade.

4.2 Aporte Teórico Utilizado

O algoritmo proposto por Aggarwal, Gates e Yu [AGY04] revisado na Seção 3.4, pressupõe que, dado um *corpus* composto integralmente de documentos previamente classificados, seja possível, partindo desta classificação inicial, obter automaticamente um novo conjunto de classes que, sob julgamento humano, seja qualitativamente equivalente ou superior à taxonomia original.

Tendo em vista as deficiências da classificação ementária, o problema do agrupamento e classificação de documentos jurídicos a ser tratado neste estudo apresenta características

muito semelhantes àquelas tratadas no experimento de Aggarwal, Gates e Yu [AGY04], onde os documentos estão previamente classificados, mas acredita-se que o conjunto de atributos alvo possa ser melhorado.

No entanto, há que se notar uma diferença, posto que ignorar tal divergência implica em resultados cujo impacto pode ser determinante para o fracasso deste estudo: o algoritmo de agrupamento daqueles autores realiza descartes, a título de ruído, de documentos, no passo de atribuição de documentos, e de grupos, no passo eliminação de grupos.

Considere-se, hipoteticamente, a situação de um réu preso sendo que a única forma de convencer um juiz a soltá-lo é a argumentação de outro juiz libertando um outro réu numa situação equivalente. Considere-se, também, a hipótese de que não haja nenhum outro caso semelhante a este. Se tal documento único que faça a diferença entre manter-se preso ou libertar o réu for descartado como ruído, uma vida será, definitivamente, arruinada.

Por estas razões, optamos por adotar o algoritmo de Aggarwal, Gates e Yu [AGY04], propondo sobre o mesmo algumas evoluções, tendo em vista seu uso em *corpus* jurídico, a saber:

1. eliminar o descarte de documentos e o descarte de *clusters*;
2. incluir uma operação de divisão de *clusters*;
3. testar variações no limiar para união de *clusters*.

Note-se que este algoritmo adota o pressuposto de relacionamento um-para-um entre os grupos gerados e as classes utilizadas para treinar classificador. Embora Cong, Lee, Wu e Liu [CLW+04] questionem este pressuposto e apresentem algoritmo para solucionar este problema, tem como requisito o uso de um conjunto de dados rotulados que guiem o particionamento dos dados não rotulados para garantir que, em cada partição, haja um mapeamento um-para-um entre grupos e classes. Apesar dos dados que dispomos estarem rotulados, questiona-se a qualidade desses rótulos e em nossa proposta buscamos melhorar este conjunto de rótulos. Assim, não podemos aplicar o algoritmo de Cong, Lee, Wu e Liu [CLW+04] pois este se baseia numa confiança, que não dispomos, nos rótulos de classe pré-existentes.

4.3 Visão Geral da Solução Adotada

Conforme ilustrado na Figura 4.2, neste estudo propomos experimentar processo composto de duas fases, “A” e “B”, no qual:

1. submetem-se documentos obtidos do *corpus* de jurisprudência $\mathcal{J}$, a um processo de agrupamento, gerando grupos, $\vec{S}_i''$, que determinam as classes a serem usadas pelo categorizador;

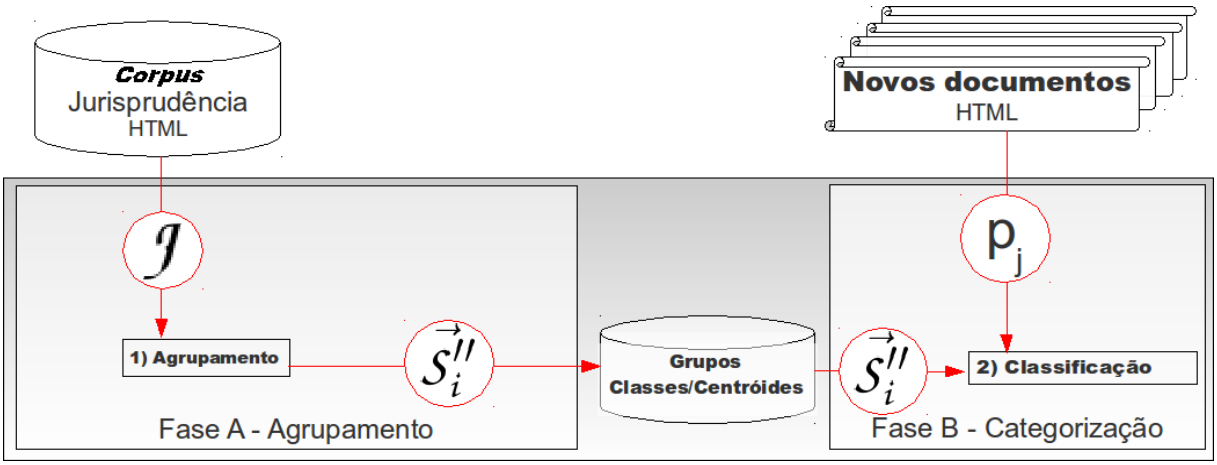


Figura 4.2 – Processo de Agrupamento e Classificação

2. novos documentos p_j são submetidos ao categorizador, que utiliza as classes geradas pelo agrupamento, $\vec{S_i''}$, para classificá-los. Embora omitido na Figura 4.2 visando a sua clareza, uma vez determinada a classe do novo documento, recuperam-se os documentos constantes da jurisprudência que compõem o grupo correspondente a esta classe.

Os códigos identificadores dos documentos de jurisprudência $j_k \in C_i$ que compõem o grupo correspondente à classe obtida são registrados no Processo Eletrônico p_j, permitindo que o usuário consulte o inteiro teor da respectiva jurisprudência j_k.

Após o julgamento do processo jurídico, o magistrado produz um novo documento, contendo a decisão judicial e o integra à jurisprudência. Por esta razão, a Fase "A" deve ser executada novamente. A periodicidade em que esta fase deva ser reexecutada é um ajuste que os administradores deste sistema poderão adequar às suas necessidades específicas[1] e não faz parte do escopo deste estudo.

4.4 Detalhamento da Solução Adotada

Na Figura 4.3 detalhamos os processos de agrupamento para geração de classes, Fase "A", descrito na Seção 4.5.1, e classificação de novos documentos, Fase "B", descrito na Seção 4.5.2.

Na Fase "A", temos:

1. **Pré-Processamento**: obtêm-se os documentos j_i com decisões em processos judiciais, do *corpus* de jurisprudência $\mathcal{J}$ e realiza-se o pré-processamento, conforme descrito na Seção 4.4.2, onde, de cada documento j_i obtem-se um vetor de atributos $\vec{j_i} = (a_{i_0}, ..., a_{i_n})$ e a_{i_n} é o enésimo atributo extraído de j_i. Obtem-se, assim, um conjunto de vetores de atributos $\mathbb{J} = \{\vec{j_i} \in \mathbb{J}\}$;

[1] Diariamente, semanalmente, etc.

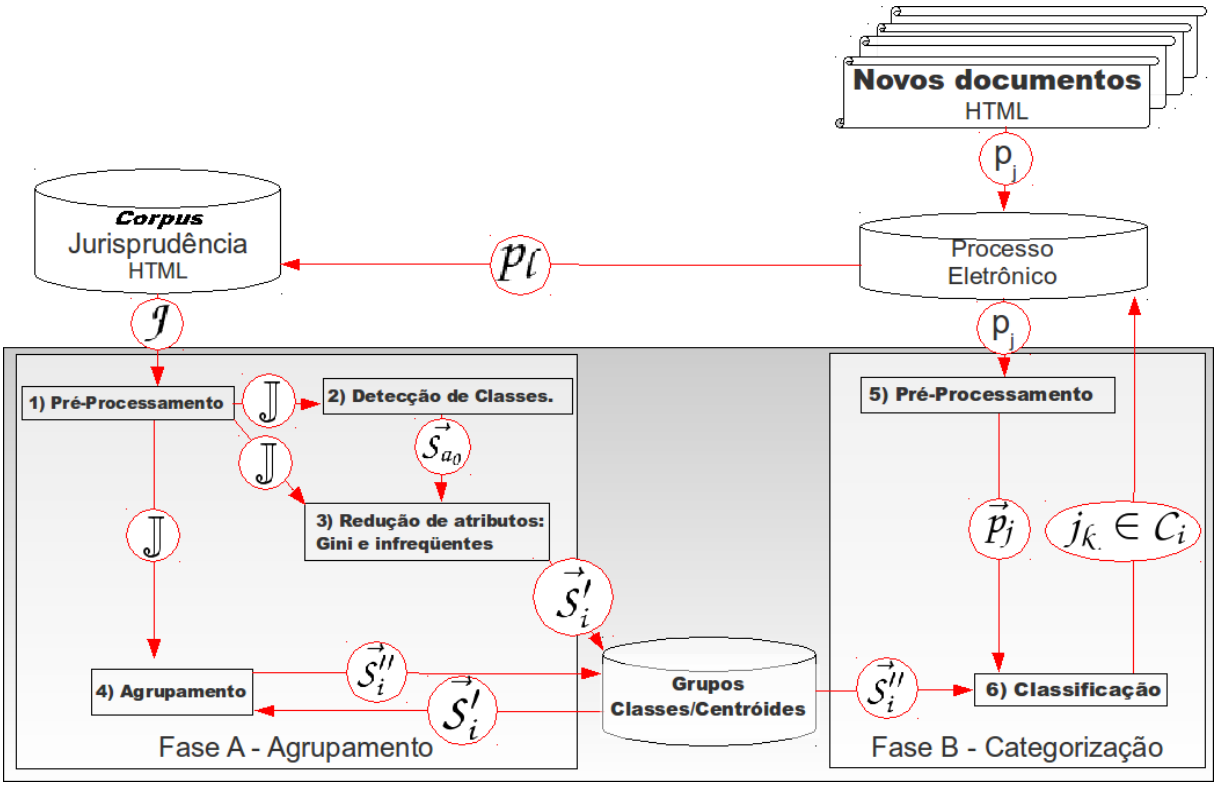


Figura 4.3 – Arquitetura detalhada do agrupamento e da categorização

2. **Detecção de Classes**: seleciona entre os atributos de $\vec{j}_i$, como rótulo de classe C_{a_0} inicial, o primeiro atributo a_0 obtido das respectivas ementas. Considera-se cada classe como grupo inicial $\vec{S}_{a_0}$;

3. **Redução de atributos**: são descartados de $\mathbb{J}$, os atributos a_i que tenham os maiores Índices Normalizados Gini, ou que ocorram em apenas um documento; dos atributos a_i restantes presentes nos documentos $\vec{j}_i \in \mathbb{J}$ calculam-se os centróides $\vec{S}'_i$;

4. **Agrupamento**: agrupam-se os documentos de $\mathbb{J}$, usando como sementes iniciais dos grupos os centróides $\vec{S}'_i$, conforme descrito na Seção 4.5.1, alterando a relação documento/grupo, ou gerando novos grupos, produzindo novo conjunto de centróides $\vec{S}''_i$.

Na Fase “B”, temos:

5 **Pré-Processamento**: obtêm-se os documentos p_j juntados aos Processos Eletrônicos $\mathcal{P}$ e realiza-se o pré-processamento, conforme descrito na Seção 4.4.2 onde, de cada documento p_j, obtem-se um vetor de atributos $\vec{p}_j = (a_{j_0}, ..., a_{j_n})$ e a_{j_n} é o enésimo atributo extraído de p_j. Obtem-se, assim, um conjunto de vetores de atributos $\mathbb{P} = \{\vec{p}_j \in \mathbb{P}\}$;

6 **Classificação**: os vetores de atributos $\vec{p}_j$ são classificados em uma das classes C_i definidas pelos grupos $\vec{S}''_i$ gerados na Fase “A”, utilizando algoritmo descrito na Seção 4.5.2 e na Seção 3.4. Os códigos identificadores dos documentos de jurisprudência $j_k \in C_i$ que compõem o grupo correspondente à classe obtida são registrados no

Processo Eletrônico p_j, permitindo que o usuário consulte o inteiro teor da respectiva jurisprudência j_k.

À medida que os magistrados julgarem os processos, novos documentos p_ℓ serão incluídos no *Corpus* de Jurisprudência e, conseqüentemente, serão necessárias, de tempos em tempos, novas execuções do agrupamento, fase "A".

Para implementar nossos exemplos de uso, foi montado *corpus* composto de documentos obtidos através do *site* do Tribunal Regional Federal da 4ª Região, conforme descrito na Subseção 4.4.1. O pré-processamento destes documentos consiste em: 1) extração de palavras e referências legislativas usando *Parser* desenvolvido conforme descrito na Seção 4.4.3; 2) lematização usando *lematizador* híbrido probabilístico e baseado em regras; e 3) identificação de termos usando tesauros jurídicos, conforme a Seção 4.4.2.2.

Com redução da dimensionalidade dos atributos, foram eliminados aqueles com maior Índice Normalizado Gini ou que ocorriam em, apenas, um documento, vide Seção 3.4.

Nas subseções a seguir, apresentamos detalhamento da implementação da solução proposta.

4.4.1 Composição do *Corpus*

O *corpus* foi construído com documentos de jurisprudência do Tribunal Regional Federal da 4ª Região através do seguinte caminho de *hyperlinks*: "Jurisprudência" $\Rightarrow$ "TRU4 e Turmas Recursais" $\Rightarrow$ "Consulta Jurisprudência da TRU4 e Turmas Recursais", que leva à seguinte URL: http://www.trf4.jus.br/trf4/jurisjud/pesquisa.php?tipo=2 . O formulário constante desta página foi preenchido marcando-se os campos "Acórdãos", "Súmulas" e "Decisões Monocráticas a partir de 08/2006" e selecionando-se o período de 9 de janeiro de 2.006 a 27 de maio de 2.009. Desta maneira, foram selecionados todos os documentos proferidos por estas turmas no período. Isto resultou num conjunto composto de 43.806 documentos. Alguns documentos referiam-se a processos protegidos pelo sigilo judicial e continham, apenas, uma mensagem informando a existência desta proteção. Após eliminá-los, restaram 43.704 documentos.

No entanto, Aggarwal, Gates e Yu propuseram em [AGY04] algoritmo de *Hard Clustering* como forma de definir novas classes para posterior categorização de documentos. Considerando-se que algoritmos deste tipo assumem o pressuposto de que cada documento trata de um único tema e que é comum que os litígios judiciais abordem múltiplos temas igualmente relevantes, considerou-se a necessidade de, neste estudo, descartar do *corpus* os documentos que versassem acerca de múltiplos temas, a fim de evitar que a presença de termos característicos de temas distintos em um único documento torne-se fonte geradora de erros de agrupamento/classificação.

Desta maneira, considerando-se que:

1. a classificação expressa na ementa dos documentos segue a terminologia padronizada no Tesauro da Justiça Federal[2] [SMS+07], mantido pelo Conselho da Justiça Federal[3];

2. ao redigir a ementa, o judiciário transcreve toda a hierarquia dos termos, desde a raiz até o termo mais específico, realizando um caminhamento em profundidade sobre a árvore do tesauro;

3. tal redação apresenta uma sintaxe regular, separando os termos por um ponto '**.**' ou um hífen '**-**';

descartaram-se todos os documentos que tinham termos de ramos distintos dos tesauros, restando, então, 2.612 documentos, que foram analisados por especialista humano que eliminou os documentos multitemáticos ainda restantes (não detectados pela heurística descrita acima), classificando os demais 1.192 documentos, que constituíram o *corpus* aqui utilizado. A Figura A.1, Apêndice A, apresenta a tela do programa desenvolvido para este fim.

4.4.2 Pré-Processamento de Documentos

4.4.2.1 Estruturas Terminológicas

Para poder identificar a ocorrência de termos jurídicos nos documentos, foram utilizados dois tesauros especializados no domínio: o Vocabulário Controlado Básico, VCB [JAS+07] ; e o Tesauro da Justiça Federal, TJF [SMS+07], também conhecido como Vocabulário Controlado da Justiça, VCJ.

O Senado Federal Brasileiro[4] é o mantenedor do VCB [JAS+07], que abrange vários domínios do conhecimento, com foco no domínio do Direito, que representa 3.400 termos.

Ele está estruturado como um tesauro. Assim, há indicações de equivalência de termos. Sendo, portanto, possível verificar que "crime por computador" é semanticamente equivalente a "crime de informática" e o mesmo se aplica a "Uniformização de jurisprudência" e "Súmula vinculante".

Este tesauro está disponível em formato PDF e, para que um programa possa utilizar seus dados, foi preciso convertê-lo para formato textual e extrair as informações com um *parser*. Apesar de, originalmente, tratar-se de texto livre, obedece a uma estrutura sintática bastante regular, com algumas irregularidades quando um campo usa mais de uma linha. Nesse caso, foi necessário juntar manualmente as linhas para que cada campo estivesse inteiramente definido em uma única linha.

Na Tabela 4.1, pode-se ver um excerto do conteúdo do Tesauro. As linhas 1, 12, 14 e 16 iniciam a definição de um termo. Nas linhas 2, 3 e 4, após as palavras "NÃO USE"

[2] http://www2.jf.jus.br/jspui/bitstream/1234/5509/3/tesauro_juridico.pdf .
[3] http://www.jf.jus.br/cjf .
[4] http://www.senado.gov.br/

Tabela 4.1 – Excerto de Estrutura do Tesauro do Senado Federal

A Estrutura do Tesauro	
1.	Mora
2.	NÃO USE Direito de mora
3.	NÃO USE Mora do credor
4.	NÃO USE Mora do devedor
5.	TG Pagamento
6.	TE Purgação da mora
7.	TR Cláusula penal
8.	TR Inexecução das obrigações
9.	TR Juros
10.	TR Perdas e danos
11.	CDD 342.1422
12.	Mora do credor
13.	USE Mora
14.	Mora do devedor
15.	USE Mora
16.	Direito de mora
17.	USE Mora

encontram-se definições não oficiais equivalentes do termo. Na linha 5, após o "TG" (termo geral), encontra-se o hiperônimo do termo. Na linha 6 encontra-se um hipônimo, logo após o "TE" (termo específico). Nas linhas 7, 8, 9 e 10, após o "TR" encontram-se os termos relacionados, mas não equivalentes. Nas linhas 13, 15 e 17, após o "USE" encontram-se os termos equivalentes (e oficiais). Finalmente, na linha 11, após o "CDD", encontra-se o código de classificação das bibliotecas.

Apesar de possuir esta estrutura hierárquica, está longe do Tesauro ser uma única grande árvore. De acordo com Jaegger *et al.* [JAS+07], está, na verdade, muito fragmentado, apresentando uma grande quantidade de sub-árvores desconectadas. Assim, esta estrutura hierárquica do Tesauro foi ignorada neste estudo. Focamos nas relações de equivalência, com vistas à normalização dos termos buscando reduzir a dimensionalidade dos atributos. As relações "TR" são "dicas" para a existência de alguma forma de relação entre os termos, mas não há informações mais detalhadas a respeito da natureza de tais relações. Assim, também foram ignoradas.

O TJF[5] [SMS+07], assim como o VCB, está no formato PDF e também foi preciso convertê-lo para formato textual e extrair as informações com um *parser*. A sintaxe utilizada é muito semelhante à do VCB. Assim, "TG" indica um hiperônimo e "TE" indica um hipônimo. Não se encontra fracionado em subárvores como o VCB. Apresenta-se na forma de um grafo conexo. Conforme a Tabela 4.2, verifica-se que os rótulos "TG" são numerados. Esta numeração indica a distância vertical entre os nodos. A presença de dois rótulos "TG1" indica que há dois hiperônimos com distância de um nível. O termo "crime" e, na seqüência,

[5]Disponível para *download* em http://www2.jf.jus.br/jspui/bitstream/1234/5509/3/tesauro_juridico.pdf

Tabela 4.2 – Estrutura do Tesauro da Justiça Federal

A Estrutura do Tesauro
1. LATROCÍNIO
2. TG1 CRIME HEDIONDO
3. TG2 CRIME
4. TG3 DELITO
5. TG1 ROUBO
6. TG2 CRIME CONTRA O PATRIMÔNIO
7. TG3 CRIME
8. TG4 DELITO
9. TR MORTE
10. CAT DPN/DPN21

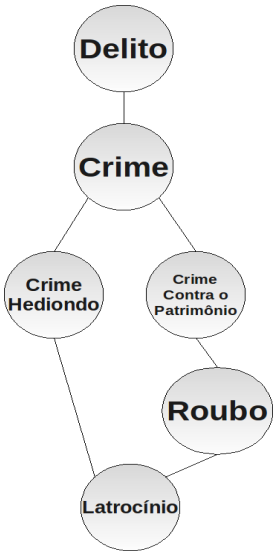


Figura 4.4 – Exemplo de Estrutura de grafo presente no Tesauro da Justiça Federal

o termo “delito”, são alcançáveis por dois caminhos distintos. A Figura 4.4 mostra mais claramente estas relações.

No presente estudo, são utilizadas, apenas, as indicações de equivalência de termos. Como apresentado na Tabela 4.1, o VCB utiliza a expressão “NÃO USE” para indicar termos equivalentes ao constante da linha 1. Da mesma maneira, na Tabela 4.3, vemos que o TJF utiliza a expressão “UP’ para indicar a equivalência dos termos. A expressão regular, no padrão POSIX, “(NÃO USE|UP) ($[\hat{}\backslash n]$+)” é utilizada para detectar estas relações de equivalência em ambos os vocabulários.

Tabela 4.3 – Sintaxe das Indicações de Equivalência no TJF

A Estrutura do Tesauro
1. LEI BRASILEIRA
2. UP LEGISLAÇÃO BRASILEIRA
3. UP LEGISLAÇÃO NACIONAL

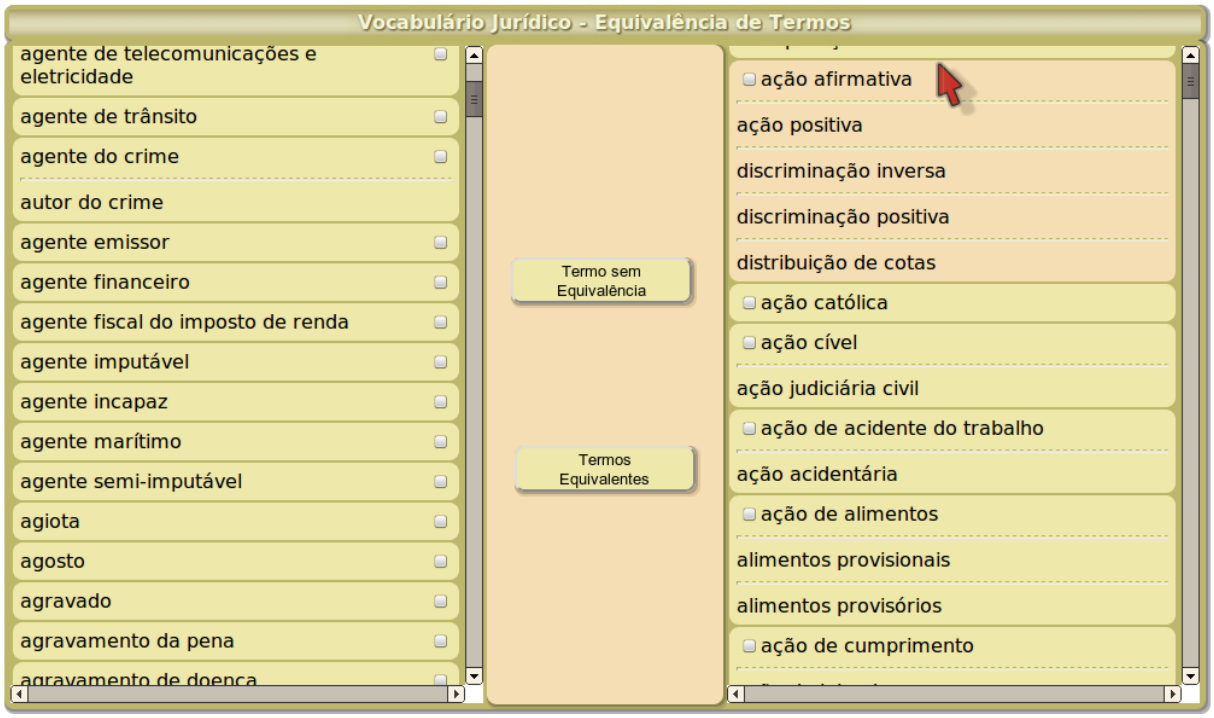


Figura 4.5 – Programa para Mesclagem de Tesauros

4.4.2.2 Base Lexical

Para minimizar o ruído nos dados nos processos de *clustering* e categorização, busca-se identificar diferentes formas de expressão, gerando atributos únicos, bem como identificar ambigüidades no texto, gerando atributos distintos. Para tanto, vários métodos de normalização têm sido utilizados na fase de pré-processamento de textos. Optou-se pela lematização, evitando ambigüidades introduzidas pelo *stemming*, conforme verificado por Korenius *et al.* [KLJ+04] em experimentos com textos finlandeses, e Gonzalez [Gon05], com a língua portuguesa.

Assim, foi construída uma base lexical para auxiliar a lematização. Foram importados e mesclados 3 dicionários: o Dicionário de Português Brasileiro Unitex (Unitex-PB[6]) organizado por Muniz [MN04] e as versões portuguesa[7] e latina[8] do *Wiktionary*, um projeto da *Wikimedia Foundation*[9]. Decidiu-se pela importação deste último, uma vez que muitos termos jurídicos estão em latim[10].

Após importar os dicionários, e montar uma estrutura unificada, detectou-se a ausência de muitas das palavras que ocorriam nos Tesauros e nos documentos do *corpus*. Assim, foi necessário importar estas novas palavras para o dicionário unificado.

Procedeu-se então ao *merge* dos tesauros. De acordo com as indicações neles con-

[6] http://www.nilc.icmc.usp.br/nilc/projects/unitex-pb/web/dicionarios.html
[7] http://dumps.wikimedia.org/ptwiktionary/
[8] http://dumps.wikimedia.org/lawiktionary/
[9] http://www.wikimedia.org/
[10] Como *fumus bonis iuris* (fumaça do bom direito) e *periculum in mora* (perigo na demora), por exemplo.

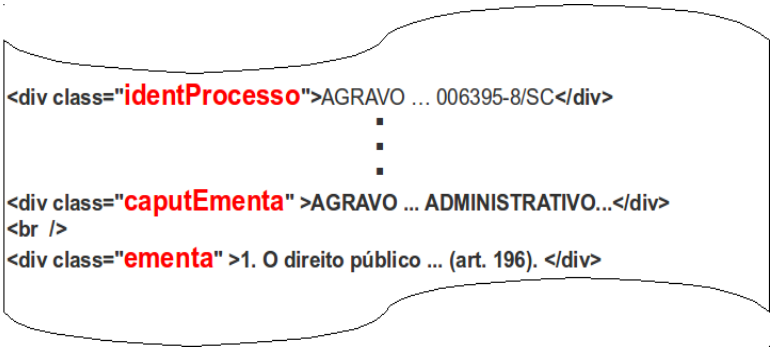


Figura 4.6 – Estrutura da Jurisprudência do TRF/4ª

tidas, gerou-se, para cada um, uma lista de grupos de termos, onde cada grupo contem termos equivalentes entre si. Escreveu-se um programa que comparou os grupos das duas listas, unindo grupos sempre em que ambos apresentassem pelo menos um termo em comum. Por exemplo: o VCB apresentou os termos equivalentes “ação afirmativa”, “ação positiva” e “discriminação inversa”, já o TJF apresentou os termos equivalentes “discriminação inversa”, “discriminação positiva” e “distribuição de cotas”. Em virtude de ambos os grupos conterem o termo “discriminação inversa”, fez-se, automaticamente, a sua fusão, obtendo um único conjunto com os termos equivalentes “ação afirmativa”, “ação positiva”, “discriminação inversa”, “discriminação positiva” e “distribuição de cotas”. O tesauro resultante foi composto, assim, de 1.796 grupos de termos oriundos dos dois tesauros, 7.044 grupos oriundos apenas do VCB e 4.514 grupos oriundos apenas do TJF. Após a mesclagem automática, procedeu-se a uma verificação manual por especialista humano para detectar mais similaridades entre os grupos dos dois tesauros. A Figura 4.5 ilustra a *interface* da ferramenta desenvolvida para este fim.

Desenvolveu-se, então, um *parser* e um lematizador, descritos na Seção 4.4.3, para lematizar os termos dos tesauros. As seqüências de *lemmata* obtidas dos tesauros foram armazenadas na base lexical. Termos equivalentes, como “crime por computador” e “crime de informática”, receberam mesma identificação.

4.4.2.3 Arquitetura do Pré-Processamento

A Figura 4.7 mostra a arquitetura do pré-processamento proposto.

Os documentos do *corpus* construído estão em formato HTML e constituem a entrada para o pré-processamento. Possuem a estrutura geral apresentada na Figura 4.6, onde se vêem metadados como, por exemplo, o número do processo e a ementa, de especial interesse e cujo *caput* apresenta uma classificação composta de termos padronizados constantes do Tesauro da Justiça Federal, dispostos em seqüência consoante a hierarquia do referido tesauro. O conteúdo textual a ser processado encontra-se entre *tags* “DIV” cujas classes CSS são “caputEmenta”, “ementa”, “paragrafoNormal” e “citacao”, e é extraído aplicando expressões regulares.

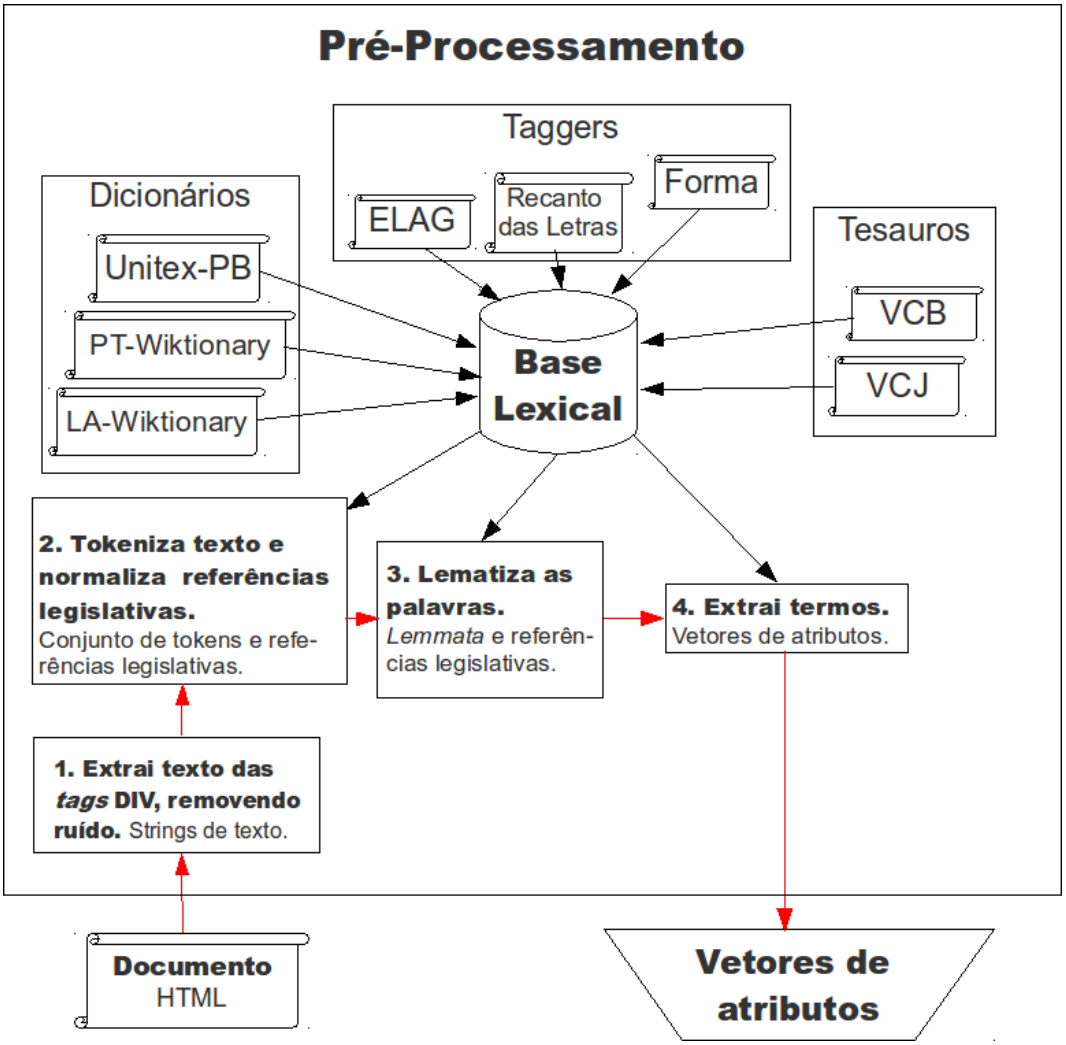


Figura 4.7 – Arquitetura do Pré-Processamento

4.4.3 *Parsing*, Lematização, Reconhecimento de Termos e Descarte de Atributos

Para construir os vetores de atributos, que são compostos de termos dos documentos, foram eliminados metadados e extraídos parágrafos dos documentos em HTML, usando expressões regulares que buscaram as *tags* com classes CSS indicando seu conteúdo. Isto resultou em um *array* de parágrafos com texto puro, exceto pela presença de *tags* “B”, “I”, “U” e “DD”. As três primeiras são indicadores de negrito, itálico e sublinhado; a última é indicadora de itemização. Embora não sejam utilizadas neste estudo, sendo descartadas após a extração de termos, optou-se por não excluí-las para que, em futuros trabalhos, sejam mais uma alternativa à disposição para o pré-processamento. As *tags* de negrito e sublinhado indicam que o autor do documento confere maior relevância ao trecho em destaque. Isto poderia ser levado em consideração para dar maior peso aos atributos gerados. O itálico poderia auxiliar o *tagger* a, por exemplo, delimitar um termo.

Tabela 4.4 – Normalização de referências legislativas

“artigo 34 do decreto-lei n° 8192 de fevereiro de 1972”	“dl 8192/1972”, “dl 8192/1972 art. 34”
“decreto-lei n° 8192 de fevereiro de 72, por meio dos artigos 34, 35 e 36”	“dl 8192/1972”, “dl 8192/1972 art. 34”, “dl 8192/1972 art. 35”, “dl 8192/1972 art. 36”

4.4.3.1 *Parser*

O *parser* desenvolvido extrai *tokens* usando expressões regulares que reconhecem palavras, números, pontuação, URLs, e-mails, datas, números de processo e referências legislativas. A Tabela 4.4 apresenta exemplos de normalização de referências legislativas produzidas, mostrando que referências a um ou mais artigos de uma norma, resultam num *token* para a norma, além de um *token* para cada par norma-artigo. O processo visa aumentar mais a similaridade entre os documentos que abordem o mesmo par artigo-norma e aumentar um pouco a similaridade aqueles que referenciarem artigos diferentes de uma mesma norma. Evita, ainda, aumento de similaridade indesejado entre documentos que façam referência a artigos de mesmo número em normas distintas.

4.4.3.2 Lematizador

O lematizador pesquisa na base lexical os *tokens* identificados como palavras, recuperando os respectivos *lemmata*. Os *tokens* não encontrados na base são descartados. Havendo mais de um *lemma* relacionado a um *token*, faz-se a desambiguação lexical. As únicas alternativas de desambiguação são os *lemmata* obtidos da base lexical[11]. O desambiguador itera entre dois desambiguadores não gulosos. Um baseado em regras e o outro probabilístico. Ambos desambiguadores podem decidir 1) pela desambiguação, 2) eliminar uma das ambigüidades ou 3) não realizar nenhuma operação. A eliminação de uma das alternativas, proporciona ao próximo desambiguador melhores condições de decisão.

Na primeira iteração, o desambiguador probabilístico assume comportamento não guloso, desambiguando somente ante grandes diferenças de probabilidades entre as alternativas de desambiguação. Na segunda invocação, o desambiguador probabilístico se torna guloso, selecionando a alternativa de maior probabilidade.

Para o desambiguador baseado em regras importou-se 69 regras do ELAG (*Elimination of Lexical Ambiguities by Grammars*) providas por Muniz[12] [MN04]. Foram acrescentadas mais 271 novas regras inspiradas nas regras gramaticais apresentadas por Ricardo Sérgio

[11] Por exemplo: o *token* “par”, em função do sufixo “ar”, poderia ser etiquetado como verbo por um desambiguador guiado pelo sufixo. No entanto, não há um lemma etiquetado como verbo para este *token* na base lexical.

[12] http://www.nilc.icmc.usp.br/nilc/projects/unitex-pb/web/gramaticas.html .

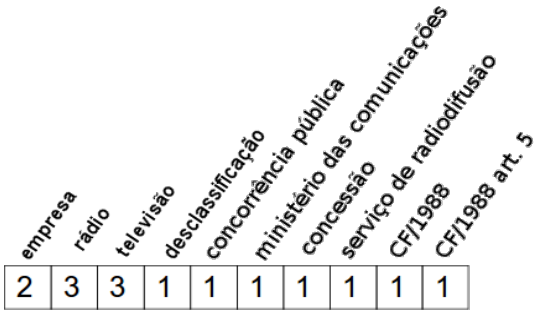


Figura 4.8 – Exemplo de vetor de atributos

no *site* "Recanto das Letras"[13].

O desambiguador probabilístico usa tabela de probabilidades do *tagger* denominado FORMA de Gonzalez [Gon05], que decide com base em sufixos do *token*. Neste estudo, seu algoritmo sofreu duas modificações: 1) são considerados apenas os *lemmata* presentes na base lexical, e, 2) na primeira iteração, assume um comportamento não guloso.

4.4.3.3 Reconhecimento de Termos

Após a lematização, identificam-se os termos dos tesauros. A Figura 4.8 apresenta um modelo de vetor de atributos, gerado ao fim do pré-processamento, resultante do seguinte texto: "*1. Trata-se de Ação Ordinária proposta pela empresa Porto de Cima Rádio e Televisão Ltda, objetivando a desclassificação das empresas Rádio e Televisão Rotioner Ltda (ROTIONER) e Rádio e Televisão Canal 29 do Paraná Ltda (SESAL) da Concorrência Pública nº 150j97-SSRjMC, promovida pelo Ministério da Comunicações, destinada a outorgar a concessão do serviço de radiodifusão de sons e imagens na localidade de Curitiba, Paraná.(...) É que a Constituição Federal/88, em seu artigo 5º, assevera que (...)*".

4.4.3.4 Descarte de atributos

Além disto, implementou-se, também, uma variação no pré-processamento dos documentos. A seleção de atributos empregada pelos autores baseou-se no índice Gini, descartando as palavras que tinham distribuição muito homogênea entre as classes e, também, descartando palavras que ocorriam em poucos documentos.

Neste estudo avaliou-se, alternativamente, uma combinação da solução dos autores com a proposta de Feinerer e Hornik [FH08], que apresenta o *Keyword Based Clustering*. No entanto, em [FH08], definem-se manualmente os termos que compõem os centróides dos *clusters*. Em nosso exemplo de uso, a alternativa de pré-processamento que se propõe é a mudança da representação dos documentos no modelo de espaço vetorial de *bag of words* para *bag of terms and law references* utilizando o índice Gini para eleger os termos e

[13] http://recantodasletras.uol.com.br/gramatica/638792 http://recantodasletras.uol.com.br/gramatica/88821 http://recantodasletras.uol.com.br/gramatica/80651 http://recantodasletras.uol.com.br/gramatica/78991 .

referências legislativas com a distribuição mais desigual ou que tenham ocorrido em apenas um documento.

Para tanto, foram utilizados tanto os termos do TJF quanto do VCB. Note-se que o documento com a decisão judicial, embora seja redigido por um juiz, contém citações de textos produzidos pelas partes do processo, que podem adotar o VCB. O Ministério Público Federal, por exemplo, adota, oficialmente, o VCB. Considerando-se que não é incomum que o juiz transcreva diversos parágrafos da argumentação do Ministério Público, acrescentando, ao final, frases como “é a minha decisão” ou “decido de acordo com o Ministério Público”, percebe-se que, nestes casos, os termos relevantes para o pré-processamento serão, necessariamente, oriundos do VCB.

4.5 Processo de Agrupamento e Classificação

4.5.1 Agrupamento

Conforme exposto no *caput* deste capítulo, o problema de agrupamento e classificação de documentos jurídicos a ser tratado neste estudo apresenta características muito semelhantes àquelas tratadas no experimento de Aggarwal, Gates e Yu [AGY04], excetuando-se, no entanto, o descarte de documentos e grupos, já ressalvado, também, no referido caput.

O critério de parada do processo de *clustering* estabelecido por Aggarwal, Gates e Yu [AGY04] baseou-se na redução de atributos. Quando a quantidade de atributos era inferior a 200 encerrava-se o *clustering*. No estudo por nós desenvolvido, os vetores de atributos possuem dimensionalidade bem inferior ao dos referidos autores em função de cada atributo representar termos jurídicos ou referências legislativas e não as palavras dos documentos. Por esta razão, após alguns testes, decidiu-se estabelecer o limite mínimo de 20 atributos como critério de parada.

Em seu artigo, Aggarwal, Gates e Yu [AGY04], abordam a questão da divisão de *clusters* em grupos menores. Informam que o algoritmo não suporta tal operação por crerem que esta seria não-supervisionada e que isto poderia gerar incoerências com os rótulos de classe original. Neste estudo optou-se por implementar esta operação e analisar a validade de tal ponderação. Foram definidas e experimentadas duas alternativas de algoritmos para implementar esta operação, apresentados nas Subsubseções a seguir.

4.5.1.1 Algoritmo de Divisão

Este algoritmo acrescenta um passo de divisão na iteração principal do algoritmo de Aggarwal, Gates e Yu [AGY04], no qual os *clusters* podem ser divididos em grupos menores. Para tanto, definiu-se que seriam selecionados para divisão os *clusters* que apresentem muita variação de similaridade entre seus respectivos centróides e documentos, baseado

na hipótese de que esta seja uma característica encontrada em grupos que contenham subgrupos razoavelmente bem separados.

Assim, sejam $\bar{\delta}$ e $\delta\delta$ a média e o desvio-padrão dos desvios-padrão das similaridades *intra-cluster* e C_n, δ_n o n-ésimo *cluster* e o desvio-padrão de suas similaridades internas. Divide-se todo C_n se $\delta_n > \bar{\delta} + 2 \times \delta\delta$. Para efetivar a divisão definiu-se um novo passo no algoritmo proposto por Aggarwal, Gates e Yu [AGY04], no qual se realiza um processo de *subclustering* aglomerativo, definido abaixo e detalhado no Algoritmo 1.

1. O *cluster* é fracionado em *subclusters* contendo um único documento. Faz-se uma redução de atributos[14] usando o índice Gini;

2. Realiza-se uma iteração semelhante à iteração principal, porém sem o passo de divisão de *clusters*. Os dois primeiros passos não são executados na primeira iteração porque cada *cluster* contém um único documento.

   (a) **Atribuição de Documentos**: atribui-se cada documento ao *cluster* de centróide mais similar;

   (b) **Seleção de Atributos**: realiza-se a redução de atributos de acordo com o algoritmo principal;

   (c) **Aglomeração**: realiza-se a união de *clusters* conforme definição do algoritmo principal.

3. O conjunto de *clusters* resultante é retornado para o algoritmo principal, substituindo o *cluster* de onde se originaram.

4.5.1.2 Algoritmo de Divisão Implícita

Conforme descrito no Capítulo 5, a inclusão do passo de divisão detalhado no Algoritmo 1 teve um impacto significativo na velocidade do processamento. Por esta razão, buscou-se uma alternativa que viabilizasse a divisão dos *clusters* sem causar tão grande impacto no custo do processamento. Inspirando-se no algoritmo TOD [FK99] *apud* [LCF+07], foi alterado o passo de atribuição de documentos a *clusters* proposto no algoritmo original de Aggarwal, Gates e Yu [AGY04], substituindo-se o descarte de documentos pela criação de um novo *cluster* contendo unicamente o documento outrora selecionado para descarte, conforme detalhado no Algoritmo 2.

Embora o novo *cluster* seja composto de apenas um documento, novas iterações poderão atrair documentos dos grupos mais próximos. Conforme ilustrado na Figura 4.9, 1) o documento “A” está além do limiar de similaridade do *cluster* e, assim, 2) cria-se um novo *cluster*

[14] Esta redução é utilizada exclusivamente no escopo do processamento do *subclustering*, sendo descartada após a finalização deste passo de divisão.

Divide (D)

begin

S $\leftarrow$ D;
Words $\leftarrow$ Initial_value1;
Threshold $\leftarrow$ Initial_value2;
Minimum $\leftarrow$ Initial_value3;
First_time $\leftarrow$ true;

$$\bar{\delta} \leftarrow \frac{\sum_{i=1}^{|S|} \textbf{SimilarityDeviation}(S_i)}{|S|};$$

$$\delta\delta \leftarrow \sqrt{\frac{\sum_{i=1}^{|S|}(\ \textbf{SimilarityDeviation}(S_n)) - \bar{\delta})^2}{|S|}};$$

StdDeviationThreshold $\leftarrow \bar{\delta} + \delta\delta$;
for *i* $\leftarrow 1$ **to** |S| **do**
  if **SimilarityDeviation**(S_i) > StdDeviationThreshold **then**
    repeat
      if $\neg$ First_time **then**
        $S_i \leftarrow$ **Assign** (S_i);
        $S_i \leftarrow$ **Project** (S_i ,Words);
      end
      $S_i \leftarrow$ **Merge** (S_i ,Threshold);
      First_time $\leftarrow$ false;
      Iteration $\leftarrow$ Iteration $+1$;
    until | **Dimensions**(S_i)| < Minimum ;
  end
end
return(S);

end

MeanSimilarity (S_n)

begin

$$\text{SeedSimilarity} \leftarrow \frac{\sum_{i=1}^{|S_n|} \textbf{DocSimilarity}(\text{Centroid}(S_n), \text{Document}_i)}{|S_n|};$$

return(SeedSimilarity);

end

SimilarityDeviation (S_n)

begin

$$\text{SeedDeviation} \leftarrow \sqrt{\frac{\sum_{i=1}^{|S_n|}(\ \textbf{DocSimilarity}(\text{Centroid}(S_n), \text{Document}_i) - \ \textbf{MeanSimilarity}(S_n))^2}{|S_n|}};$$

return(SeedDeviation);

end

Algoritmo 1: **Divide**

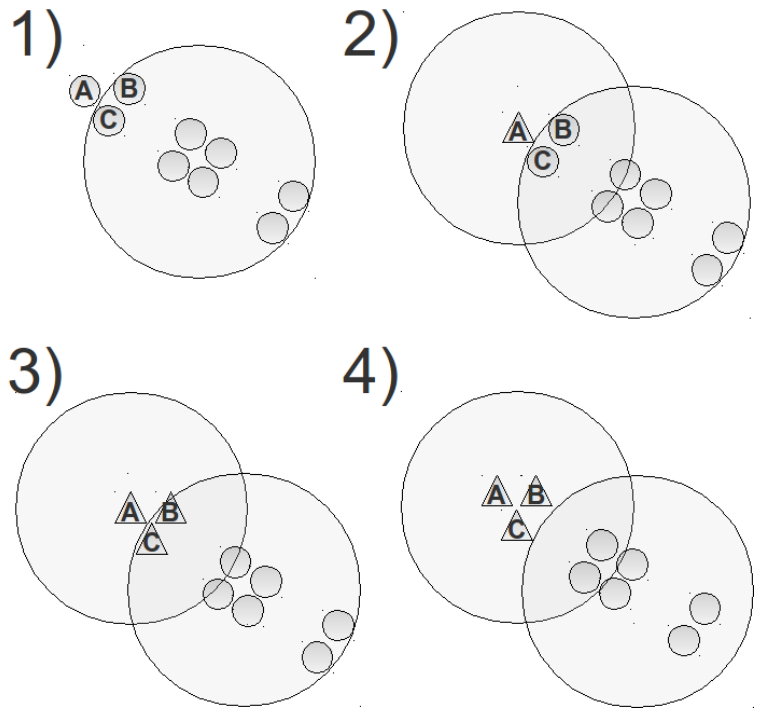


Figura 4.9 – Sucessivas iterações podem atrair documentos para o *cluster* recém-criado

com centróide em “A”, 3) na iteração seguinte, os documentos “B” e “C”, embora estejam dentro do limiar do *cluster* atual, são mais similares ao centróide do novo *cluster*, sendo atribuídos a ele e, por fim, 4) faz-se o recálculo dos centróides.

Assign (D)
begin
  ⋮
  if **DocSimilarity**(Centroid(S_n), Document$_i$) < Threshold **then**
    $S \leftarrow S \cup \{ \{ \text{Document}_i \} \}$;
  end
  ⋮
end

Algoritmo 2: **Assign**

4.5.2 Categorização

Novos documentos a classificar são submetidos ao mesmo pré-processamento para obtenção de vetores de atributos usados no *clustering*. No entanto, não será feito novo cálculo do índice normalizado Gini, serão descartados os mesmos atributos descartados na fase de pré-processamento dos documentos a agrupar, garantindo que os vetores sejam compostos pelos mesmos atributos.

A categorização utiliza classes obtidas assumindo o pressuposto de relação um-para-um com os grupos obtidos na fase de clustering e foi determinada usando o mesmo processo de determinação da classe de maior dominância proposto por Aggarwal, Gates e Yu [AGY04], conforme descrito na Seção 3.4. Este processo baseia-se no cálculo de proximidade do documento a classificar em relação a cada um dos centróides dos grupos correspondentes às classes, usando a mesma função de similaridade de cosseno usada na fase de agrupamento. No entanto, se na fase de agrupamento há um limiar mínimo de similaridade para atribuição de um documento ao grupo[15], na fase de categorização não há limite mínimo de similaridade para a classificação. Há, porém, mais um procedimento, inexistente na fase de agrupamento, onde se determina a dominância de uma classe sobre o documento a ser categorizado, descrita na Seção 3.4, no qual, quando um documento se encontra em uma região limítrofe entre k classes, se faz novo cálculo de proximidade em relação aos respectivos centróides desconsiderando-se, agora, os atributos que sejam não nulos nos centróides envolvidos.

4.6 Considerações Finais

Foi apresentada aqui a arquitetura de nossa proposta, vide Seções . Detalhando, então os passos empreendidos em sua implementação. Foram, então, executados exemplos de uso de nossa reimplementação do algoritmo original de Aggarwal, Gates e Yu [AGY04], e das variações propostas para evolução deste algoritmo, utilizando, para tanto, o *corpus* jurídico organizado conforme descrito na Seção 4.4.1, a fim de descobrir se o uso de algoritmos de aprendizado de máquina podem ser utilizados satisfatoriamente para acelerar o processo de pesquisa de jurisprudência. No próximo capítulo, apresentamos relato da avaliação e análise dos resultados obtidos ao executarmos nossos exemplos de uso.

Foi apresentada aqui a arquitetura de nossa proposta, vide Seções 4.2, 4.3 e 4.4. Detalhando, então, os passos empreendidos em sua implementação, apresen- tamos, na Seção 4.4.2.1, os vocabulários controlados do Senado Federal e do Conselho da Justiça Federal. Na Seção4.4.2.2, construímos nosso dicionário mediante importação de dois dicionários em língua portuguesa e um em latim. Desenvolvemos um parser que reconhece os tokens constantes deste dicionário e referências legislativas, apresentado na Seção 4.4.3, e um lematizador híbrido, que alterna a aplicação de regras gramaticais e cálculo de probabilidade, visto na Seção 4.4.3.2. Desenvolvemos, também, um extrator de de atributos que reconhece os termos jurídicos dos vocabulários controlados e as referências legislativas, apresentado na Seção 4.4.3.3. E completamos o pré-processamento realizando o descarte de atributos baseado no Índice Normalizado Gini, além de atributos pouco freqüentes, conforme exposto na Seção 4.4.3.4. Construímos um corpus com jurisprudência baixada do

[15] Exceto nas variantes do algoritmo em que não há descarte de documento e não se realiza a divisão implícita de grupos.

Tribunal Regional Federal da 4ª Região, conforme descrito na Seção 4.4.1 que, após o pré-processamento, proveu os vetores de atributos para executar os exemplos de uso de nossa reimplementação do algoritmo original de Aggarwal, Gates e Yu [AGY04], e das variações propostas para evolução deste algoritmo, descritas na Seção 4.5.1, a fim de descobrir se o uso de algoritmos de aprendizado de máquina podem ser utilizados satisfatoriamente para acelerar o processo de pesquisa de jurisprudência, e, em especial, atendendo os quesitos de não realizar descarte de documentos ou grupos e implementando duas alternativas de operação de divisão de grupos, descritas nas Seções 4.5.1.1 e 4.5.1.2. Por fim, categorizamos documentos usando classes obtidas assumindo o pressuposto de uma relação um-para-um com os grupos gerados pelo algoritmo evoluído, conforme exposto na Seção 4.5.2. Todas os programas desenvolvidos em nosso exemplo de uso foram implementados usando linguagem PHP[16]. No próximo capítulo, apresentamos relato da avaliação e análise dos resultados obtidos ao executarmos nossos exemplos de uso.

[16]http://www.php.net/

5. Avaliação

5.1 Considerações Iniciais

Para averiguar a efetividade desta proposta junto ao exemplo de uso, avaliando, assim, a aplicabilidade e, portanto, perspectivas de sua implantação em ambiente real, realizaram-se experimentações, detalhadas na Seção 5.2. A Seção 5.3 descreve a avaliação dos resultados obtidos, que foi dividida em dois momentos: avaliação do agrupamento, descrita na Seção 5.3.1 e avaliação da categorização, descrita na Seção 5.3.2. As Seções 2.7 e 2.6 apresentam uma rápida revisão de métodos comumentemente utilizados para avaliação de agrupamentos e de classificação, respectivamente.

5.2 Parâmetros Adotados na Validação

Dividimos os documentos do *corpus* em 3 conjuntos:

1. **Treino**: 716 documentos, para realizar o agrupamento;
2. **Teste**: 238 documentos, para a primeira classificação;
3. **Operação**: 238 documentos, para a classificação final;

O procedimento adotado, como critério de divisão, consistiu em selecionar, seqüencialmente, 3 documentos para o conjunto de treino, 1 para o conjunto de teste e um para o conjunto de operação, reiniciando o processo até que se esgotassem os documentos. Desta maneira, a divisão dos codcumentos ficou ligada à ordem em que os documentos ingressaram no corpus. Essa, por sua vez, seguiu a ordem em que foram realizados os downloads dos documentos. Conforme descrito na Seção 4.4.1, foram buscados os documentos através de pesquisa por data no site do Tribunal Regional Federal da 4ª Região, compreendendo o período de 9 de janeiro de 2.006 a 27 de maio de 2.009. Foram, inicialmente, baixados em ordem cronológica crescente, os documentos do ano de 2.009. Em seguida os documentos do ano de 2.008. Após, os de 2.007 e, finalmente, os de 2.006.

Foram selecionados os 716 documentos do conjunto de treino para agrupar e gerar as classes. Foram extraídos, ao todo, 1.255.266 *tokens* destes documentos. Cada documento apresenta uma média de 1.753,16 *tokens*. Após a extração dos atributos, obteve-se, por documento, uma média de 138,54 atributos (62,9 atributos distintos, em média). O *parsing* e a desambigüação levaram em torno de 1h 30min e a detecção de atributos consumiu em torno de 15 minutos. Assim, o tempo médio de pré-processamento é de menos de 9s por documento.

Tabela 5.1 – Variações empregadas em cada execução do agrupamento

Execução	Opções
1	Algoritmo original, sem alterações
2	Desabilitado o descarte de grupos
3	Desabilitado o descarte de documentos
4	Habilitado o passo de divisão
5	Desabilitados todos os descartes e habilitado o passo de divisão
6	Desabilitados todos os descartes e habilitada a divisão implícita

Obtidos os atributos, foram determinados os grupos/classes iniciais. Para tanto, o primeiro atributo obtido da ementa de cada documento foi usado como rótulo de classe. No Apêndice B, vemos a Tabela B.2 que apresenta um resumo dos grupos iniciais obtidos e a quantidade de documentos associada a cada um.

Descartaram-se, então, os atributos via Índice Normalizado Gini, listados na Tabela D.1, no Apêndice D. Decidiu-se pelo descarte dos 50 atributos com o maior Índice Normalizado Gini. Não foi possível descartar mais atributos devido a alguns documentos e grupos ficarem com poucos atributos. Foram descartados, também, todos os atributos que ocorriam somente em um documento. A Tabela B.1, encontrada no Apêndice B, apresenta as dimensionalidades iniciais dos grupos.

Para melhor observar o efeito das alterações propostas, executamos o algoritmo de agrupamento várias vezes, ativando, seletivamente, cada alteração proposta e, posteriormente, ativando-as em conjunto, conforme indicado na Tabela 5.1. O limiar de similaridade utilizado foi de 50%. O limiar de descarte de grupos foi de 4 documentos. Limiares de similaridade e descarte superiores a estes resultavam em descarte de todos os documentos no algoritmo original de Aggarwal, Gates e Yu [AGY04], pois a exigência de maior similaridade aumentava o descarte de documentos e diminuia a quantidade de documentos no grupos, fazendo com que os grupos atingissem o limiar de descarte e fossem, tambem, descartados. As iterações iniciaram com, no máximo, 200 atributos nos centróides e encerraram-se com, no mínimo, 24 atributos. Cada algoritmo de agrupamento levou entre 30min e 1h 30min de execução, exceto pelo algoritmo que implementou o passo de divisão de grupos, que levou em torno de 3h 30min para executar. O algoritmo de categorização classificou, em média, um documento a cada 2,02s.

Depois de executados os agrupamentos, foram calculados dois índices internos de qualidade dos agrupamentos de cada um dos conjuntos de grupos gerados, detalhados na Seção 5.3.1.

Selecionou-se, então, o conjunto gerado pelo algoritmo evoluído sem descartes de documentos ou grupos e com divisão implícita de grupos, por apresentar a melhor performance média dos índices internos para prover as classes utilizadas em todos exemplos de uso de classificação, descritos na Seção 5.3.2.

Os 238 documentos do conjunto de teste foram categorizados em 161 das 465 classes correspondentes ao grupos obtidos no agrupamento realizado através do algoritmo evoluído selecionado. Os resultados da categorização foram submetidos a validação por especialista humano.

Após a validação por especialista humano, conforme descrito na Seção 5.3.2, analisaram-se os resultados obtidos, verificando que obteve-se uma precisão de, aproximadamente, 57%. Analisou-se, também, a relação entre os verdadeiros/falsos positivos e diversos parâmetros, tais como quantidade de documentos no grupo, quantidade de atributos no centróide e no documento categorizado, quantidade de palavras nos atributos originados de termos jurídicos, etc. Desta análise não se identificou qualquer relação entre estes parâmetros e o sucesso/insucesso na categorização. Por esta razão, suspeitando de que tal relação não tivesse raízes nestes parâmetros, procedeu-se a uma análise mais detalhada, nas Seções 5.4.1 e 5.5, dos casos extremos: os falsos positivos categorizados com alta similaridade e os verdadeiros positivos com baixa similaridade.

Nesta análise, percebeu-se que, em muitos centróides os atributos de maior peso tinham semântica muito genérica e, assim, formulou-se a hipótese de que poder-se-ia minimizar este problema dando pesos proporcionais à semântica dos atributos, conforme detalhado na Seção 5.6. Também percebeu-se que o passo de projeção, onde se faz o recálculo dos centróides, não reconhecia a presença de novos atributos não nulos decorrentes da inclusão de novos documentos.

Procedeu-se à implementação de novo exemplo de uso, retornando ao ponto da detecção dos atributos nos documentos. Desta vez, atribuiu-se pesos proporcionais à especificidade dos atributos. A informação do grau de especificidade dos termos foi obtida a partir dos tesauros, e as referências legislativas receberam pesos arbitrados, conforme critérios detalhados na Seção 5.7. Os demais procedimentos de pré-processamento seguiram o mesmo rito, descartando-se os 50 atributos com o maior Índice Normalizado Gini e os atributos que ocorriam em somente um documento.

Repetiu-se a execução dos algoritmos de agrupamento, descrito na Seção 5.8, conforme o algoritmo original e as cinco variações do algoritmo evoluído. O passo de projeção foi alterado, permitindo que novos atributos não nulos ingressem no centróide em decorrência da inclusão dos atributos dos novos documentos no centróide. No Apêndice C, vemos a Tabela C.2 que apresenta um resumo dos grupos finais obtidos e a quantidade de documentos associada a cada um.

Foi realizado novo cálculo dos índices de qualidade dos agrupamentos e, desta vez, o algoritmo que descarta documentos e não descarta grupos apresentou a melhor performance média. No entanto sua performance média superou a performance média do algoritmo de divisão implícita em, apenas, 2% e, sendo tão pequena a diferença e por não realizar descartes, preferimos selecionar o conjunto de grupos gerado por este último para prover as classes usadas na fase de categorização.

Não foi possível utilizar os 238 documentos do conjunto de operação devido à indisponibilidade de tempo para validação por especialista humano. No tempo que dispúnhamos, a única maneira que encontramos de avaliar ao menos 100 categorizações foi o emprego de dois especialistas humanos. Cada especialista humano avaliou um conjunto de 55 categorizações, composto de um conjunto de 50 categorizações distinto do conjunto recebido pelo outro avaliador, e de um conjunto de 5 categorizações iguais às do conjunto de 5 categorizações recebido pelo outro avaliador. Totalizando, assim, 105 categorizações distintas.

Os 105 documentos selecionados aleatoriamente do conjunto de operação, foram categorizados em 74 das 453 classes correspondentes ao grupos obtidos no agrupamento realizado através do algoritmo evoluído selecionado.

Após a validação pelos especialistas humanos, conforme descrito na Seção 5.9, analisou-se os resultados obtidos, verificando que obteve-se uma precisão de, aproximadamente, 50,5%. Também repetiu-se a análise da relação entre os verdadeiros/falsos positivos e os diversos parâmetros analisados anteriormente. Desta vez, pôde-se identificar que alguns destes parâmetros mantêm um razoável grau de relação com o sucesso na categorização.

5.3 Avaliações Realizadas

A avaliação dos exemplos de uso implementados compreendeu duas fases:

1. **Avaliação do agrupamento**: procedeu-se à análise dos agrupamentos utilizando o algoritmo originalmente proposto por Aggarwal, Gates e Yu [AGY04] e as evoluções aqui propostas, conforme detalhado na Seção 5.3.1;

2. **Avaliação da classificação**: dentre os métodos de agrupamento implementados selecionou-se aquele que apresentou melhor combinação de avaliação por índices internos com velocidade de processamento. Realizou-se, então, categorização dos documentos do conjunto de teste utilizando as classes obtidas através do agrupamento selecionado e procedeu-se a uma avaliação por especialista humano, detalhada na Seção 5.3.2.

5.3.1 Análise dos Agrupamentos

Para comparar os agrupamentos obtidos nas execuções dos vários algoritmos, buscamos medidas de qualidade de agrupamentos que, embora não exigissem validação por especialista humano, face à indisponibilidade de tempo, oferecessem resultados que se aproximassem daqueles obtidos mediante sua validação. Ingaramo, Pinto, Rosso e Errecalde [IPR+08] realizaram experimento através do qual, após a geração dos *clusters*, uti-

lizando os *corpora* CICLing-2002[1], R8[2] e os *corpora* do WSI SemEval [AS07] *apud* [IPR+08], compararam as medidas Λ-*Measure*, $\bar{\rho}$-*Measure*, Índice Dunn[3], Índice Davies-Bouldin e *Relative Hardness Measure*, buscando detectar quais índices apresentavam resultados semelhantes à avaliação humana por meio da *F-Measure*. Os autores demonstraram que as medidas $\bar{\rho}$-*Measure* e *Relative Hardness Measure* apresentaram resultados muito semelhantes à avaliação humana. Ressaltaram, porém, que os *corpora* se caracterizam por conterem textos pequenos e que sua avaliação não deve ser estendida, sem maiores investigações, a contextos diferentes.

Embora os experimentos de Ingaramo, Pinto, Rosso e Errecalde [IPR+08], tratem de documentos pequenos e os documentos de nosso exemplo de uso sejam mais extensos, o pré-processamento utilizado, conforme descrito na Seção 4.4.2, realizou grande redução da dimensionalidade dos atributos, obtendo, por exemplo, um vocabulário médio de 22,65 termos/referências legislativas por documento, enquanto que no WSI SemEval o vocabulário médio é de 47,65 palavras.

Assim, por apresentarem resultados que se aproximam bastante de resultados obtidos por avaliação humana e por considerar-se que há aplicabilidade destas medidas em nosso contexto, optamos por avaliar os agrupamentos obtidos através do cálculo das medidas $\bar{\rho}$-*Measure* e *Relative Hardness Measure*.

Tabela 5.2 – Medidas internas aferidas em cada agrupamento

Alg.	Descarte		Divisão		RH		$\bar{\rho}$-*Measure*		$\Delta\bar{\%}$
	Doc.	Grupo	Expl.	Impl.	Abs.	Δ%	Abs.	Δ%	
1	✔	✔			0.089		0.99		
2		✔			0.088	↑ 1,2%	1.97	↑ 100,14%	↑ **50,67%**
3	✔				0.063	↑ 29,21%	1.38	↑ 39,39%	↑ 34,3%
4	✔	✔	✔		0.082	↑ 7,87%	0.43	↓ 46,57%	↓ 19,35%
5			✔		0.046	↑ 48,31%	1.30	↑ 31,31%	↑ 39,81%
6				✔	**0.055**	↑ **38,2%**	**1.79**	↑ **80,81%**	↑ 59,51%

A Tabela 5.2 apresenta um comparativo das medidas aferidas. As medidas do algoritmo original foram destacadas em azul. Para cada medida, reportamos o valor absoluto aferido e a sua variação percentual em relação ao algoritmo original. A seta aponta para cima em caso de melhoria e para baixo em caso contrário. A melhor performance foi destacada em vermelho e a segunda melhor performance, em negrito.

Os algoritmos 2 e 5 obtiveram a melhor performance em uma das medidas. Mas, obtiveram performances muito baixas em outra das medidas. A última coluna da Tabela 5.2 apresenta uma média dos percentuais de variação e indica que o algoritmo 6, segunda melhor performance nas duas aferições, teve o melhor desempenho médio. Além disto,

[1] Composto de 48 resumos dos artigos apresentados na Conferência CICLing 2002.

[2] Um *subset* do Reuters-21578, disponível em http://www.daviddlewis.com/resources/testcollections/reuters21578/ .

[3] Conforme adaptação de Bezdek [BLA+97].

apenas os algoritmos 5 e 6 não realizam descartes, que, conforme elencado na Seção 4.2, é uma característica desejada num sistema de pesquisa jurisprudencial. Por estas razões, selecionamos o algoritmo 6 para aprofundar nossos estudos.

Para verificar se o resultado da aferição representa melhoria significativa particionamos aleatoriamente o conjunto de treino em 8 conjuntos disjuntos, 4 conjuntos contendo 86 documentos e 4 contendo 85 documentos. Não foi possível dividir em maior quantidade de conjuntos pois o algoritmo de Aggarwal, Gates e Yu [AGY04], em sua forma original, realiza grande quantidade de descartes e, por reduzir-se o tamanho do conjunto, a divisão em mais de 8 conjuntos implicou em descarte de 100% dos documentos na maioria dos conjuntos. Os demais conjuntos terminavam com um único grupo e, assim, também não era possível realizar o cálculo de qualquer medida, pois:

1. no cálculo da medida *Relative Hardness*, onde n é a quantidade de categorias, temos uma divisão por zero em vista da expressão $n \times (n-1)$ no denominador e;

2. no cálculo da medida $\bar{\rho}$, para $k = 1$ classe, temos

$$C = C_i \therefore |C| = |C_i| \therefore \frac{|C_i|}{|C|} = 1$$

então simplificamos o cálculo da medida para

$$\bar{\rho} = \sum_{i=1}^{k} \frac{w(C_i)}{|C_i|^{\theta}}$$

e

$$w(C) = |C|^{\theta} \therefore w(C_i) = |C_i|^{\theta}$$

logo

$$\bar{\rho} = \sum_{i=1}^{k} \frac{w(C_i)}{|C_i|^{\theta}} \therefore \bar{\rho} = \sum_{i=1}^{k} \frac{w(C_i)}{w(C_i)} \therefore \bar{\rho} = \sum_{i=1}^{k} 1$$

assim

$$\bar{\rho} = 1$$

para quaisquer documentos no grupo, independentemente da similaridade entre eles. Deixa de fazer sentido uma medida de densidade independizada da similaridade entre as instâncias.

Também não foi possível reduzir ainda mais o limiar de similaridade pois o passo de aglomeração do algoritmo acabava por unir todos os grupos num único grupo. Na iteração seguinte, com o centróide recalculado, a maioria dos documentos era descartada. Para alguns dos conjuntos, restava um único grupo ao final da iterações e para outros, após o descarte de documentos, a quantidade de documentos restantes estava abaixo do limiar de descarte de grupos, implicando no descarte do último grupo.

Tabela 5.3 – *Sign Test* para *Relative Hardness*

Alg.	Partição								Total
	1	2	3	4	5	6	7	8	
1	0,067	0,085	0,078	0,089	0,084	0,069	0,085	0,086	0
6	**0,045**	**0,053**	**0,058**	**0,058**	**0,052**	**0,053**	**0,057**	**0,050**	8

Tabela 5.4 – *Sign Test* para $\bar{\rho}$-$Measure$

Alg.	Partição								Total
	1	2	3	4	5	6	7	8	
1	0,111	0,097	0,119	0,118	0,093	0,079	0,117	0,103	0
6	**0,159**	**0,137**	**0,153**	**0,170**	**0,146**	**0,143**	**0,157**	**0,148**	8

Para cada partição executamos novo agrupamento utilizando o algoritmo original de Aggarwal, Gates e Yu [AGY04] e a evolução proposta pelo algoritmo 6, onde não se descartam documentos nem grupos e realiza-se a divisão implícita de grupos. Estabeleceram-se as hipóteses nulas $H_{0_{RH}} : RH_1 = RH_6$ e $H_{0_{\bar{\rho}}} : \bar{\rho}_1 = \bar{\rho}_6$, onde RH é a medida *Relative Hardness*, $\bar{\rho}$ é a Medida Esperada de Densidade e RH_i e $\bar{\rho}_i$ são as aferições das respectivas medidas em relação ao i-ésimo algoritmo.

Conforme as Tabelas 5.3 e 5.4, as evoluções que propusemos ao algoritmo de Aggarwal, Gates e Yu [AGY04], usando a variante que realiza a divisão implícita, superaram o algoritmo original em todas as partições tanto pela medida *Relative Hardness* quanto pela medida $\bar{\rho}$-$Measure$. Para que se possa considerar que a melhoria de performance seja significativa com 5% de confiança, o teste de sinal [She04, Sal97] *apud* [Dem06] utilizado exige que, nas 8 medições, o algoritmo proposto obtenha, no mínimo, 7 vitórias.

5.3.2 Análise da Classificação

Para validar a classificação dos documentos do conjunto de teste, submetemos os resultados ao exame de especialista humano com experiência em pesquisa e classificação de documentos jurídicos, atuante no Ministério Público Federal.

Para tanto foi desenvolvido um programa apresentado na Figura 5.1, onde o especialista visualiza duas colunas: a da esquerda apresenta o inteiro teor do documento classificado e a da direita apresenta o inteiro teor dos documentos que compõem o grupo que gerou a classe correspondente. Durante o processo de agrupamento, foram gerados rótulos para os grupos. No entanto, estes rótulos foram gerados para depuração durante o desenvolvimento dos programas e não foram apresentados à especialista. Uma vez que o objetivo é realizar uma pesquisa de documentos utilizando as metodologias de aprendizado de máquina, a avaliação da especialista deve ser focada no resultado final sob o ponto de vista de usuário. Ou seja, se o usuário pretende obter documentos, a especialista deve avaliar se os documentos recuperados são úteis ou não para o usuário.

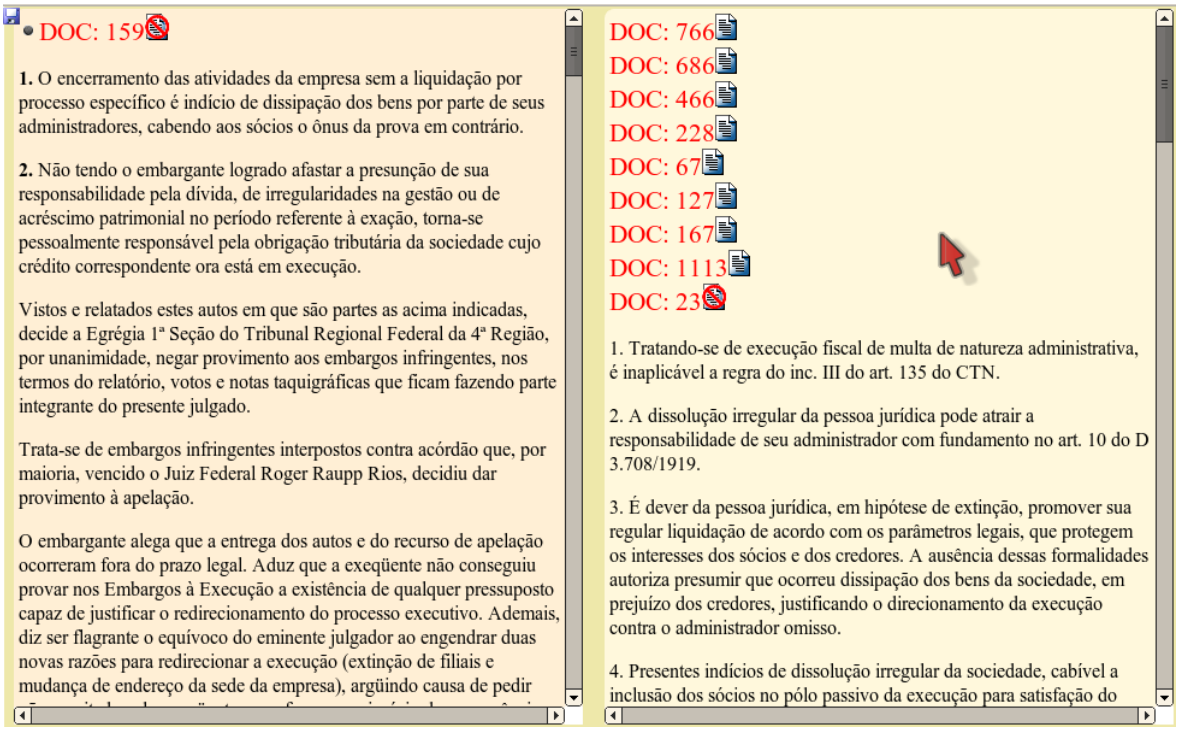


Figura 5.1 – Ferramenta de Validação da Categorização

O programa de validação permite, apenas, que a especialista indique se o documento foi bem ou mal classificado. A especialista foi orientada a considerar que o documento classificado, visualizado na coluna da esquerda, seja um processo jurídico em andamento e que os documentos visualizados na coluna da direita seriam os retornados por um aplicativo de pesquisa de documentos. Assim, para cada classificação, a especialista foi orientada a "verificar se os resultados da pesquisa continham, absolutamente, toda a informação necessária para que o jurista faça suas referências à jurisprudência quando redigir sua argumentação, dispensando, portanto, a realização de novas pesquisas na jurisprudência, marcando a pesquisa com um sinal de ✓. Caso contrário, a pesquisa deve ser marcada com um sinal de ✗".

Ao final da validação, 136 documentos (57%) foram considerados verdadeiros positivos (VP) e 100 documentos (42%) foram considerados falsos positivos (FP) pela especialista. Note-se que, na fase de agrupamento, o limiar de similaridade utilizado para associar um documento a um grupo foi de 0,5 (50%). Já na fase de classificação, não existe limite mínimo de similaridade. Os documentos foram todos categorizados em alguma das classes geradas pelo agrupamento. A Figura 5.2 apresenta a quantidade de verdadeiros positivos (VP) e de falsos positivos (FP) tabulados em faixas de similaridade, iniciando pelos categorizados com mais de 50% de similaridade com a classe, seguidos de faixas de Δ5% até um mínimo de 10% de similaridade.

Para melhor analisar a validação da categorização, foram obtidos de cada classificação, os seguintes indicadores:

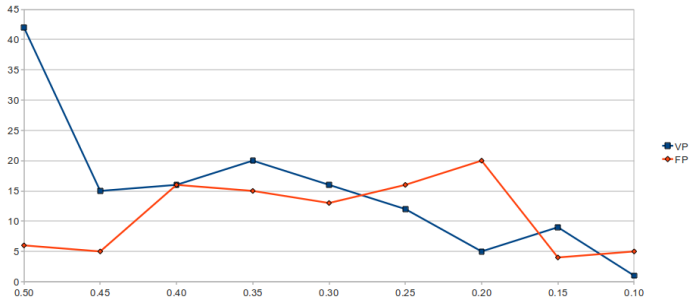


Figura 5.2 – Gráfico da Validação por especialista

1. **simcateg**: similaridade entre o documento e a classe;
2. **qtdoc**: quantidade de documentos no grupo correspondente à classe;
3. **coesao**: coesão do grupo correspondente à classe;
4. **simmean**: similaridade média dos documentos do grupo correspondente à classe;
5. **maxsim**: similaridade do documento de maior similaridade com o centróide do grupo correspondente à classe;
6. **minsim**: similaridade do documento de menor similaridade com o centróide do grupo correspondente à classe;
7. **qtattseed**: quantidade de atributos no centróide do grupo correspondente à classe;
8. **qtattdocs**: soma da quantidade de atributos nos documentos do grupo correspondente à classe;
9. **maxattdoc**: quantidade de atributos do documento com mais atributos no grupo correspondente à classe;
10. **minattdoc**: quantidade de atributos do documento com menos atributos no grupo correspondente à classe;
11. **meanattdoc**: média de atributos nos documentos do grupo correspondente à classe;
12. **maxngram**: quantidade de palavras do termo com o maior número de palavras no grupo correspondente à classe;
13. **minngram**: quantidade de palavras do termo com o menor número de palavras no grupo correspondente à classe;
14. **meanngram**: quantidade média de palavras dos termos no grupo correspondente à classe;

15. **qtterm**: quantidade de atributos originados de termos dos vocabulários jurídicos do grupo correspondente à classe;
16. **qtrefleg**: quantidade de atributos originados de referências legislativas do grupo correspondente à classe;
17. **qtmerge**: quantidade de uniões de grupos realizadas no grupo correspondente à classe;
18. **maxattcomm**: maior quantidade de atributos em comum entre os documentos e o centróide do grupo correspondente à classe;
19. **minattcomm**: menor quantidade de atributos em comum entre os documentos e o centróide do grupo correspondente à classe;
20. **meanattcomm**: quantidade média de atributos em comum entre os documentos e o centróide do grupo correspondente à classe;

Não foi detectada nenhuma evidência de relação entre estes atributos e o sucesso/falha na classificação, exceto por uma fraca relação com a similaridade entre o documento classificado e a classe, já evidenciada na Figura 5.2.

5.4 Informação Não Extraída dos Documentos

Após avaliação dos dados obtidos pelos após a execução dos exemplos de uso, percebeu-se que o grau de similaridade dos documentos categorizados com os centróides dos grupos geradores das classes apresenta algum nível de relação com o sucesso/falha da classificação. No entanto, esta relação é insuficiente para explicar satisfatoriamente os resultados da classificação.

Assim, procedemos a uma revisão mais detalhada dos casos extremos. Analisamos, então, os 11 documentos erroneamente classificados cuja similaridade com os centróides supera 45%. Analisamos, também, os 14 documentos corretamente classificados cuja similaridade com os centróides é inferior a 25%.

5.4.1 Falsos Positivos com Alta Similaridade

Percebe-se que uma combinação de dois fatores muito contribuiu para a incidência dos falsos positivos estudados: atributos com alta freqüência e atributos com semântica demasiadamente genérica.

O documento 554, por exemplo, foi classificado na classe correspondente ao grupo 15449, rotulado como “crime”. Tem como atributos os listados na Tabela 5.5. Percebe-se a predominância do atributo “crime”: quase o triplo do segundo atributo mais freqüente e

mais que o triplo do terceiro atributo. Além deste, os atributos “código penal”, “justiça de o trabalho”, “multa”, “legislação penal” e “circunstância atenuante” são, também, demasiadamente genéricos. Nota-se, também, que não há atributos originados de referências legislativas. O grupo é composto de 5 documentos, cujos temas podem ser vistos na Tabela 5.6. Percebe-se que, em verdade, não há identificação de temas entre quaisquer dois documentos do grupo. Além disto, não se pode falar em sanar este problema aumentando a quantidade de iterações do algoritmo na expectativa de que o passo de projeção elimine mais atributos do centróide, pois os atributos eliminados seriam os de maior especificidade semântica.

Tabela 5.5 – Atributos do Grupo “Crime”

Atr.	Peso	Atr.	Peso
crime	0,546	sanção	0,205
código penal	0,173	salário mínimo	0,171
servidor público	0,169	prestação de serviço a o comunidade	0,157
peculato	0,117	caixa econômico	0,116
justiça de o trabalho	0,101	multa	0,090
liberdade	0,089	passaporte	0,075
correspondência	0,075	falsidade ideológico	0,072
polícia federal	0,065	vítima	0,064
legislação penal	0,063	falsificação	0,060
órgão público	0,054	empregado	0,054
certidão de nascimento	0,054	circunstância atenuante	0,053
decreto executivo	0,049	administração público	0,046

A título de comparação, o oposto ocorre com a classificação do documento 1039 no grupo 15447, rotulado como “estação de rádio”. A classificação ocorreu com similaridade de 72%, a mais alta dentre as classificações em nosso exemplo de uso. Tanto o documento classificado como os 3 documentos agrupados versam sobre atraso na autorização para operação de emissora de rádio. Ao observar-se os atributos do centróide, percebe-se que os atributos de maior peso têm alta especificidade semântica. Além disto, tanto os documentos agrupados, como o documento classificado têm pelo menos um atributo não nulo em comum com o centróide originado de referência legislativa não genérica[4].

O grupo 19018, rotulado como “dano && indenização”, é resultante de uma divisão implícita, que iniciou-se com o documento 61 na segunda iteração, recebendo mais 3 documentos nas duas última iterações. Este grupo tem como tema a indenização por danos morais. Durante o teste de classificação, 5 documentos foram categorizados na classe correspondente a este grupo. Destes, 3 documentos são verdadeiros positivos, 2 são falsos positivos. Dos falsos positivos, um foi categorizado com baixa similaridade, 29,57%, tendo apenas um atributo não nulo em comum com o centróide, “indenização”; o outro documento

[4]Por referência legislativa genérica, entenda-se uma referência a uma legislação ampla, como a Constituição Federal ou os Códigos Civil e Penal, sem especificar um artigo.

Tabela 5.6 – Temas do Grupo “Crime”

Principais Atributos	Tema
crime, peculato, prestação de serviço a o comunidade, código penal, sanção, administração público.	Abuso dos poderes do cargo para trocar bem de sua propriedade por outro, de qualidade superior, pertencente ao patrimônio de órgão público.
crime, sanção, salário mínimo, legislação penal, multa, vítima, código penal.	Correção da dosimetria da pena por fragilidade de provas.
crime, falsidade ideológico, justiça de o trabalho, falsificação, prestação de serviço a o comunidade, salário mínimo, sanção, código penal.	Falsificação de documentos para eximir-se de obrigações trabalhistas.
servidor público, caixa econômico, correspondência, crime, justiça de o trabalho.	Ocultação de documento público com prejuízo de parte contrária em ação trabalhista.
crime, passaporte, polícia federal, certidão de nascimento, circunstância atenuante, sanção.	Falsificação de documentos para a obtenção de passaporte.

foi categorizado com similaridade mais alta, 48,3%, apesar de ter somente dois atributos não nulos em comum com o centróide. Estes dois atributos, “dano” e “indenização”, conforme a Tabela 5.8, que apresenta os atributos do centróide, são decisivos na determinação da similaridade com o centróide. O documento foi mal classificado, embora com maior similaridade, porque tratava-se de um recurso acerca da discussão do valor da causa. Causa esta que clamava danos morais. A presença de trechos de texto da ação que originou este recurso, que não versa sobre danos morais e sim sobre valor da causa originária, acabou por determinar a similaridade com este centróide.

Já o documento 979 foi erroneamente categorizado, com similaridade de 53,3%, na classe correspondente ao grupo 19116, rotulado como "crédito tributário && multa", resultante de divisão implícita. Este grupo contêm apenas um documento e somente três atributos em seu centróide, listados na Tabela 5.9. Dos atributos não nulos do documento, somente “multa” e “crédito tributário” também não são nulos no centróide. A escassez de atributos, agravada pelo fato de o atributo “multa” ser demasiadamente genérico, acabou por determinar a errônea categorização deste documento.

Tabela 5.7 – Atributos do Grupo “estação de rádio”

Atr.	Peso	Atr.	Peso
estação de rádio	0,529	radiodifusão	0,415
processo administrativo	0,276	poder judiciário	0,179
risco	0,136	administração	0,131
mora	0,126	associação	0,092
l9612/1998	0,086	poder executivo	0,086
estupro	0,076	direito e garantia individual	0,074
administração público	0,072	ec45/2004	0,057
abuso de poder	0,049	l9472/1997	0,046
empresa público	0,046	cf/1988	0,034
decreto executivo	0,032	porto	0,030
tutela	0,030	crime por omissão	0,019
ação ordinário	0,019	l9784/1999	0,019

Tabela 5.8 – Atributos do Grupo “dano && indenização”

Atr.	Peso	Atr.	Peso
dano	0,882	indenização	0,344
reparação de dano	0,071	má-fé	0,046
vítima	0,044	responsabilidade civil	0,036
direito humano	0,025	princípio da razoabilidade	0,021
processo administrativo	0,021		

5.5 Verdadeiros Positivos com Baixa Similaridade

Foram avaliadas as classificações que, embora tenham sido corretas, tiveram muito baixa similaridade entre o documento e o centróide do grupo correspondente. Foi observado que os centróides e estes documentos tinham poucos atributos não nulos em comum.

O documento 989, por exemplo, classificado com 19,5% de similaridade na classe correspondente ao grupo 15556, rotulado como “l9289/1996 art. 7”, tem apenas dois atributos não nulos em comum com o centróide deste grupo: “contador” e “renda”, ambos muito genéricos.

O documento 769, classificado com 19,3% de similaridade na classe correspondente ao grupo 19067, rotulado como “período de carência && ação ordinário”, tem apenas dois atributos não nulos em comum com o centróide deste grupo: “renda” e “ação ordinário”, ambos muito genéricos.

5.6 Possíveis Soluções

5.6.1 Problema dos Centróides com Poucos Atributos Não Nulos

Quanto ao problema de centróides com poucos atributos não nulos, observamos que o passo de atribuição de documentos do algoritmo de Aggarwal, Gates e Yu [AGY04] não

Tabela 5.9 – Atributos do Grupo “crédito tributário && multa”

Atr.	Peso	Atr.	Peso
crédito tributário	0,784	multa	0,588
inadimplemento	0,196		

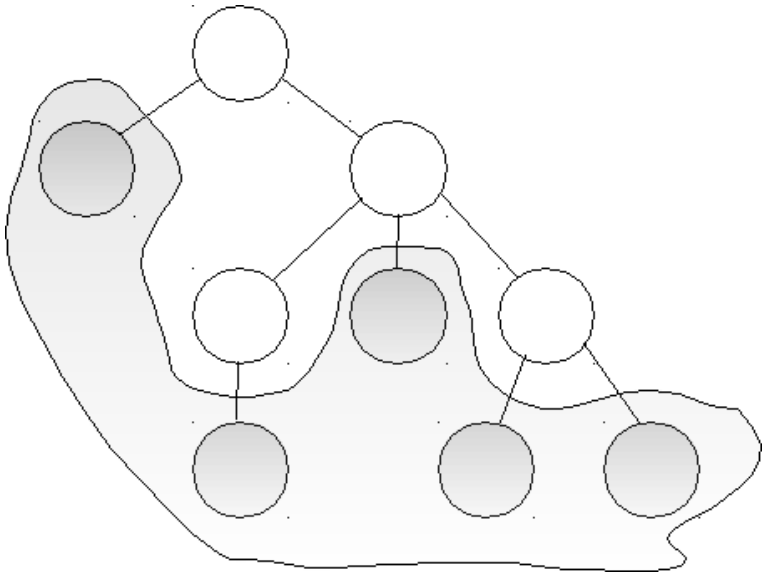


Figura 5.3 – Reconhecimento exclusivo dos termos mais específicos

prevê acréscimo de atributos não nulos aos centróides. Isto ocorre porque no experimento descrito em seu artigo, os autores trabalharam com conjunto de dados onde todo documento apresentava grande quantidade de atributos e, assim, os centróides sempre dispunham da quantidade máxima de atributos permitida para a iteração. Em nosso exemplo de uso, o emprego exclusivo de termos jurídicos e referências legislativas como atributo reduziu significativamente a dimensionalidade dos centróides. Desta maneira, uma alteração no passo de projeção, provendo esta inclusão de novos atributos não nulos no centróide, poderá melhorar a sua qualidade.

5.6.2 Problema dos Atributos com Semântica Muito Genérica

Em relação ao problema gerado por atributos de semântica muito genérica com maior peso nos centróides que os de semântica mais específica, elencamos algumas alternativas, descritas nas Seções 5.6.2.1, 5.6.2.2 e 5.6.2.3.

5.6.2.1 Descarte de Nodos Não Terminais

A maneira mais simples de evitar que atributos de semântica muito genérica ganhem demasiada relevância nos centróides dos grupos, conseqüentemente agrupando documentos com temática diversa, como ocorreu com o grupo 15449 – “crime”, é descartar estes atributos.

Embora tenhamos descartado as informações de hierarquia dos termos dos tesauros utilizados, esta poderia ser utilizada para selecionar somente os termos mais específicos, ou seja, apenas os nodos folha, como ilustrado na Figura 5.3, seriam reconhecidos no pré-processamento dos documentos.

5.6.2.2 Atribuição de Pesos aos Termos

Um possível problema que poderá surgir se adotada a alternativa apresentada na Seção 5.6.2.1, é que, eventualmente, algum documento fique sem atributos. Outro problema que pode ocorrer é que documentos que tratem de assuntos semelhantes apresentem termos distintos, porém filhos de um mesmo nodo-pai, também presente no texto, mas descartadao, não sejam reconhecidos como mais similares entre si que documentos de assuntos "muito distantes", ou seja, de grande distância entre os respectivos nodos.

Uma forma de lidar com este problema seria permanecer reconhecendo os termos genéricos, mas, atribuir pesos de acordo com o nível de especificidade. Se o peso semântico for um inteiro indicando o nível de profundidade na hierarquia, ao somá-lo ao atributo, que se insere no intervalo $(0; 1]$, estaremos garantindo, em qualquer caso, que termos mais específicos ganhem mais relevância no cálculo de similaridade. Além disto, no passo de projeção, o descarte de atributos selecionará sempre os atributos mais genéricos, em detrimento dos mais específicos. A Tabela 5.10 apresenta um exemplo de como o peso semântico poderia influir na relevância dos atributos do Grupo "dano && indenização".

Tabela 5.10 – Atributos do Grupo "dano && indenização"

Atr.	Peso	Atr.	P. Sem.	P. Final
dano	0,882	direito humano	4	4,025
indenização	0,344	responsabilidade civil	3	3,036
reparação de dano	0,071	dano	2	2,882
má-fé	0,046	indenização	2	2,344
vítima	0,044	reparação de dano	2	2,071
responsabilidade civil	0,036	má-fé	2	0,046
direito humano	0,025	vítima	2	0,044
princípio da razoabil.	0,021	princípio da razoabil.	2	2,021
processo administrativo	0,021	processo administrativo	2	2,021

5.6.2.3 Agrupamento Hierárquico

A exemplo do trabalho de Toutanova *et al.* [TCP+01], revisado na Seção 3.2.2, poderia ser implementado um algoritmo de agrupamento hierárquico onde cada nodo da árvore representaria uma classe, e os termos jurídicos e referências legislativas estariam relacionados com níveis diferentes desta árvore, conforme suas respectivas especificidades. Dife-

rentemente do experimento de Toutanova *et al.* [TCP+01], o nível dos atributos não é desconhecido, mas obtido conforme referenciado na Seção 5.6.2.2 e detalhado na Seção 5.7.

5.6.3 Atualização dos Tesauros

Se observarmos a Tabela 5.7, veremos que ali consta o termo estupro. Considerado o contexto dos demais termos, percebe-se que tal termo não parece ter a menor relação com os demais, sugerindo a possibilidade de problemas na extração dos termos. Em exame mais detalhado dos documentos, encontramos, no documento 636, as expressões "violação ao princípio da eficiência e da razoabilidade" e "VIOLAÇÃO AOS ARTIGOS 6º DA LEI 9612/98 E 9º, INCISO II, DO DECRETO 2615/98". O termo "violação", como unigrama, é sinônimo do termo "estupro". Expressões como "violação de/do/a/ao direito/referência legislativa" são comuns, mas não se encontram em nenhum dos vocabulários utilizados. Estes vocabulários contêm expressões específicas, como "violação de direito autoral", ou "violação de direito de propriedade" e, assim, são reconhecíveis como tal e não como sinônimo de estupro. No entanto, como as expressões utilizadas não existiam nos tesauros, o pré-processamento reconheceu apenas o unigrama "violação", que foi normalizado para estupro.

Tanto Sordi [SMS+07] quanto Jaegger [JAS+07] ressaltam que seus tesauros estavam incompletos. Tal defasagem agrava-se, tendo em vista o tempo transcorrido desde a época de sua publicação até o presente.

Há diversas iniciativas de geração automatizada de tesauros e ontologias. Em português encontramos a ferramenta ExatoLP de Lopes *et al.* [LFV+09], que automatiza a criação de estruturas ontológicas e poderia auxiliar neste processo.

5.6.4 Agrupamento Semi-supervisionado por Referências Legislativas

Uma outra possibilidade seria adaptar o algoritmo de agrupamento para usar uma função de similaridade que considerasse, tão somente, os atributos com origem em referências legislativas. Desta maneira, garantir-se-ía que cada grupo contivesse somente documentos que referenciassem as mesmas normas.

Alguns problemas devem, no entanto, ser levantados:

1. As partes dos processos, podem, eventualmente, não realizar pesquisa jurisprudencial suficientemente abrangente para determinar que normas se aplicam ao caso em questão e, conseqüentemente, os documentos por elas anexados ao processo jurídico não conterão referências legislativas que gerem atributos necessários à boa classificação. Tal problema, no entanto, pode ser minimizado utilizando-se uma função de similaridade diferente para a classificação, onde seriam considerados todos ou, pelo menos, alguns atributos com origem em termos jurídicos;

2. Os documentos podem conter, apenas, referências legislativas muito genéricas, como a Constituição Federal;

3. As referências legislativas podem ser redigidas de forma que não sejam detectadas pelo parser. Exemplos disto não faltam: “Magna Carta”, “Lei Orgânica da Magistratura”, “Lei da Mordaça”, “Lei Maria da Penha”, “Estatuto da Criança e do Adolescente”, etc. Seria necessário, portanto, construir-se um dicionário de nomes populares de normas jurídicas.

5.7 Atribuição de Pesos Semânticos aos Termos e referências Legislativas

Dentre as alternativas elencadas nas seções anteriores, descartamos a opção da Seção 5.6.2.1 pelas razões apresentadas na Seção 5.6.2.2. As alternativas das demais seções demandariam muito mais tempo do que dispúnhamos para serem implementadas. Desta maneira, optamos por concentramo-nos na opção da Seção 5.6.2.2.

Atribuímos, então, um peso semântico a cada termo extraído dos vocabulários jurídicos. Este peso é um inteiro indicando o nível de profundidade na hierarquia que foi somado ao atributo. Para obter o nível de profundidade, retornamos aos tesauros. No caso do TJF, a relação hierárquica é uma informação completa, ou seja, todos os termos apresentam a indicação de todos os seus hiperônimos e a sua distância dos mesmos[5] e, assim, bastou selecionar o maior n dos respectivos TGn de cada termo e usá-lo como peso do termo. No caso do VCB, a estrutura de árvore está fragmentada e há muitos termos específicos sem indicação de hiperônimo. No entanto, a maioria dos termos do VCB apresenta a Classificação Decimal de Direito (CDD), composta por um número de 3 dígitos que pode ser seguido de um ponto e um número variável de dígitos. A quantidade de dígitos após o ponto indica o grau de especificidade do termo e foi usada como peso do termo. Restaram 2016 termos do VCB que não apresentavam o CDD e, assim, receberam o peso mínimo, ou seja, zero.

Quanto às referências legislativas, atribuiu-se peso quatro às referências que não continham especificação de artigo e peso seis a qualquer referência legislativa acompanhada de especificação de artigo. As únicas normas sem referência de artigo que não receberam peso 4 foram os Códigos Civil e Penal, por serem leis extensas, que receberam, então, peso 3 e as Constituições Federal e estaduais em virtude de suas abrangências, incluindo matérias cíveis e penais, que receberam peso dois.

Determinados os pesos para os atributos, realizou-se nova extração de termos e referências legislativas dos documentos, gerando os respectivos vetores de atributos. As Seções 5.8 e 5.9, a seguir, descrevem as análises dos resultados da execução dos agrupamentos e da classificação destes novos vetores com pesos semânticos.

[5] TG1 para o termo no nodo pai, TG2 para o avô, etc.

Tabela 5.11 – Novas Medidas internas aferidas em cada agrupamento

Alg.	Descarte		Divisão		RH		$\bar{\rho}$-*Measure*		$\Delta\bar{\%}$
	Doc.	Grupo	Expl.	Impl.	Abs.	$\Delta\%$	Abs.	$\Delta\%$	
1	✔	✔			0.071		1,01		
2		✔			0.058	↑ 17,50%	0,77	↓ 24,48%	↓ 3,49%
3	✔				0.032	↑ 53,95%	**0,90**	↓ **11,56%**	↑ 21,20%
4	✔	✔	✔		0.065	↑ 8,36%	1,03	↑ **1,62%**	↑ 4,99%
5			✔		**0.033**	↑ **52,76%**	0,81	↓ 20,51%	↑ 16,13%
6				✔	0.035	↑ 50,19%	0,89	↓ 12,20%	↑ **19,00%**

5.8 Nova Análise dos Agrupamentos

Obtidos os novos vetores de atributos, agora com pesos semânticos, realizou-se nova rodada de execuções dos vários algoritmos de agrupamento, conforme a Tabela 5.1. O limiar de similaridade teve de ser reduzido para 40% porque com o aumento de atributos os documentos tornaram-se mais distintos, reduzindo a similaridade e, assim, novamente o algoritmo original de Aggarwal, Gates e Yu [AGY04] descartava muitos documentos e, posteriormente, descartava grupos. Mas, com esta redução, os centróides foram considerados muito similares e todos os grupos foram aglomerados num único grupo na primeira iteração. Na segunda iteração, após o recálculo do centróide, a maioria dos documentos era descartada pois ficaram muito distantes do centróide e, em seguida, o grupo era descartado em função do limiar de descarte de grupos. A solução foi iniciar as iterações com, no máximo, 70 atributos nos centróides, o que garantiu diferenciação a eles, impedindo que se unissem num único grupo. Isto acarretou numa diminuição da quantidade de iterações, o que não garantia um bom refinamento dos grupos e, assim, fixou-se o mínimo de 15 atributos para encerramento das iterações.

Apesar do algoritmo 6 ter sido escolhido após a avaliação dos agrupamentos na fase de teste, decidiu-se por realizar novas comparações entre eles para averiguar se as mudanças realizadas implicariam em resultados significativamente distintos. Assim, a Tabela 5.11 apresenta um comparativo das novas aferições. O *layout* desta tabela segue o padrão da Tabela 5.2.

Os algoritmos 2 e 3 obtiveram a melhor performance em uma das medidas. O algoritmo 3 teve o melhor desempenho médio. O algoritmo 6 teve o segundo melhor desempenho médio, perdendo para o algoritmo 3 por, apenas, 2,20%.

Para verificar se o resultado da aferição representa superioridade significativa do algoritmo 3 sobre o algoritmo 6 em relação às medidas RH e $\bar{\rho}$, particionamos aleatoriamente o conjunto de treino em 8 conjuntos disjuntos, 4 conjuntos contendo 90 documentos e 4 contendo 89 documentos, repetindo os agrupamentos segundo os dois algoritmos em cada uma das partições. Estabeleceram-se as hipóteses nulas $H_{0_{RH}} : RH_3 = RH_6$ e $H_{0_{\bar{\rho}}} : \bar{\rho}_3 = \bar{\rho}_6$.

Realizamos o *Sign Test* para *Relative Hardness* entre os algoritmos 3 e 6, conforme

Tabela 5.12 – *Sign Test* para *Relative Hardness* entre os algoritmos 3 e 6

Alg.	Partição								Total
	1	2	3	4	5	6	7	8	
3	0,0378	**0,0302**	0,0352	**0,0345**	**0,0374**	**0,0362**	**0,0360**	0,0334	5
6	**0,0368**	0,0307	**0,0345**	0,0365	0,0416	0,0367	0,0374	**0,0316**	3

Tabela 5.13 – *Ranks* de *Relative Hardness* para o cálculo do *Wilcoxon Sign Test* entre os algoritmos 3 e 6

	Partição							
	1	2	3	4	5	6	7	8
3	0,0378	**0,0302**	0,0352	**0,0345**	**0,0374**	**0,0362**	**0,0360**	0,0334
6	**0,0368**	0,0307	**0,0345**	0,0365	0,0416	0,0367	0,0374	**0,0316**
Δ_i	0,0010	-0,0005	0,0007	-0,0020	-0,0042	-0,0005	-0,0014	0,0018
rank	4	1,5	3	7	8	1,5	5	6

apresentado na Tabela 5.12. Verificou-se que, não se pode rejeitar a hipótese nula $H_{0_{RH}}$: $RH_3 = RH_6$, ou seja, não há diferença significativa entre a performance dos dois algoritmos em relação à medida *Relative Hardness*. Procedeu-se, então ao cálculo do Wilcoxon *signed-ranks test*, cujos $ranks$ podem ser verificados na Tabela 5.13:

$$
\begin{aligned}
R_3 &= 1,5+7+8+1,5+5 = 23 \\
R_6 &= 4+3+6 = 13 \\
T &= min(R_3, R_6) = 13 \\
z &= \frac{13-\frac{8(8+1)}{4}}{\sqrt{\frac{8(8+1)(2\times 8+1)}{24}}} \\
z &= \frac{13-18}{\sqrt{\frac{1224}{24}}} \\
z &= \frac{-5}{\sqrt{51}} \\
z &\cong \frac{-5}{7,1414} \\
z &\cong -0,7
\end{aligned}
\tag{5.1}
$$

demonstrando que $z > -1,96$ e, portanto, o Wilcoxon *signed-ranks test* também não permite rejeitar a hipótese nula $H_{0_{RH}}$ e, assim, não se verifica superioridade significativa da performance do algoritmo 3 sobre o algoritmo 6, em relação à medida *Relative Hardness*.

Realizamos, então, o *Sign Test* para $\bar{\rho}$-$Measure$ entre os algoritmos 3 e 6, conforme

Tabela 5.14 – *Sign Test* para $\bar{\rho}$-$Measure$ entre os algoritmos 3 e 6

Alg.	Partição								Total
	1	2	3	4	5	6	7	8	
3	**1,00**	**1,00**	**1,00**	**1,00**	**0,99**	**1,00**	**1,00**	**1,00**	8
6	0,92	0,95	0,92	0,91	0,91	0,92	0,90	0,98	0

Tabela 5.15 – *Sign Test* para *Relative Hardness*

Alg.	Partição								Total
	1	2	3	4	5	6	7	8	
1	0,047	0,037	0,056	0,054	0,054	0,044	0,056	**0,030**	1
6	**0,037**	**0,031**	**0,035**	**0,036**	**0,042**	**0,037**	**0,037**	0,032	7

apresentado na Tabela 5.14. Verificou-se que, pode-se rejeitar a hipótese nula $H_{0_{\bar{\rho}}} : \bar{\rho}_3 = \bar{\rho}_6$, ou seja, há diferença significativa entre a performance dos dois algoritmos em relação à medida $\bar{\rho}$-$Measure$.

Vê-se, portanto, que, há controvérsia entre os testes de significância em relação às duas medidas de qualidade interna dos agrupamentos obtidos com os algoritmos 3 e 6. Em vista disto, e por não realizar descartes, que, conforme elencado na Seção 4.2, é uma característica desejada num sistema de pesquisa jurisprudencial, selecionamos, novamente, o algoritmo 6 para aprofundar nossos estudos.

Passamos, então à comparação entre o algoritmo 1 e o algoritmo 6. Para verificar se o resultado da aferição representa melhoria significativa em relação à medida RH e se não representa piora significativa em relação à medida $\bar{\rho}$, realizamos o mesmo particionamento do conjunto de treino em 8 subconjuntos e repetimos os agrupamentos segundo os dois algoritmos em cada uma das partições. Estabeleceram-se as hipóteses nulas $H_{0_{RH}} : RH_1 = RH_6$ e $H_{0_{\bar{\rho}}} : \bar{\rho}_1 = \bar{\rho}_6$.

Para verificar se o algoritmo evoluído, usando a variante que realiza a divisão implícita, superou o algoritmo de Aggarwal, Gates e Yu [AGY04], realizamos o teste de contagem de vitórias e derrotas, tendo em vista que este é o teste que menos rejeita a hipótese nula. Conforme a Tabela 5.15, nosso algoritmo superou o original em 7 das 8 partições, permitindo rejeitar a hipótese nula e concluir que a performance do algoritmo evoluído é significativamente superior, com respeito à medida *Relative Hardness*.

Por outro lado, para aferir se o algoritmo de Aggarwal, Gates e Yu [AGY04] tem performance significativamente superior à performance de nosso algoritmo, usando a variante

Tabela 5.16 – *Sign Test* para $\bar{\rho}$-$Measure$

Alg.	Partição								Total
	1	2	3	4	5	6	7	8	
1	0,916	**0,983**	**0,957**	**0,953**	0,897	**0,924**	0,903	**1,042**	5
6	**0,922**	0,947	0,922	0,915	**0,912**	0,919	**0,904**	0,983	3

Tabela 5.17 – *Ranks* de $\bar{\rho}$-$Measure$ para o cálculo do *Wilcoxon Sign Test*

	Partição							
	1	2	3	4	5	6	7	8
Alg. 1	0,916	**0,983**	**0,957**	**0,953**	0,897	**0,924**	0,903	**1,042**
Alg. 6	**0,922**	0,947	0,922	0,915	**0,912**	0,919	**0,904**	0,983
Δ_i	-0,006	0,036	0,035	0,038	-0,015	0,005	-0,001	0,059
rank	3	6	5	7	4	2	1	8

que realiza a divisão implícita, realizamos dois testes. A Tabela 5.16, demonstra que, de acordo com o teste de contagem de vitórias e derrotas, não há superioridade significativa da performance do algoritmo de Aggarwal, Gates e Yu [AGY04] sobre o algoritmo evoluído, em relação à medida $\bar{\rho}$-$Measure$. A Tabela 5.17 apresenta os *ranks* das diferenças de performance obtidas em cada conjunto de dados, usados no cálculo do Wilcoxon *signed-ranks test*,

$$
\begin{aligned}
R_1 &= 6+5+7+2+8 = 28 \\
R_6 &= 3+4+1 = 8 \\
T &= min(R_1, R_6) = 8 \\
z &= \frac{8-\frac{8(8+1)}{4}}{\sqrt{\frac{8(8+1)(2\times 8+1)}{24}}} \\
z &= \frac{8-18}{\sqrt{\frac{1224}{24}}} \\
z &= \frac{-10}{\sqrt{51}} \\
z &\cong \frac{-10}{7.1414} \\
z &\cong -1.4
\end{aligned}
\tag{5.2}
$$

demonstrando que $z > -1,96$ e, portanto, o Wilcoxon *signed-ranks test* também não permite rejeitar a hipótese nula e, portanto, não se verifica superioridade significativa da performance do algoritmo de Aggarwal, Gates e Yu [AGY04] sobre o algoritmo evoluído, em relação à medida $\bar{\rho}$-$Measure$.

5.9 Nova Análise da Classificação

Conforme informado, em virtude de indisponibilidade de tempo, não era viável realizar a avaliação das 238 categorizações do documentos do conjunto de operação por especialista

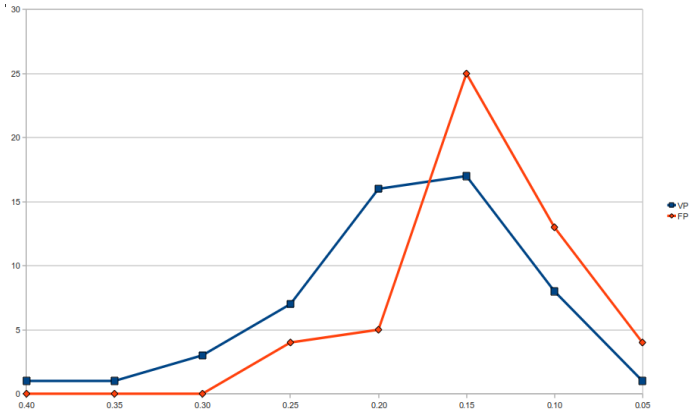


Figura 5.4 – Similaridade entre o documento e a classe - simcateg

humano. O tempo disponível era suficiente, apenas, para a avaliação de, aproximadamente, 50 categorizações. Assim, para obter a avaliação de ao menos 100 categorizações foi necessário combinar os esforços de dois especialistas humanos. O especialista humano que atuou na primeira validação da classificação, onde utilizamos o conjunto de teste, não participou desta segunda classificação.

Separamos, então, aleatoriamente 105 documentos do conjunto de operação, que foram divididos em 3 subconjuntos disjuntos. Os subconjuntos A e B contém 50 documentos e o subconjunto C contém 5 documentos. O especialista humano 1, cujo perfil detalhamos no Apêndice E, recebeu, para avaliação, as categorizações dos subconjuntos A e C. O especialista humano 2, cujo perfil detalhamos no Apêndice F, recebeu para avaliação, as categorizações dos subconjuntos B e C.

Utilizou-se o mesmo programa de validação e, ao final da validação, verificou-se que, dos 5 documentos que foram validados por ambos especialistas, apenas 1 foi objeto de discordância.

O Anexo A apresenta o inteiro teor do documento Nº 50, cuja classificação no grupo 26.936 é controversa. O Anexo B apresenta o inteiro teor do documento Nº 17, tido como o caso mais semelhante ao do documento Nº 50. Ambos possuem as seguintes características:

1. discussão entre o INSS e o segurado, que requerem a concessão de auxílio-doença ou mesmo aposentadoria por invalidez;

2. a condição de segurado e o período de carência foi devidamente provada no processo;

3. o fim da invalidez afigura-se permanente ou impossível de determinar.

Os casos divergem pelas seguintes características:

1. o INSS é vencedor no caso do documento Nº 17 e derrotado no caso do documento Nº 50;

2. difere o argumento da decisão judicial, no documento Nº 50 o segurado tinha doença congênita, portanto, não gozava de condições para o trabalho na data em que ingressou na condição de segurado; e, no caso do documento Nº 17, o juízo sequer discute a existência ou não da capacidade laborativa à epoca do ingresso na condição de segurado.

Ressaltando que a vitória ou derrota desta ou aquela parte no processo não é objeto tratado em nossa proposta de uso do aprendizado de máquina na pesquisa jurisprudencial e que, desta maneira, focamos apenas em averiguar se há identificação temática entre o documento ora classificado e os documentos do grupo correspondente a esta classe. A divergência, portanto, se resume a um ponto: a razão sobre a qual se embasa a decisão do juiz. Não nos cabe, todavia, entrar no mérito do entendimento dos especialistas humanos. Cabe, no entanto, decidir a respeito da divergência de avaliação. No caso específico, entendemos que a comprovação da validade da classificação realizada em nosso exemplo de uso deve manter-se incontroversa. Havendo dúvida, deve ser reputada como falha e, portanto, 53 documentos (50,5%) foram considerados verdadeiros positivos (VP) e 52 documentos (49,5%) foram considerados falsos positivos (FP) por ao menos um especialista humano.

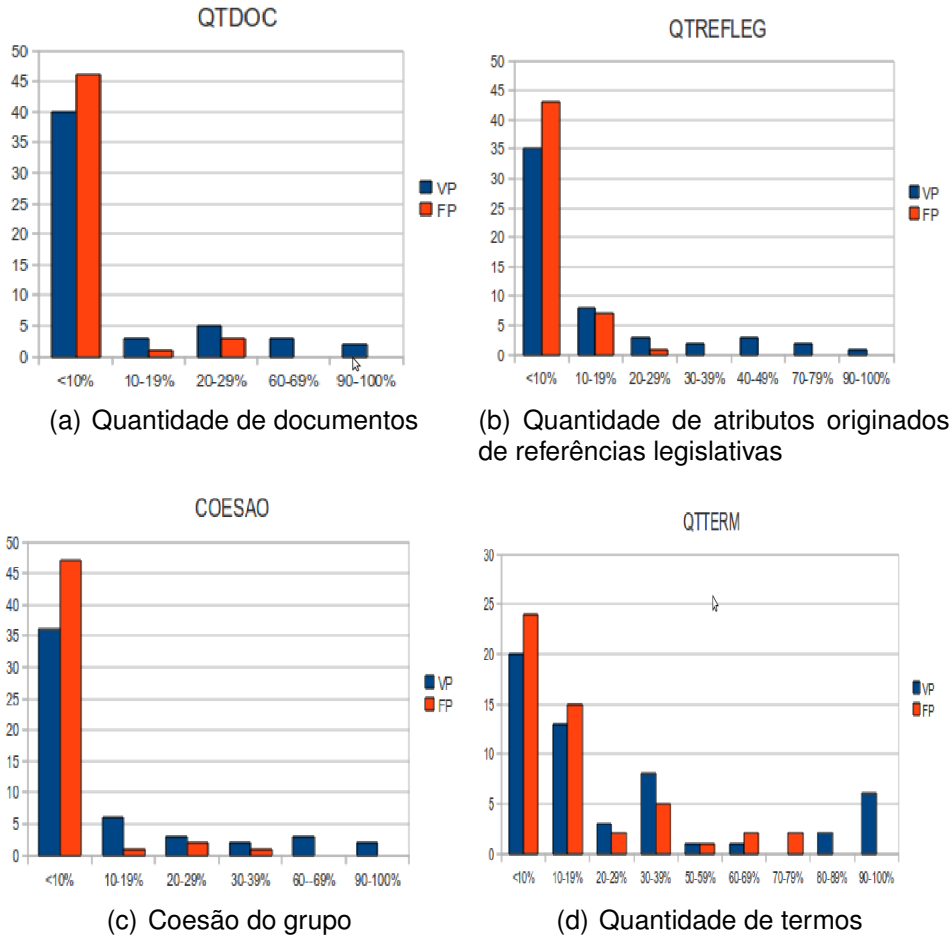
Os documentos foram categorizados em 74 das 453 classes geradas pelo agrupamento. A Figura 5.4 apresenta a quantidade de verdadeiros positivos (VP) e de falsos positivos (FP) tabulados em faixas de similaridade, iniciando pelos categorizados com mais de 40% de similaridade com a classe, seguidos de faixas de Δ5% até um mínimo de 5% de similaridade. Verifica-se, aqui, que com similaridades mais altas, é maior a probabilidade de se obter um verdadeiro positivo. O contrário também se verifica. Acima de 30% de similaridade não ocorreram falsos positivos.

Verificamos, também, que 26 dos 53 verdadeiros positivos (quase 50%) ocorreram em classes que correspondem a grupos com menos de 4 documentos, sendo 18 ($\frac{1}{3}$) em classes correspondentes a grupos com um único documento. Ressaltando que ajustamos para 4 documentos o limiar para descarte de grupos, utilizados por algumas das variações do algoritmo, percebemos importância de não realizar tal descarte. Note-se que em ambiente de produção tal proporção, provavelmente, não se verificará. Pois, em nossos exemplos de uso, lidamos com um *corpus* bem menor[6] do que o real montante disponível na instituição judiciária[7] e, conseqüentemente, o tamanho médio dos grupos aumentará.

Foram, novamente, extraídos indicadores das classificações como descrito na Seção 5.3.2. Desta vez pôde-se perceber uma fraca relação entre os indicadores **qtdoc**, Figura 5.5(a), **coesao**, Figura 5.5(c), **qtterm**, Figura 5.5(d), **qtattseed**, Figura 5.6(a), **qtrefleg**, Figura 5.5(b), **qtmerge**, Figura 5.6(b) e **qtattdoc**, Figura 5.6(c) além da similaridade entre

[6] 716 documentos

[7] O Tribunal Regional Federal da 4ª Região, desde sua fundação em 1988 já julgou mais de 3 milhões de processos, a maioria, no entanto, só existe em meio eletrônico na forma de imagem digitalizada, não sendo viável tratá-las textualmente.



(a) Quantidade de documentos

(b) Quantidade de atributos originados de referências legislativas

(c) Coesão do grupo

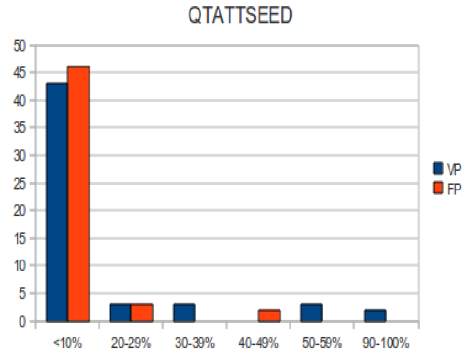
(d) Quantidade de termos

Figura 5.5 – Relação entre os indicadores **qtdoc**, **coesao**, **qtterm** e **qtrefleg** e a avaliação humana

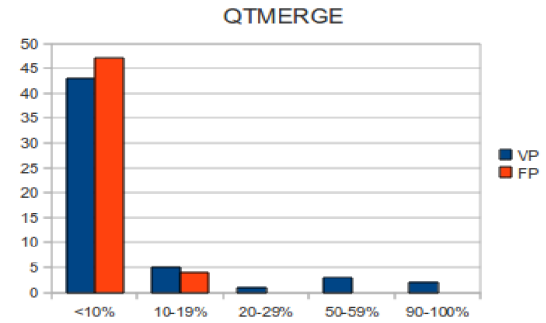
o documento classificado e o centróide do grupo correspondente à classe, **simcateg**. Para a geração destes gráficos, os indicadores citados foram normalizados, de forma que 0% representa sua menor incidência e 100% a sua maior incidência[8]. O eixo vertical representa a quantidade de verdadeiros/falsos positivos e o horizontal representa o valor normalizado dos indicadores, agrupados em faixas de 10% em 10%. O que se evidencia da análise dos gráficos das Figuras 5.5(a) a 5.5(b) é que quando há uma alta incidência do respectivo indicador, a avaliação da classificação é, sempre, positiva. O contrário, porém, não se verifica.

Não tão evidente é o fato de que a relação mais fraca é a do indicador **qtterm** e a mais forte é a do indicador **qtrefleg**. Estes indicadores estão, no entanto, relacionados entre si. Pois, ambos são contagens do tipo de origem dos atributos e, assim, quanto maior o **qtrefleg**, tanto menor o **qtterm**. Verificamos que os atributos com origem em referências legislativas nos centróides iniciais representavam cerca de 4,9% dos atributos e que nos centróides finais eles representam cerca de 15,17%, percebe-se que este tipo de atributo

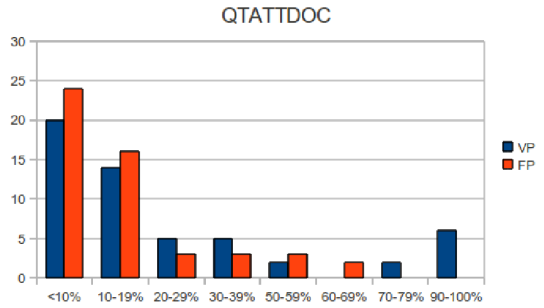
[8]No caso do indicador **qtdoc**, por exemplo, 0% representa grupos com um único documento e 100% representa grupos com 25 documentos.



(a) Quantidade de atributos no centróide



(b) Quantidade de uniões de grupos realizadas no grupo



(c) Soma da quantidade de atributos nos documentos do grupo

Figura 5.6 – Relação entre os indicadores **qtattseed**, **qtmerge** e **qtattdoc** e a avaliação humana

tem um papel importante na qualidade da categorização. Em trabalhos futuros, consideraremos implementar o exemplo de uso brevemente descrito na Seção 5.6.4, agrupamento semi-supervisionado por referências legislativas.

Além disto, se considerarmos que, em produção, teremos um conjunto de documentos muito maior, a quantidade de documentos em cada grupo vai aumentar e, conseqüentemente, também aumentará a quantidade de atributos não nulos nos centróides. Espera-se que, com isto, aumente a incidência de verdadeiros positivos, quando na fase de categorização.

5.10 Impressões dos Especialistas Humanos

De acordo com o especialista humano 1, as falhas de classificação ocorreram quando o documento a ser classificado versava sobre matéria de muita especificidade. Citou o exemplo do documento 20, onde o tema era “prescrição intercorrente” e os documentos retornados versavam sobre “prescrição”, mas não “intercorrente”. Tal documento apresenta 12 ocorrências do termo “prescrição intercorrente”, mas apresenta 10 ocorrências do termo “prescrição”, além de 1 de “extinção do processo” e 1 de “imprescritibilidade”, termos estes, que ocorrem em documentos atribuidos ao grupo gerador da classe na qual o documento

20 foi classificado. Além disto, reproduzindo parcialmente o seguinte trecho: “Lei 11.051, de 30.12.2004, permite a decretação da prescrição intercorrente por iniciativa judicial (...) a prescrição intercorrente em matéria tributária (...) viabilizando o decreto de prescrição”, percebemos que a última referência a “prescrição”, considerado o contexto, é, na verdade, uma referência a “prescrição intercorrente” e não ao termo mais genérico “prescrição”. A correta detecção do termo, expresso como “prescrição”, mas com o sentido de “prescrição intercorrente”, extrapola nossa proposta, demandando a implementação de desambigüação semântica.

Este especialista foi questionado a respeito da utilidade de um sistema que realize pesquisa jurídica automaticamente, com base nos documentos anexados ao processo eletrônico, retornando documentos com a jurisprudência correlata, com a precisão ora aferida (cerca de 50,4%). Manifestou-se dizendo que tal retorno aceleraria seu processo de pesquisa de jurisprudência. Informou que utilizando um sistema padrão de pesquisa de jurisprudência, a saber pesquisa por palavras-chave na ementa dos documentos, costuma ter que ler uma média de 35 documentos antes de encontrar aquele que lhe traga informação jurídica necessária para sua argumentação no caso em que está trabalhando. A respeito das ementas, informa que em torno de 30% das ementas ou estão erradas ou não suficientemente específicas.

O especialista humano 2 manifesta que costuma realizar buscas utilizando combinação de parâmetros ou expressões muito específicas e que comumentemente se depara com resultados extremos: ou não retorna nenhum documento ou retornam centenas ou milhares, inclusive para combinação de vários termos. Ressaltou que uma pesquisa usando as tecnologias ora propostas não trarão benefício nos casos em que retorna muitos (10) documentos e nenhum trata do problema específico. Mas, ressalta que se um único deles estiver relacionado ao tema buscado é mais útil que as buscas convencionais que retornam “dezenas ou mesmo centenas de casos muito genéricos”. Ressalta, ainda, que impressionou-se muito quando obteve poucos documentos (5, no máximo) e todos ou quase todos eram úteis para a solução de seu problema. Atentou, também, para o fato de que as buscas convencionais falham em reconhecer documentos quando usam expressões diferentes da buscada, e entende ser este o maior problema das pesquisas textuais.

Por fim, este especialista ainda ressaltou que o benefício social da implantação de um sistema baseado na metodologia proposta não se restringiria à aceleração da tramitação processual, mas, também, na relocação de recursos humanos para tarefas mais especializadas, citando, como exemplo, os estagiários e advogados iniciantes, comumentemente alocados para realização de pesquisa jurisprudencial.

Note que o especialista humano 1 aponta como maior causa dos erros de classificação a extrema especificidade do documento classificado. Já o especialista humano 2 entendeu que, comparando com os sistemas de busca convencionais, o “sistema” ora avaliado conseguiu retornar documentos com maior grau de especificidade. Embora pareçam visões

conflitantes, note-se que o especialista humano 1 não emitiu seu parecer fazendo comparação com os sistemas convencionais. Simplesmente frisou que os erros que ocorreram foram em virtude do maior grau de especificidade do documento classificado.

5.11 Considerações Finais

Neste capítulo avaliamos os resultados das execuções do algoritmo original de Aggarwal, Gates e Yu [AGY04], e das variações propostas para evolução deste algoritmo. Tendo em vista não encontrarmos relações entre os indicadores extraídos dos resultados da validação por especialista humano, e após análise mais detalhada de algumas classificações, verificamos a necessidade de valorizar termos cuja semântica fosse mais específica, em detrimento de termos muito genéricos, como “crime”.

Adotados os procedimentos da Seção 5.6.2.2, executamos novamente, nosso exemplo de uso, gerando novos agrupamentos. Recalculamos as medidas internas destes agrupamentos para selecionar aquele que proveria as classes para categorização. Descobrimos que, do ponto de vista das medidas internas, o agrupamento gerado pelo algoritmo 3, que descarta documentos mas não descarta nem divide grupos, obteve melhor performance. No entanto, tendo em vista que sua performance média superou por meros 2% a performance média do algoritmo 6, que não realiza descartes e realiza a divisão implícita, optamos por trabalhar com este último.

A classificação desta segunda execução de nosso exemplo de uso foi avaliada por dois especialistas humanos que, além de identificarem verdadeiros e falsos positivos, apresentaram breve relato de suas impressões sobre nossa proposta de pesquisa de jurisprudência, sinalizando que o uso de aprendizado de máquina para este fim pode contribuir para acelerar o processo de cognição no meio judicial brasileiro. Aprofundamos tais ponderações no próximo capítulo, onde concluímos nosso estudo.

Conclusão

No decorrer deste estudo, revisamos os fundamentos teóricos na área do aprendizado de máquina. Na área de aprendizado não supervisionado, estudamos o *K-Means* [Mac67], o mais popular algoritmo *flat* para *hard clustering* e o algoritmo *EM* [DLR77], clássico algoritmo *flat* para *soft clustering*. Realizamos, também, breve revisão de outras metodologias nesta área, como agrupamentos hierárquicos e agrupamentos semi-supervisionados. Na área de aprendizado supervisionado, destacamos, de nossa revisão, as redes Bayesianas [Pea85] e o algoritmo SVM [BGV92].

Aprofundamos nosso estudo revisando trabalhos relacionados ao que aqui desenvolvemos. Vários destes trabalhos apresentaram soluções calcadas em processo de agrupamento baseado em modificações do EM visando a geração de redes bayesianas [Fri97, Fri98, ELF+00, EF01, TCP+01, CLW+04]. Em outros, o processo empregou o SVM ou alguma de suas variações para a classificação [RFK02a, ZWC+03, LCF+07, HCT07, FH08, NS04] precedido de diferentes algoritmos de agrupamento.

O estudo dos trabalhos relacionados conclui aprofundando a proposta de Aggarwal, Gates e Yu [AGY04]. Percebemos que esta proposta, de partir de uma prévia classificação para gerar um conjunto de classes melhorado, identifica-se com a classificação ementária que encontramos na jurisprudência brasileira. Além disto, sua proposta de conduzir as iterações reduzindo a dimensionalidade dos centróides e, com isso, acelerar o processamento a cada iteração, sinalizou a viabilidade do uso deste algoritmo em ambiente de produção. O Tribunal Regional Federal da 4ª Região, que forneceu o conjunto de documentos que compôs nosso *corpus* de estudo, em maio de 2009, quando iniciamos os procedimentos de *download* da jurisprudência, contava com mais de 3 milhões de documentos em sua jurisprudência e concluía, em média, 700 processos por dia. Os processos conclusos tornam-se nova jurisprudência e, assim, é preciso que se reexecute o agrupamento sobre o *corpus*. Tal execução necessita estar conclusa no dia seguinte, quando nova leva de processos serão julgados. Neste contexto, conta muito a seleção de algoritmos que se destaquem na velocidade de processamento. Além disto, o categorizador proposto utilizou função de similaridade comparando os novos documentos com os centróides dos grupos geradores das classes e, assim, o seu treinamento se constitui do próprio processo de agrupamento. Já os trabalhos que se utilizam do SVM ou de seus derivados, demandam, ainda, o treinamento do classificador após a execução do agrupamento e, portanto, agravam o problema de incluir novos documentos no *corpus* e, eventualmente, o problema da descoberta de novas classes.

Apresentamos, então, nossa proposta de adaptação do algoritmo de Aggarwal, Gates e Yu [AGY04] para uso em *corpus* jurídico. Verificamos ser necessária a adaptação do al-

goritmo, conforme exposto na Seção 5.9, e não meramente sua utilização em *corpus* com características distintas daquele empregado originalmente por seus autores. Isso porque, em sua forma original, o algoritmo realiza descartes de documentos e grupos, a título de ruído. Ressaltamos que no contexto de nosso estudo, o da pesquisa jurídica, tais "ruídos" são valorizados. É o caso de temas novos, sem precedentes, tal como o exemplo de recente exposição dedicado à questão das células-tronco. Objeto este de forte disputa, chegando a mobilizar setores de nossa sociedade. Por esta razão propusemos modificar o algoritmo evitando todo e qualquer descarte. Outra modificação proposta, seguindo discussão suscitada pelos próprios autores, foi a de implementar um passo de divisão de grupos, que não chegou a ser explorada por Aggarwal, Gates e Yu [AGY04]. Inicialmente implementamos a divisão de grupos como um passo extra da iteração. No entanto, percebemos que, por não descartar documentos no passo de atribuição, obtínhamos, obviamente, grupos de baixa densidade que, conseqüentemente, se tornavam os maiores candidatos à divisão. Assim, inspirados no algoritmo TOD [FK99] *apud* [LCF+07], que cria um novo grupo quando um documento está muito distante de qualquer centróide, experimentamos realizar a divisão dos grupos dentro do próprio passo de atribuição; denominamos tal procedimento de divisão implícita, ao invés de criar um passo próprio para tanto. Desta maneira, economizou-se o custo computacional imposto pelo passo de divisão que necessita avaliar a variância de similaridade *intra-cluster* de cada grupo para decidir se o grupo deve ser dividido. Outra economia computacional advinda desta modificação foi a eliminação das sub-iterações realizadas no passo de divisão de grupos.

Verificamos, também, que o algoritmo usando a divisão implícita obteve melhor performance média que o algoritmo que implementou passo de divisão de grupos, com relação às medidas de qualidade internas *Relative Hardness Measure* e $\bar{\rho}$-$Measure$, em face do conjunto de treino utilizado. O algoritmo que usou a divisão implícita não apresentou, no entanto, melhor performance média que a variante que descarta documentos, não descarta grupos e não realiza qualquer divisão de grupos. Porém, consideramos que tal diferença, 2,2%, era um prejuízo aceitável se considerado o fato de que estávamos elegendo uma versão que não realizava descarte de documentos. Preferimos grupos um pouco menos densos e um pouco menos distintos entre si a arriscarmos a possibilidade de que alguém, por exemplo, não seja liberto porque a argumentação capaz de convencer um juiz de sua soltura existia, mas não foi encontrada. Além disso, o especialista humano 2, ao manifestar suas impressões acerca dos resultados de nosso exemplo de uso, declarou ser preferível a recuperação de um conjunto pequeno de documentos[9], desde que, obviamente, eles contenham a informação buscada. Ressalte-se que o limiar para descarte de grupos utilizado originalmente por Aggarwal, Gates e Yu [AGY04] era de 8 documentos e, em nosso exemplo de uso, tivemos de reduzir para 4. Ainda assim, um a menos que o máximo expressado como ideal pelo especialista humano 1.

[9] Até 5 documentos.

Tabela 5.18 – Quantidade máxima de categorias usadas nos trabalhos relacionados

Referência	Quantidade de Categorias
Feinerer e Hornik [FH08]	2
Nguyen e Smeulders [NS04]	2
Cong, Lee, Wu e Liu [CLW+04]	2
Nigam e Ghani [NG00]	2
Zeng *et al.* [ZWC+03]	10
Li, Chi, Fan e Xue [LCF+07]	10
Toutanova *et al.* [TCP+01]	15
Raskutti, Ferrá e Kowalczyk [RFK02a]	20
Friedman e Elidan [EF01]	20
Hao, Chiang e Tu [HCT07]	90
Aggarwal, Gates e Yu [AGY04]	1.167
Este estudo	453

Finalmente, ao analisarmos a classificação, verificamos que foi obtida uma precisão de 50,5%. Embora pareça, a princípio, que esta precisão é muito baixa, ressaltamos que não se pode, por exemplo, compará-la com performances obtidas por classificadores binários. Nosso classificador obteve esta performance em face de 453 possíveis classes e não de duas como o fazem os classificadores binários. A Tabela 5.18 apresenta a quantidade máxima de categorias usadas nos experimentos de classificação discutidos nos trabalhos relacionados. Exceto pelo algoritmo no qual baseamos nosso trabalho, que superou em pouco mais de 150% o quantitativo de classes obtidas em nosso exemplo de estudo, os demais trabalhos lidaram com quantidade bem inferior de classes. O único trabalho que reporta a precisão de seus resultados é o de Toutanova *et al.* [TCP+01], que, usando 15 classes, obteve resultados variando de cerca de 58% a 88%.

Em relação ao custo computacional, entendemos ser viável a implantação de sistema baseado em nossa proposta para atender a carga de processos no ritmo de seu crescimento num ambiente de produção. Embora não tenhamos feito aferições exaustivas, todos os exemplos de uso executaram em um computador com processador AMD Athlon64 X4@2.6Ghz, com 4Gb de memória. Já o equipamento recém adquirido pelo Ministério Público Federal para uso com o processo eletrônico é um Dual Hexa-Core HT@2.6Ghz, que expõe 24 processadores para o sistema operacional e dispõe de 32Gb de memória. Desde o *parsing* até o agrupamento, gastou-se menos de 3 horas e isto considerando-se que cada um destes procedimentos foi implementado em programas separados e, assim, neste tempo, deve-se levar em conta a repetição da inicialização de variáveis que, por carregarmos o banco de dados lexical em memória, representa parcela significativa deste tempo. A carga do tesauro representa em torno de 12 minutos, por exemplo. Ao unificar estes procedimentos num só programa, o tempo de execução sofrerá redução significativa.

Enfim, nesta nossa primeira proposta de uso de aprendizado de máquina para recupe-

ração de jurisprudência, cremos haver contribuído com respeito a:

1. **Uso do aprendizado de máquina na pesquisa jurisprudencial**: apresentamos processo de categorização auxiliado por agrupamento baseado em algoritmo, selecionado em virtude de

   (a) partir de um conjunto de classes previamente determinado e gerar um novo conjunto de classes melhorado e, assim, reduzir os erros de classificação encontrados nas ementas da jurisprudência, além de descobrir as classes sem que seja necessário pré-configurar sua quantidade antes da execução do algoritmo;

   (b) ter boa performance computacional, reduzindo a dimensionalidade dos atributos a cada iteração, sinalizando a sua viabilidade de implantação em ambiente de produção.

2. **Evolução do algoritmo selecionado**: adaptamos este algoritmo para as necessidades específicas da pesquisa de jurisprudência, realizando com sucesso modificações que

   (a) eliminaram os descartes de documentos e grupos, que poderiam impedir que fossem encontrados documentos relativos a casos sem precedentes que, se apresentados ao juiz do caso em andamento, podem fazer a diferença entre o sucesso ou insucesso da respectiva demanda;

   (b) implementaram a divisão de grupos, inexistente no algoritmo original e que permite que os grupos tornem-se mais refinados sem a necessidade de se realizar descartes;

   validando, assim, a evolução deste algoritmo, que, em nosso exemplo de uso, teve performance superior à do algoritmo original em relação à medida *Relative Hardness*, e equivalente, no caso da medida $\bar{\rho}$, e cujos resultados finais foram recebidos positivamente por especialistas humanos.

3. **Prototipação do processo proposto**: implementamos protótipo do algoritmo proposto e executamos nosso exemplo de uso em computadores de uso pessoal, utilizando linguagem interpretada, sem focar na otimização do código e, ainda assim, o pré-processamento consumiu 9s por documento, o agrupamento levou 1h 30min para agrupar 716 documentos e o algoritmo de categorização classificou um documento a cada 2s. Verificamos, portanto, a viabilidade de sua utilização em ambiente de produção.

4. **Proposta de novo paradigma**: ao analisar os resultados do processo de agrupamento e classificação implementado, percebemos que adotar um paradigma “bag of terms and law references” pode trazer benefícios superiores, não somente a um

paradigma “bag of words”, mas até mesmo sobre um paradigma ‘’bag of terms”. Embora seja necessário, ainda, aprofundar esta questão, nossa análise dos resultados indica que os atributos com origem em referências legislativas têm um papel mais importante no sucesso da classificação do que supúnhamos inicialmente.

Além disto, elencamos como contribuições secundárias

1. **Merge de Dicionários**: unificação dos dicionários Unitex-PB e Wiktionary em língua portuguesa e Wiktionary em latim, produzindo um dicionário mais completo;
2. **Corpus**: organização de corpus jurídico em língua portuguesa, contendo jurisprudência do Tribunal Regional Federal da 4a Região;
3. **Parser**: codificação de parser para a língua portuguesa, reconhecendo palavras constantes do dicionário, e, também, referências legislativas, tal como explicado na Seção 4.4.3;
4. **Tagger**: codificação de tagger que lematiza os tokens extraídos pelo parser, baseando-se em dicionário, utilizando método iterativo alternando o uso de regras gramaticais e probabilidades;
5. **Merge de Tesauros**: unificação dos tesauros jurídicos do Senado Federal e do Conselho da Justiça Federal, identificando automaticamente termos iguais e, através de especialista, termos equivalentes;
6. **Extrator de termos**: codificação de reconhecedor de termos jurídicos na seqüência de lemas obtida através do tagger, usando o tesauro jurídico unificado.

Em nosso exemplo de uso, separamos 1.192 documentos dos 43.704 obtidos do Tribunal Regional Federal da 4ª Região. Conforme informado, este procedimento foi necessário, tendo em vista que o algoritmo de Aggarwal, Gates e Yu [AGY04] atribui documentos a um único grupo. Por discutirem acerca de múltiplos temas, tais documentos foram descartados. No entanto, verificando que nosso exemplo de uso tratou, apenas, 2,73% dos documentos obtidos, percebemos que, em trabalhos futuros, será necessário dedicar nossos esforços a acrescentar novas evoluções nesse algoritmo a fim de habilitá-lo a agrupar e classificar documentos em múltiplos grupos/classes. Uma das possibilidades consideradas é incorporação da técnica de segmentação de documentos, proposta por Tagarelli e Karypis [TK08].

Também merece mais atenção o papel que as referências legislativas representam na qualidade dos resultados da categorização. Em trabalhos futuros, consideramos alterar o algoritmo proposto aumentando o seu grau de semi-supervisionamento. Atualmente, esse algoritmo é considerado semi-supervisionado por partir de grupos gerados em função de classificação prévia. Mas, após a inicialização dos grupos, não há quaisquer restrições nas operações de atribuição de documentos e de aglomeração e divisão de grupos.

Poderíamos, assim, aplicar restrição na função de similaridade do algoritmo de agrupamento, levando-a a considerar apenas os atributos oriundos de referências legislativas. Os atributos originados de termos jurídicos, embora ignorados, continuariam presentes nos vetores dos documentos e dos centróides. Desta maneira, posteriormente, a função de similaridade do algoritmo de categorização livre dessa restrição, utilizaria tanto atributos oriundos de termos jurídicos quanto atributos originados de referências legislativas. Isso poderá permitir que os documentos produzidos pelas partes possam ser categorizados mesmo na eventualidade de seus advogados desconhecerem uma ou mais legislações pertinentes ao caso em questão.

Um problema que decorrerá desta técnica é uma redução drástica dos atributos, que poderão, em muitos casos, atingir patamares mínimos, como 5 ou menos atributos. Em conseqüência disto, poderá ficar comprometida a performance desse algoritmo, por basear-se na quantidade de atributos para regular suas iterações e determinar seu encerramento. Neste caso, as iterações poderiam ser regidas por outra variável, como a coesão, por exemplo;

Outros pontos que podem ser desenvolvidos em trabalhos futuros:

1. Categorização dos 133 documentos do conjunto de operação que restaram sem ser categorizados. Tal classificação poderia ser validada, inclusive, perante conjuntos de classes oriundos de vários algoritmos de agrupamento, não somente aqueles prototipados em nosso exemplo de uso, mas, também, outros que, na ocasião, se considere oportuno acrescentar ao conjunto de estudo;

2. Paralelização do algoritmo, tendo em vista que nos últimos anos tem-se observado que a arquitetura dos computadores disponíveis no mercado vem abandonando o modelo monoprocessado. Ao executar os protótipos dos algoritmos, em nosso exemplo de uso, observou-se constantemente, que a taxa de uso de uma das CPUs subia para 100%, equanto as demais CPUs oscilavam entre 0% e 20%. Este é um claro indicativo do benefício que se pode obter através da paralelização

   (a) dos procedimentos de *parsing* e desambigüação que pode ocorrer tanto em nível de documento quanto em nível de parágrafo;

   (b) do passo de atribuição, onde cada documento é comparado com os centróides de um conjunto estático de grupos e todas as modificações ocorrem em um conjunto novo de grupos;

   (c) do passo de aglomeração, onde primeiro se comparam todos os centróides dos grupos, decidindo-se por quais devem ser mesclados e, posteriormente, realiza-se a mescla;

   (d) do passo de projeção, cujos descartes de atributos realizados nos centróides dos grupos é uma operação independente entre os grupos;

(e) dos cálculos de similaridade, fortemente baseados em somatórios de sub-expressões cujos dados são independentes entre si.

Tal conclusão leva em conta, principalmente, a consideração de que equipamentos de ambiente corporativo, como o Ministério Público Federal, têm maior disponibilidade de processadores que equipamentos de uso pessoal.

Referências Bibliográficas

[AGY04] C. Aggarwal, S. Gates e P. Yu. “On using partial supervision for text categorization”, *IEEE Transactions on Knowledge and data Engineering*, vol. 16-2, Dez 2004, pp. 245–255.

[Alp04] E. Alpaydin. “Introduction to machine learning”. Cambridge, Massachusetts:MIT Press, 2004, 423p.

[AS07] E. Agirre e A. Sorca. “Semeval2007 task 02: evaluating word sense induction and discrimination systems”. In: SemEval Workshop, 2007, pp. 7–12.

[Ben73] J. Benzécri. “L’analyse des données: L’analyse des correspondances”. Dunod, 1973, 619p.

[BGV92] B. Boser, I. Guyon e V. Vapnik. “A training algorithm for optimal margin classifiers”. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, 1992, pp. 144–152.

[BHHS+02] A. Ben-Hur, D. Horn, H. Siegelmann e V. Vapnik. “Support vector clustering”, *The Journal of Machine Learning Research*, vol. 2, Jan 2002, pp. 125–137.

[BLA+97] J. Bezdek, W. Li, Y. Attikiouzel e M. Windham. “A geometric approach to cluster validity for normal mixtures”, *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 1-4, Dez 1997, pp. 166–179.

[BM98] A. Blum e T. Mitchell. “Combining labeled and unlabeled data with co-training”. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, 1998, pp. 92–100.

[BR04] M. Braschler e B. Ripplinger. “How effective is stemming and decompounding for german text retrieval?”, *Information Retrieval*, vol. 7-3, Jul 2004, pp. 291–316.

[CAKZ+05] J. Conrad, K. Al-Kofahi, Y. Zhao e G. Karypis. “Effective document clustering for large heterogeneous law firm collections”. In: Proceedings of the Tenth International Conference on Artificial Intelligence and Law, 2005, pp. 177–187.

[CCR+02] J. Cappelleri, W. Cefalu, J. Rosenstock, I. Kourides e R. Gerber. “Treatment satisfaction in type 2 diabetes: A comparison between an inhaled insulin regimen and a subcutaneous insulin regimen* 1”, *Clinical Therapeutics*, vol. 24-4, Abr 2002, pp. 552–564.

[CDA+98] S. Chakrabarti, B. Dom, R. Agrawal e P. Raghavan. “Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies”, *The VLDB Journal–The International Journal on Very Large Data Bases*, vol. 7-3, Ago 1998, pp. 163–178.

[CDH+01] J. Carlberger, H. Dalianis, M. Hassel, O. Knutsson et al. “Improving precision in information retrieval for swedish using stemming”. In: Proceedings of NODAL-IDA, 2001, pp. 21–22.

[CH67] T. Cover e P. Hart. “Nearest neighbor pattern classification”, *IEEE Transactions on Information Theory*, vol. 13-1, Jan 1967, pp. 21–27.

[CLW+04] G. Cong, W. Lee, H. Wu e B. Liu. “Semi-supervised text classification using partitioned EM”. In: Database Systems for Advanced Applications, 2004, pp. 229–239.

[CV95] C. Cortes e V. Vapnik. “Support-vector networks”, *Machine Learning*, vol. 20-3, Mar 1995, pp. 273–297.

[DB79] D. Davies e D. Bouldin. “A cluster separation measure”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1-2, Abr 1979, pp. 224–227.

[Dem06] J. Demšar. “Statistical comparisons of classifiers over multiple data sets”, *The Journal of Machine Learning Research*, vol. 7, Dez 2006, pp. 1–30.

[DLR77] A. Dempster, N. Laird e D. Rubin. “Maximum likelihood from incomplete data via the EM algorithm”, *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39-1, Jan 1977, pp. 1–38.

[Dun73] J. Dunn. “A fuzzy relative of the isodata process and its use in detecting compact well separated clusters”, *Journal of Cybernetics*, vol. 3-3, Jan 1973, pp. 32–57.

[EF01] G. Elidan e N. Friedman. “Learning the dimensionality of hidden variables”. In: Proceedings of the Seventeenth Conference in Uncertainty in Artificial Intelligence, 2001, pp. 144–151.

[ELF+00] G. Elidan, N. Lotner, N. Friedman e D. Koller. “Discovering hidden variables: A structure-based approach”. In: Advances in Neural Information Processing Systems (NIPS), 2000, pp. 779–786.

[FH08] I. Feinerer e K. Hornik. “Text mining of supreme administrative court jurisdictions”. In: Proceedings of the 31st Annual Conference of the Gesellschaft für Klassifikation, Mar 2008, pp. 569–576.

[FK99] M. Friedman e A. Kandel. “Introduction to pattern recognition: statistical, structural, neural, and fuzzy logic approaches”. Imperial College Press, 1999, 329p.

[Fri97] N. Friedman. “Learning belief networks in the presence of missing values and hidden variables”. In: International Conference on Machine Learning, 1997, pp. 125–133.

[Fri98] N. Friedman. “The bayesian structural EM algorithm”. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, 1998, pp. 129–138.

[GCB05] N. Grira, M. Crucianu e N. Boujemaa. “Unsupervised and Semi-Supervised Clustering: a Brief Survey”, Relatório Técnico, MUSCLE European Network of Excellence (FP6), Ago 2005, pp. 1–12.

[GML+09] S. García, D. Molina, M. Lozano e F. Herrera. “A study on the use of non-parametric tests for analyzing the evolutionary algorithms’ behaviour: a case study on the CEC’2005 special session on real parameter optimization”, *Journal of Heuristics*, vol. 15-6, Jun 2009, pp. 617–644.

[Gon05] M. Gonzalez. “Termos e Relacionamentos em Evidência na Recuperação de Informação”, Tese de Doutorado, Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, 2005, 182p.

[HA85] L. Hubert e P. Arabie. “Comparing partitions”, *Journal of Classification*, vol. 2-1, Jul 1985, pp. 193–218.

[HCT07] P. Hao, J. Chiang e Y. Tu. “Hierarchically SVM classification based on support vector clustering method and its application to document categorization”, *Expert Systems with Applications*, vol. 33-3, Out 2007, pp. 627–635.

[Hof99] T. Hofmann. “The cluster-abstraction model: Unsupervised learning of topic hierarchies from text data”. In: International Joint Conference on Artificial Intelligence, 1999, pp. 682–687.

[HT06] P. Halácsy e V. Trón. “Benefits of deep NLP-based lemmatization for information retrieval”. In: CLEF 2006 Workshop/Working Notes, 2006, 9p.

[HTF+05] T. Hastie, R. Tibshirani, J. Friedman e J. Franklin. “The elements of statistical learning: data mining, inference and prediction”, *The Mathematical Intelligencer*, vol. 27-2, Jun 2005, pp. 83–85.

[HW79] J. Hartigan e M. Wong. “Algorithm as 136: A k-means clustering algorithm”, *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28-1, Jan 1979, pp. 100–108.

[IPR+08] D. Ingaramo, D. Pinto, P. Rosso e M. Errecalde. “Evaluation of internal validity measures in short-text corpora”. In: Proceedings of the 9th International Conference on Computational Linguistics and Intelligent Text Processing, 2008, pp. 555–567.

[JAS+07] F. Jaegger, A. Araújo, A. Souza, D. Toledo, D. Gorovitz, C. Sandes, E. Oliveira, L. Cunha, L. Gesteira, L. Reis, M. Carvalho e M. Innecco. “Vocabulário controlado básico”, Serviço de Gerência da Rede Virtual de Bibliotecas-Congresso Nacional-RVBI, junho 2007, 564p.

[JN02] M. Jordan e A. Ng. “On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes”. In: Advances in Neural Information Processing Systems 14: Proceedings of the 2002 Conference, 2002, 841p.

[Joa98] T. Joachims. “Text categorization with support vector machines: Learning with many relevant features”. In: Machine Learning: ECML-98, 10th European Conference on Machine Learning, Abr 1998, pp. 137–142.

[Joa99] T. Joachims. “Transductive inference for text classification using support vector machines”. In: Machine Learning-International Workshop then Conference, 1999, pp. 200–209.

[KLJ+04] T. Korenius, J. Laurikkala, K. Järvelin e M. Juhola. “Stemming and lemmatization in the clustering of finnish text documents”. In: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, 2004, pp. 625–633.

[Lan95] K. Lang. “Newsweeder: Learning to filter netnews”. In: Proceedings of the Twelfth International Conference on Machine Learning, 1995, pp. 331–339.

[LCF+07] B. Li, M. Chi, J. Fan e X. Xue. “Support cluster machine”. In: Proceedings of the 24th International Conference on Machine Learning, 2007, pp. 505–512.

[LFV+09] L. Lopes, P. Fernandes, R. Vieira e G. Fedrizzi. “Exato LP – an automatic tool for term extraction from portuguese language corpora”. In: LTC’09 – Fourth Language and Technology Conference, 2009, pp. 427–431.

[LG94] D. Lewis e W. Gale. “A sequential algorithm for training text classifiers”. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1994, pp. 3–12.

[LSZ04] A. Lavelli, F. Sebastiani e R. Zanoli. “Distributional term representations: an experimental comparison”. In: CIKM ’04: Proceedings of the Thirteenth ACM

International Conference on Information and Knowledge Management, 2004, pp. 615–624.

[Lug04] G. Luger. “Inteligência artificial”. Porto Alegre:Bookmann, 2004, 4ª edição, vol. 1, 774p.

[Mac67] J. MacQueen. “Some methods for classification and analysis of multivariate observations”. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, 14p.

[MFBS+00] Y. Maarek, R. Fagin, I. Ben-Shaul e D. Pelleg. “Ephemeral Document Clustering for Web Applications”, Relatório Técnico, IBM Research, 2000, 26p.

[Mit97] T. Mitchell. “Machine learning”. McGraw-Hill, Mar 1997, 414p.

[MM01] O. Mangasarian e D. Musicant. “Lagrangian support vector machines”, *The Journal of Machine Learning Research*, vol. 1, Mar 2001, pp. 161–177.

[MM08] A. Mukhopadhyay e U. Maulik. “Unsupervised pixel classification in satellite imagery: a two-stage fuzzy clustering approach”, *Fundamenta Informaticae*, vol. 86-4, Out 2008, pp. 411–428.

[MN04] M. Muniz e M. Nunes. “A Construção de Recursos Linguístico-computacionais para o Português do Brasil: o Projeto de Unitex-PB”, Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2004, 72p.

[MRM+98] A. McCallum, R. Rosenfeld, T. Mitchell e A. Ng. “Improving text classification by shrinkage in a hierarchy of classes”. In: Proceedings of the Fifteenth International Conference on Machine Learning, 1998, pp. 359–367.

[MS00] C. Manning e H. Schütze. “Foundations of statistical natural language processing”. MIT Press, 2000, 680p.

[NG00] K. Nigam e R. Ghani. “Analyzing the effectiveness and applicability of co-training”. In: Proceedings of the Ninth International Conference on Information and Knowledge Management, 2000, pp. 86–93.

[NMT+00] K. Nigam, A. McCallum, S. Thrun e T. Mitchell. “Text classification from labeled and unlabeled documents using EM”, *Machine Learning*, vol. 39-2, Mai 2000, pp. 103–134.

[NS04] H. Nguyen e A. Smeulders. “Active learning using pre-clustering”. In: Proceedings of the Twenty-First International Conference on Machine Learning, 2004, 79p.

[Pea85] J. Pearl. “Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning”, Relatório Técnico, UCLA, 1985, pp. 329–334.

[PG94] E. Porath e I. Gilboa. “Linear measures, the gini index, and the income-equality trade-off”, *Journal of Economic Theory*, vol. 64-2, Dez 1994, pp. 443–467.

[PR07] D. Pinto e P. Rosso. “On the relative hardness of clustering corpora”. In: Text, Speech and Dialogue, 2007, pp. 155–161.

[PWS01] T. Pham, M. Worring e A. Smeulders. “Face detection by aggregated bayesian network classifiers”, In: Machine Learning and Data Mining in Pattern Recognition, vol. 2123, 2001, pp. 249–262.

[Ran71] W. Rand. “Objective criteria for the evaluation of clustering methods”, *Journal of the American Statistical Association*, vol. 66-336, Dez 1971, pp. 846–850.

[RFK02a] B. Raskutti, H. Ferrá e A. Kowalczyk. “Combining clustering and co-training to enhance text classification using unlabelled data”. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 620–625.

[RFK02b] B. Raskutti, H. Ferrá e A. Kowalczyk. “Using unlabelled data for text classification through addition of cluster parameters”. In: Proceedings of the 19th International Conference on Machine Learning, 2002, pp. 514–521.

[Rou87] P. Rousseeuw. “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis”, *Journal of Computational and Applied Mathematics*, vol. 20, Nov 1987, pp. 53–65.

[Sal97] S. Salzberg. “On comparing classifiers: Pitfalls to avoid and a recommended approach”, *Data Mining and Knowledge Discovery*, vol. 1-3, Set 1997, pp. 317–328.

[SEP06] B. Stein, S. Eissen e M. Potthast. “Syntax versus semantics”. In: 3rd International Workshop on Text-Based Information Retrieval (TIR-06), 2006, 47p.

[SEW03] B. Stein, S. Eissen e F. Wißbrock. “On cluster validity and the information need of users”. In: Proceedings of the 3rd IASTED International Conference on Artificial Intelligence and Applications, Set 2003, pp. 216–221.

[She04] D. Sheskin. “Handbook of parametric and nonparametric statistical procedures”. Chapman & Hall/CRC, 2004, 1736p.

[SK99] P. Somervuo e T. Kohonen. “Self-organizing maps and learning vector quantization for feature sequences”, *Neural Processing Letters*, vol. 10-2, Out 1999, pp. 151–159.

[SKK00] M. Steinbach, G. Karypis e V. Kumar. “A Comparison of Document Clustering Techniques”, Relatório Técnico, University of Minnesota, 2000, pp. 525–526.

[SMS+07] N. Sordi, M. Medeiros, E. Santos, G. Silva, N. Tavares, C. Galuban, R. Grasso, J. Martins, R. Castro, S. Carvalho, C. Castro, F. Léda, M. Tosta, C. Lopes, A. Lima, C. Lima, D. Dallegrave, F. D’Andrada, J. Teixeira e J. Lopes. “Tesauro jurídico da justiça federal”, Conselho da Justiça Federal, Fev 2007, 377p.

[SN99] B. Stein e O. Niggemann. “On the nature of structure and its identification”. In: Graph-Theoretic Concepts in Computer Science, 1999, pp. 122–134.

[Str05] P. Strömbäck. “The Impact of Lemmatization in Word Alignment”, Dissertação de Mestrado, Department of Linguistics and Philology, Uppsala University, 2005, 31p.

[TCP+01] K. Toutanova, F. Chen, K. Popat e T. Hofmann. “Text classification in a hierarchical mixture model for small training sets”. In: Proceedings of the Tenth International Conference on Information and Knowledge Management, 2001, pp. 105–113.

[TK08] A. Tagarelli e G. Karypis. “A segment-based approach to clustering multi-topic documents”. In: Text Mining Workshop, SIAM Data Mining Conference, 2008, 12p.

[TSK09] P. Tan, M. Steinbach e V. Kumar. “Introdução ao data mining: Mineração de dados”. Rio de Janeiro:Ciência Moderna, 2009, 900p.

[Wil45] F. Wilcoxon. “Individual comparisons by ranking methods”, *Biometrics Bulletin*, vol. 1-6, Dez 1945, pp. 80–83.

[XB91] X. Xie e G. Beni. “A validity measure for fuzzy clustering”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13-8, Ago 1991, pp. 841–847.

[YL99] Y. Yang e X. Liu. “A re-examination of text categorization methods”. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1999, pp. 42–49.

[YP97] Y. Yang e J. Pedersen. “A comparative study on feature selection in text categorization”. In: Machine Learning-International Workshop then Conference, 1997, pp. 412–420.

[ZRL96] T. Zhang, R. Ramakrishnan e M. Livny. “BIRCH: an efficient data clustering method for very large databases”. In: ACM SIGMOD Record, 1996, pp. 103–114.

[ZWC+03] H. Zeng, X. Wang, Z. Chen, H. Lu e W. Ma. “CBC: Clustering based text classification requiring minimal labeled data”. In: Third IEEE International Conference on Data Mining-ICDM, 2003, pp. 443–450.

Apêndice A. Programa de Seleção de Documentos

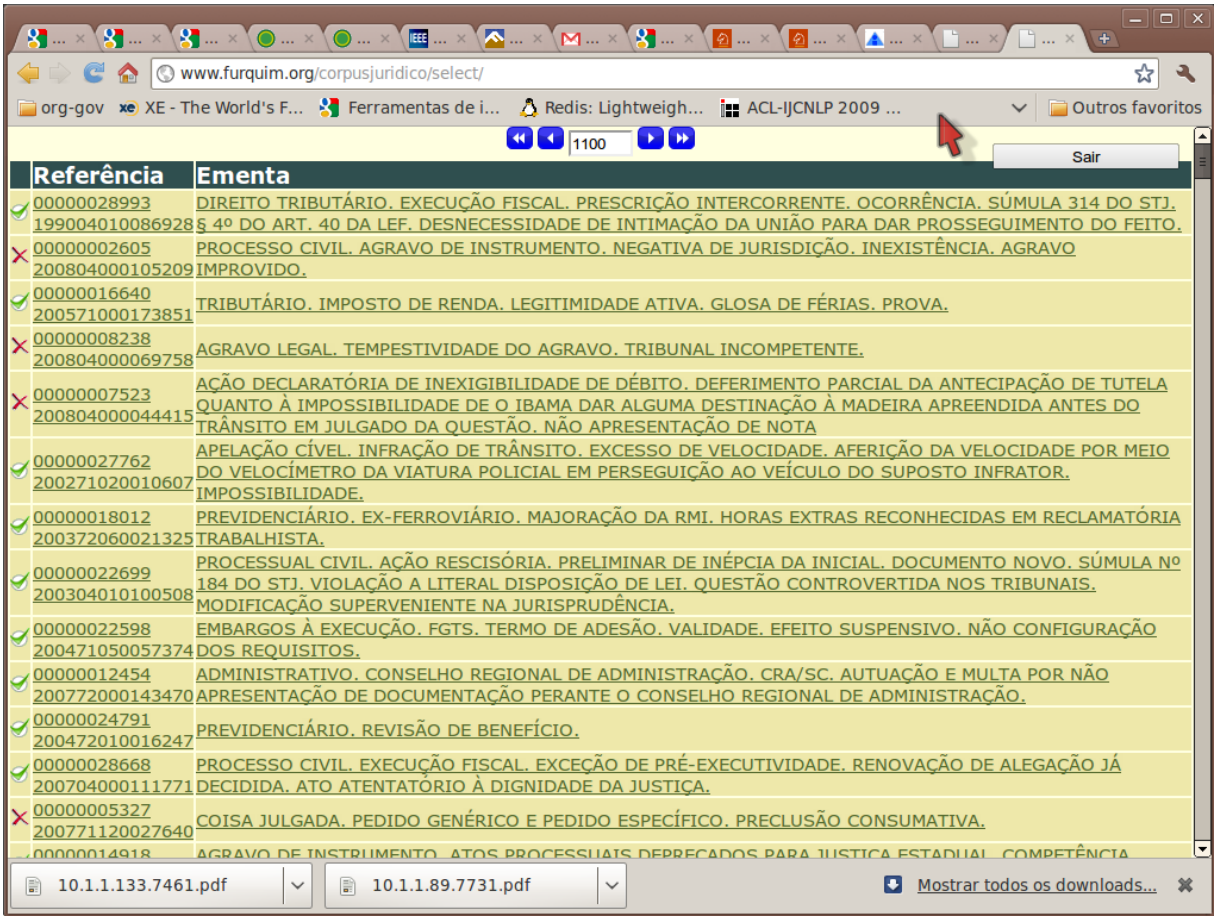


Figura A.1 – Programa para seleção/descarte de documentos

Apêndice B. Grupos Iniciais

A Tabela B.1 apresenta os grupos iniciais organizados pela quantidade de atributos não nulos em seus centróides. Seja $G'_j \subseteq G, \forall G_{ij} \in G'_j \rightarrow |\overrightarrow{\Phi(\vec{c}_{ij})}| = k \wedge \nexists G_{in} \in G'_n \rightarrow |\overrightarrow{\Phi(\vec{c}_{ij})}| = k$, onde $\overrightarrow{\Phi(\vec{c}_{ij})}$ é um vetor composto dos atributos não nulos de $\vec{c}_{ij}$ e todos grupos G_{ij} num mesmo G'_j têm centróides c_{ij} com a mesma quantidade de atributos não nulos. A Tabela B.2 apresenta a quantidade inicial de documentos em cada grupo. Seja $G'_j \subseteq G, \forall G_{ij} \in G'_j \rightarrow |G_{ij}| = k \wedge \nexists G_{in} \in G'_n \rightarrow |G_{in}| = k$, onde todos grupos G_{ij} num mesmo G'_j têm a mesma quantidade de documentos.

Tabela B.1 – Quantidade de atributos não nulos nas Classes/Grupos Iniciais

$\lvert\overrightarrow{\Phi(\vec{c}_{ij})}\rvert$	$\lvert G'_j\rvert$	$\lvert\overrightarrow{\Phi(\vec{c}_{ij})}\rvert$	$\lvert G'_j\rvert$	$\lvert\overrightarrow{\Phi(\vec{c}_{ij})}\rvert$	$\lvert G'_j\rvert$	$\lvert\overrightarrow{\Phi(\vec{c}_{ij})}\rvert$	$\lvert G'_j\rvert$	$\lvert\overrightarrow{\Phi(\vec{c}_{ij})}\rvert$	$\lvert G'_j\rvert$
3	4	4	9	5	7	6	2	7	7
8	5	9	9	10	7	11	6	12	8
13	7	14	8	15	9	16	4	17	2
18	6	19	4	20	4	21	5	22	1
23	3	24	2	25	7	26	3	27	1
28	3	29	4	30	5	31	2	32	3
33	1	34	2	35	2	36	2	37	1
38	1	39	5	41	3	42	4	43	3
44	4	45	4	46	1	47	2	48	2
49	3	50	1	51	1	52	2	53	2
55	1	56	1	57	1	60	1	61	1
65	4	66	1	67	1	71	1	79	2
81	1	86	1	91	1	94	1	95	1
96	1	97	1	102	2	106	1	114	1
124	1	128	1	129	1	132	1	138	1
140	1	179	1	240	1	447	1		

Tabela B.2 – Quantidade de Documentos nas Classes/Grupos Iniciais

$\lvert G_{ij}\rvert$	$\lvert G'_j\rvert$	$\lvert G_{ij}\rvert$	$\lvert G'_j\rvert$	$\lvert G_{ij}\rvert$	$\lvert G'_j\rvert$	$\lvert G_{ij}\rvert$	$\lvert G'_j\rvert$	$\lvert G_{ij}\rvert$	$\lvert G'_j\rvert$
1	130	2	35	3	20	4	10	5	5
6	8	7	3	8	3	9	2	10	2
11	2	12	1	14	2	20	2	47	1
79	1								

Apêndice C. Grupos Finais

A Tabela C.1 apresenta os grupos obtidos após a execução do algoritmo 6, que não descarta documentos nem grupos e realiza a divisão implícita de grupos, organizados pela quantidade de atributos não nulos em seus centróides. Seja $G'_j \subseteq G, \forall G_{ij} \in G'_j \rightarrow |\overrightarrow{\Phi(\vec{c}_{ij})}| = k \wedge \nexists G_{in} \in G'_n \rightarrow |\overrightarrow{\Phi(\vec{c}_{ij})}| = k$, onde $\overrightarrow{\Phi(\vec{c}_{ij})}$ é um vetor composto dos atributos não nulos de $\vec{c}_{ij}$ e todos grupos G_{ij} num mesmo G'_j têm centróides c_{ij} com a mesma quantidade de atributos não nulos. A Tabela C.2 apresenta a quantidade final de documentos em cada grupo. Seja $G'_j \subseteq G, \forall G_{ij} \in G'_j \rightarrow |G_{ij}| = k \wedge \nexists G_{in} \in G'_n \rightarrow |G_{in}| = k$, onde todos grupos G_{ij} num mesmo G'_j têm a mesma quantidade de documentos.

Tabela C.1 – Quantidade de atributos não nulos nas Classes/Grupos Finais

$\lvert\overrightarrow{\Phi(\vec{c}_{ij})}\rvert$	$\lvert G'_j \rvert$	$\lvert\overrightarrow{\Phi(\vec{c}_{ij})}\rvert$	$\lvert G'_j \rvert$
9	2	10	1
12	1	15	3
16	1	17	423
26	4	27	9
28	2	35	1
36	1	37	3
44	1	67	1

Tabela C.2 – Quantidade de Documentos nas Classes/Grupos Finais

$\lvert G_{ij} \rvert$	$\lvert G'_j \rvert$	$\lvert G_{ij} \rvert$	$\lvert G'_j \rvert$
1	362	2	50
3	11	4	10
5	2	6	5
7	3	8	3
9	1	10	2
12	1	14	1
16	1	25	1

Apêndice D. Atributos Descartados Via Índice Normalizado Gini

Tabela D.1 – Atributos descartados

preço	tribunal regional federal (trf)
auto	unanimidade
relatório	contrato de o ficar
cédula	tribunal
voto	provimento
processo	ação
turma	mérito
julgamento	código de processo civil (cpc)
fazenda público	bem
matéria	contra-razão
dia	juízo
exercício	ministério público federal (mpf)
juiz	fato
terceiro	razão
superior tribunal de justiça (stj)	diário de o justiça (dj)
direito	pedido
prazo	lei
precedente	legislação
autor (direito penal)	circunstância agravante
relator	medida
debate	jurisprudência
recurso especial (resp)	ministro
pagamento	processo civil
adotar	agravo de instrumento
estado	parágrafo

Apêndice E. Sobre o Especialista Humano 1

O especialista humano 1 é Bacharel em Ciências Jurídicas pela Universidade Federal do Rio Grande do Sul e atua no gabinete da Vice-Presidência do Tribunal de Justiça do Estado do Rio Grande do sul. Após a validação dos resultados da classificação, manifestou-se da seguinte forma:

"*Luís, se nós tivéssemos um sistema de pesquisa como este que avaliei, mesmo que ele acertasse somente na metade casos, isso ainda seria eliminar metade do trabalho de pesquisa na jurisprudência. Nos casos em que o seu sistema errasse, faríamos a pesquisa nas páginas de busca já oferecidas pelos tribunais. Eu costumo ter que ler algo em torno de 30 a 40 documentos antes de encontrar um que resolva o meu problema.*

Quanto à tua pergunta sobre as ementas, eu acho que mais ou menos 30% delas ou são muito genéricas ou, o que é mais raro, mas também acontece, estão erradas.

Voltando ao teu sistema, quando ele não encontrava nenhum documento que servisse para o caso avaliado, o problema mais comum era que ele me trazia documentos que discutiam assuntos que até tinham a ver com o do caso, mas eram muito genéricos e acabavam não servindo como argumentação. Por exemplo: tinha um caso, do documento 20, que era de 'prescrição intercorrente' e os documentos da pesquisa eram sobre 'prescrição', mas não 'prescrição intercorrente'."

Apêndice F. Sobre o Especialista Humano 2

O especialista humano 2 é Bacharel em Ciências Jurídicas pela Universidade Federal do Rio Grande do Sul e atua como advogado. Após a validação dos resultados da classificação, manifestou-se com as seguintes palavras:

"*Comparando com minha experiência na busca manual nos sites de busca do TJRS, do TRF4, do TRT4, do STJ, do STF e do TST que uso mais comumente, ou não encontrando nenhuma referência para uma certa combinação de parâmetros ou expressões muito específicas ou, mais comum, uma quantidade muito grande (centenas ou milhares) não só para uma expressão mais comum (por exemplo 'conflito de competência') mas também para combinações de expressões (como 'conflito de competência', tributário, federal e estadual'), o sistema que testei pareceu-me poder ajudar bastante. O sistema certamente não ajuda quando aponta um número relativamente grande de alternativas (10) e nenhuma tem relação com o problema específico; mas se um em 10 tiver relação, isso ajuda mais do que as várias dezenas ou mesmo centenas de casos muito genéricos que os sistemas de busca indicarão numa primeira tentativa. E quando apresentou relativamente poucos documentos (1, 2, 3, 4, 5...) e todos ou quase todos tiverem relação isso foi realmente espantoso.*

Se o sistema indicasse uma lista de probabilidade de pertinência ou algo assim, poderia mesmo apontar um número maior de ocorrências, mas, reitero, o importante não é que me aponte uma enormidade de casos (que é o problema mais comum), mas uma certa quantidade de casos com maior probabilidade de serem pertinentes, mesmo que os documentos usem expressões diferentes, que é o maior problema das pesquisas textuais (que é um problema 'invisível', já que, apesar da enormidade de casos que uma pesquisa comum costuma dar, mesmo com vários parâmetros, casos pertinentes escritos com expressões sutilmente diferente simplesmente não aparecem nas pesquisas).

Acho que aumentará a produtividade dos escritórios de advocacia. Na justiça federal ou ao menos no trf4 o trabalho de carregador de processos já se está extinguindo - só para os processos antigos. Então, os estagiários ou advogados iniciantes poderão ser melhor alocados para fazer pesquisa de jurisprudência e, com um sistema desses (não sei se seria privado ou estatal, ou dos próprios tribunais), poderiam 'produzir' muito mais."

Anexo A. Teor do documento Nº 50

1. Quatro são os requisitos para a concessão do benefício em tela: (a) a qualidade de segurado do requerente; (b) o cumprimento da carência de 12 contribuições mensais, (c) a superveniência de moléstia incapacitante para o desenvolvimento de qualquer atividade que garanta a subsistência, (d) o caráter definitivo da incapacidade.

2. Tratando-se de deficiência física congênita, e inexistindo evidência de que, à época de sua filiação ao RGPS, o autor reunisse plena capacidade laboral, é de ser indeferido o benefício pretendido.

Vistos e relatados estes autos em que são partes as acima indicadas, decide a Colenda Turma Suplementar do Tribunal Regional Federal da 4ª Região, por unanimidade, negar provimento à apelação, nos termos do relatório, votos e notas taquigráficas que ficam fazendo parte integrante do presente julgado.

Marcos da Rosa ajuizou ação ordinária contra o INSS, em 02/12/2005, objetivando a concessão de auxílio-doença ou, alternativamente, aposentadoria por invalidez, a contar do cancelamento do benefício de auxílio-doença, em 30/11/2005.

Sentenciando, o MM. Juízo a quo julgou improcedente o pedido, condenando o autor ao pagamento das custas e honorários advocatícios, estes fixados em R$ 650,00, suspendendo a exigibilidade de tais verbas em razão da AGJ concedida.

Irresignado, apelou o demandante. Em suas razões, sustenta que a amputação de um dos seus braços foi comprovada nos autos, sendo evidente sua incapacidade para a função de agricultor.

Com contra-razões, vieram os autos a esta Corte.

É o relatório.

À revisão.

Dos requisitos para a concessão do benefício

Quanto à aposentadoria por invalidez, reza o art. 42 da Lei 8.213/91:

"Art. 42. A aposentadoria por invalidez, uma vez cumprida, quando for o caso, a carência exigida, será devida ao segurado que, estando ou não em gozo de auxílio-doença, for considerado incapaz e insuceptível de reabilitação para o exercício de atividade que lhe garanta a subsistência, e ser-lhe-á paga enquanto permanecer nesta condição."

Já no que tange ao auxílio-doença, dispõe o art. 59 do mesmo diploma:

"Art. 59 - O auxílio-doença será devido ao segurado que, havendo cumprido, quando for o caso, o período de carência exigido nesta Lei, ficar incapacitado para o seu trabalho ou para a sua atividade habitual por mais de 15 dias consecutivos."

Por sua vez, estabelece o art. 25:

"Art. 25. A concessão das prestações pecuniárias do Regime Geral de Previdência Social depende dos seguintes períodos de carência:

I - auxílio-doença e aposentadoria por invalidez: 12 contribuições mensais;"

Da análise dos dispositivos acima elencados, pode-se concluir que quatro são os requisitos para a concessão do benefício em tela: (a) a qualidade de segurado do requerente; (b) o cumprimento da carência de 12 contribuições mensais; (c) a superveniência de moléstia incapacitante para o desenvolvimento de atividade laboral que garanta a subsistência, e (d) o caráter permanente da incapacidade (para o caso da aposentadoria por invalidez) ou temporário (para o caso do auxílio-doença).

Ainda quanto ao tema, algumas observações fazem-se necessárias:

Em primeiro lugar, no que toca à qualidade de segurado, caso o requerente cesse o recolhimento das contribuições, devem ser observadas as regras constantes no art. 15 e parágrafos:

"Art. 15. Mantém a qualidade de segurado, independentemente de contribuições:

I - sem limite de prazo, quem está em gozo de benefício;

II - até 12 (doze) meses após a cessação das contribuições, o segurado que deixar de exercer atividade remunerada abrangida pela Previdência Social ou estiver suspenso ou licenciado sem remuneração;

III - até 12 (doze) meses após cessar a segregação, o segurado acometido de doença de segregação compulsória;

IV - até 12 (doze) meses após o livramento, o segurado retido ou recluso;

V - até 3 (três) meses após o licenciamento, o segurado incorporado às Forças Armadas para prestar serviço militar;

VI - até 6 (seis) meses após a cessação das contribuições, o segurado facultativo.

§ 1º O prazo do inciso II será prorrogado para até 24 (vinte e quatro) meses se o segurado já tiver pago mais de 120 (cento e vinte) contribuições mensais sem interrupção que acarrete a perda da qualidade de segurado.

§ 2º Os prazos do inciso II ou do § 1º serão acrescidos de 12 (doze) meses para o segurado desempregado, desde que comprovada essa situação pelo registro no órgão próprio do Ministério do Trabalho e da Previdência Social.

§ 3º Durante os prazos deste artigo, o segurado conserva todos os seus direitos perante a Previdência Social.

§ 4º A perda da qualidade de segurado ocorrerá no dia seguinte ao do término do prazo fixado no Plano de Custeio da Seguridade Social para recolhimento da contribuição referente ao mês imediatamente posterior ao do final dos prazos fixados neste artigo e seus parágrafos."

Quanto à carência, é de ser observada a regra constante no parágrafo único do art. 24: "Havendo perda da qualidade de segurado, as contribuições anteriores a essa data só serão computadas para efeito de carência depois que o segurado contar, a partir da nova filiação à Previdência Social, com, no mínimo, 1/3 do número de contribuições exigidas para o cumprimento da carência definida para o benefício a ser requerido. Dessa forma, cessado

o vínculo, eventuais contribuições anteriores à perda da condição de segurado somente poderão ser computadas se cumpridos mais quatro meses, nos termos do dispositivo acima transcrito.

Quanto à inaptidão laboral, a inteligência do § 2º do art. 42 admite a concessão do benefício ainda que a enfermidade seja anterior à filiação, desde que o impedimento para o trabalho decorra de progressão ou agravamento da doença ou lesão.

Por fim, tenho que os benefícios de auxílio-doença e aposentadoria por invalidez são fungíveis, sendo facultado ao julgador (e, diga-se, à Administração), conforme a espécie de incapacidade constatada, conceder um deles, ainda que o pedido tenha sido limitado ao outro. Dessa forma, o deferimento do amparo nesses moldes não configura julgamento ultra ou extra petita.

Da comprovação da incapacidade

Tratando-se de aposentadoria por invalidez ou auxílio-doença, o Julgador firma a sua convicção, via de regra, por meio da prova pericial.

Além disso, o caráter da incapacidade, a privar o segurado do exercício de todo e qualquer trabalho, deve ser avaliado conforme as circunstâncias do caso concreto. Isso porque não se pode olvidar de que fatores relevantes - como a faixa etária do requerente, seu grau de escolaridade, assim como outros - são essenciais para a constatação do impedimento laboral.

Em tal sentido, já se manifestou a Terceira Seção desta Corte:

"EMBARGOS INFRINGENTES. PREVIDENCIÁRIO. CONCESSÃO DE AUXÍLIO-DOENÇA E APOSENTADORIA POR INVALIDEZ. INCAPACIDADE PARCIAL. PERÍCIA.

1. Comprovado pelo conjunto probatório que a parte autora é portadora de enfermidades que a incapacitam total e permanentemente para o trabalho agrícola, considerados o quadro clínico e as condições pessoais, é de ser concedida a aposentadoria por invalidez, ainda que a perícia mencione que a incapacidade laborativa seja parcial, pois não incapacita para atividades que não exijam esforço físico.

2. É imprescindível considerar além do estado de saúde, as condições pessoais do segurado, como a sua idade, a presumível pouca instrução, a limitada experiência laborativa e, por fim, a realidade do mercado de trabalho atual, já exíguo até para pessoas jovens e que estão em perfeitas condições de saúde."(EIAC 1998.04.01.053910-7, Rel. João Batista Pinto Silveira, DJU 1º-3-2006).

Do caso dos autos

Durante a instrução processual foi realizada perícia médica pelo Departamento Médico Judiciário do TJRS, em 28/03/2007 (fls. 58/59), cujo laudo técnico explicita e conclui:

a- enfermidade: agenesia do antebraço esquerdo;

b- incapacidade: existente;

c- grau da incapacidade: parcial;

d- prognóstico da incapacidade: definitiva;

e - início da incapacidade: deformidade congênita.

Referiu o expert, ainda, que o autor apresenta limitação da capacidade laborativa e encontra-se apto para o exercício de atividades laborativas que não necessitem do emprego de função bi-manual.

Do exame dos autos, constata-se, ainda, a concessão de auxílio-doença no período de 27/02/2005 a 30/01/2006, CID Q 71.2 (fl. 27, c/c fl. 39).

Do preenchimento dos requisitos

Os requisitos carência e condição de segurado foram cumpridos, tendo em vista a concessão de auxílio-doença anterior, além de não se tratar de matéria controvertida nos autos.

No que diz respeito à incapacidade, esta é claramente preexistente, pois, conforme consta no laudo médico pericial, trata-se de deficiência física congênita, inexistindo evidência de que, à época de sua filiação ao RGPS, o autor reunisse plena capacidade laboral.

Ressalte-se que, no sistema previdenciário vigente, em que a filiação à Previdência Social decorre de ato da exclusiva vontade desta e sem prévio exame médico, caberia ao autor produzir prova robusta de que, por ocasião de sua filiação ao Regime Geral de Previdência Social, não era incapaz para os seus afazeres e que, posteriormente, quedou-se inapto para aquelas próprias tarefas. Contudo, não provou tal ocorrência.

Por esses fundamentos, é de ser mantida a decisão de origem.

Dispositivo

Ante o exposto, voto por negar provimento à apelação, nos termos da fundamentação.

Anexo B. Teor do documento Nº 17

1. Tratando-se de auxílio-doença ou aposentadoria por invalidez, o Julgador firma sua convicção, via de regra, por meio da prova pericial.

2. Considerando a sugestão do perito judicial neurológico, no sentido de que o autor não trabalhe em locais onde possa cortar-se (dando como exemplo o "açougue"), e tendo em vista a profissão do autor ser justamente a de "açougueiro", restou caracterizada a incapacidade do segurado para suas atividades habituais. Assim, é devido ao requerente o benefício de auxílio-doença até sua efetiva recuperação ou reabilitação.

3. Quanto ao termo inicial do auxílio-doença, deve ser fixado na data do requerimento administrativo do benefício nº 117.788.243-1 (05-07-2000), com o pagamento das parcelas devidas desde então, observando-se os valores já pagos na via administrativa.

Vistos e relatados estes autos em que são partes as acima indicadas, decide a Egrégia 5ª Turma do Tribunal Regional Federal da 4ª Região, por unanimidade, dar provimento à apelação da parte autora nos termos do relatório, votos e notas taquigráficas que ficam fazendo parte integrante do presente julgado.

Algemiro de Castro Agne ajuizou ação previdenciária contra o INSS, postulando o restabelecimento do benefício de auxílio-doença ou a concessão de aposentadoria por invalidez, desde o requerimento administrativo do benefício nº 31/117.788.243-1, tendo em vista padecer de epilepsia e atrofia cerebral, que o incapacitam para o exercício de atividades laborativas. Requereu, ainda, a antecipação dos efeitos da tutela.

À fl. 23, foi indeferido o pedido antecipatório.

Citado, o Instituto Previdenciário apresentou sua contestação.

Às fls. 66/71, foram juntados os laudos médicos judiciais.

Na sentença (26-05-2008), a magistrada a quo julgou improcedente o pedido da parte autora, condenando-a ao pagamento das custas processuais e dos honorários advocatícios, estes fixados em R$ 900,00, cuja exigibilidade, contudo, restou suspensa em virtude do benefício de AJG concedido.

Em suas razões de apelação, o autor sustentou que, em virtude de seus ataques epilépticos, está incapacitado para suas atividades habituais de açougueiro. Assim, requereu o julgamento de procedência do pedido.

Apresentadas as contra-razões, vieram os autos a esta Corte para julgamento.

É o relatório.

À revisão.

Inicialmente, cabe referir que a qualidade de segurado e a carência mínima exigidas para a concessão dos benefícios requeridos não restaram questionadas nos autos. Ademais, o próprio INSS reconheceu o preenchimento de tais requisitos quando concedeu à parte autora o benefício de auxílio-doença, nos períodos de 19-07-1999 a 11-04-2000, 18-08-

2000 a 08-06-2001, 21-08-2001 a 30-11-2001, 28-03-2002 a 30-04-2004, 07-06-2004 a 12-11-2004, 13-04-2005 a 13-06-2005, e também concede-lhe auxílio-doença, desde 28-09-2005 (com DIB em 24-02-2005), sem data prevista para cessação, conforme consulta ao sistema Plenus, cujos extratos determino a juntada aos autos. Assim, tenho esses requisitos por incontroversos.

Resta, pois, averiguar a existência de incapacidade laboral que justifique a concessão dos benefícios postulados.

Tratando-se de auxílio-doença ou aposentadoria por invalidez, o Julgador firma sua convicção, via de regra, por meio da prova pericial.

No caso concreto, foram realizadas duas perícias médicas, cujos laudos foram juntados às fls. 66/71. O primeiro, efetuado em 16-06-2005 por especialista em neurologia do DMJ, apresentou as seguintes "história"e "conclusão":

"HISTÓRIA

O periciado refere que há cerca de 05 anos iniciou com crises convulsivas (SIC) caracterizadas pelo 'lado esquerdo se retorcer' com subseqüente perda de consciência. Diz que teve quadro semelhante à época que estava no exército (serviço obrigatório).

Atualmente diz que faz uso de Carbamazepina, Fenitoína e Gardenal, além de usar Imipramina para depressão. Conta que tem de 03 a 04 'ataques' por mês, mesmo com o uso dos remédios acima descritos.

Em relação ao trabalho, exercia a função de açougueiro, tendo parado de trabalhar em dezembro de 2001.

Na revisão neurológica, o periciado conta (sic) teve 01 a 02 crises nos últimos meses, mas esteve sem crises antes. Relata que fazendo o uso adequado dos medicamentos acima mantém as crises parcialmente controladas (SIC)."

(...)

CONCLUUSÃO

Do ponto de vista neurológico, o periciado é portador de epilepsia generalizada (CID10: G40), porém sem caracterizar epilepsia de difícil controle. Portanto, está apto a exercer atividades laborativas remuneradas. Porém, deve-se atentar que, para zelar pelo bem estar do paciente com epilepsia, sugere-se não trabalhar ou ter atividades de lazer em locais onde possa, caso tenha crise convulsiva, se queimar (ex. com caldeiras ou fornos), se cortar (ex. serralheria, açougue), se afogar (salva-vidas) ou cair de alturas (ex. postes de luz)."

O segundo laudo, realizado por psiquiatra (também do DMJ), teve, por sua vez, a seguinte conclusão:

"(...)O periciado não apresenta patologia psiquiátrica que o incapacite para as atividades(...)".

Considerando, pois, a sugestão do perito judicial neurológico, no sentido de que o autor não trabalhe em locais onde possa cortar-se (dando como exemplo o "açougue"), e tendo

em vista a profissão do autor ser justamente a de "açougueiro"(conforme foi relatado em ambas as perícias e, também, consoante consta no sistema Plenus), entendo que restou caracterizada a incapacidade do segurado para suas atividades habituais. Assim, é devido ao requerente o benefício de auxílio-doença até sua efetiva recuperação ou reabilitação.

Vale ressaltar que o autor esteve em gozo de auxílio-doença nos períodos de 19-07-1999 a 11-04-2000 (NB 113.565.978-5), 18-08-2000 a 08-06-2001 (NB 117.788.243-1), 21-08-2001 a 30-11-2001 (NB 121.929.197-5), 28-03-2002 a 30-04-2004 (NB 508.004.198-2), 07-06-2004 a 12-11-2004 (NB 508.226.434-2), 13-04-2005 a 13-06-2005 (NB 514.044.244-9), sendo que na concessão dos referidos benefícios (exceto no último) foi constatada, pelo corpo médico da Autarquia, a moléstia de epilepsia como causa incapacitante do segurado.

Quanto ao termo inicial do auxílio-doença, penso que deve ser fixado na data do requerimento administrativo do benefício nº 117.788.243-1 (05-07-2000), devendo o INSS pagar as parcelas vencidas desde então, observando-se os valores já pagos na via administrativa nos intervalos antes mencionados.

A atualização monetária, a partir de maio de 1996, deve-se dar pelo IGP-DI, de acordo com o art. 10 da Lei nº 9.711/98, combinado com o art. 20, §§5º e 6º, da Lei nº 8.880/94.

Os juros de mora devem ser fixados à taxa de 1% ao mês, a contar da citação, com base no art. 3º do Decreto-Lei nº 2.322/87, aplicável analogicamente aos benefícios pagos com atraso, tendo em vista o seu caráter eminentemente alimentar, consoante firme entendimento consagrado na jurisprudência do STJ e na Súmula 75 desta Corte.

Vencido na lide, deve o INSS arcar com os ônus de sucumbência.

Os honorários advocatícios devem ser fixados em 10% sobre o valor das parcelas devidas até a data de julgamento do presente acórdão, a teor das Súmulas 111 do STJ e 76 desta Corte.

No que toca aos honorários periciais, fixo-os em R$ 234,80 (para cada perícia), nos termos da Resolução 440/05 do CJF.

Tendo o feito tramitado perante a Justiça Estadual gaúcha, deve a Autarquia responder pela metade das custas devidas, consoante a Súmula 2 do extinto Tribunal de Alçada do Rio Grande do Sul e o art. 11, a, da Lei Estadual gaúcha n. 8.121/85.

Ante o exposto, voto por dar provimento à apelação da parte autora.