UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

AGRUPAMENTO DE DOCUMENTOS FORENSES UTILIZANDO REDES NEURAIS ART1

GEORGER ROMMEL FERREIRA DE ARAÚJO

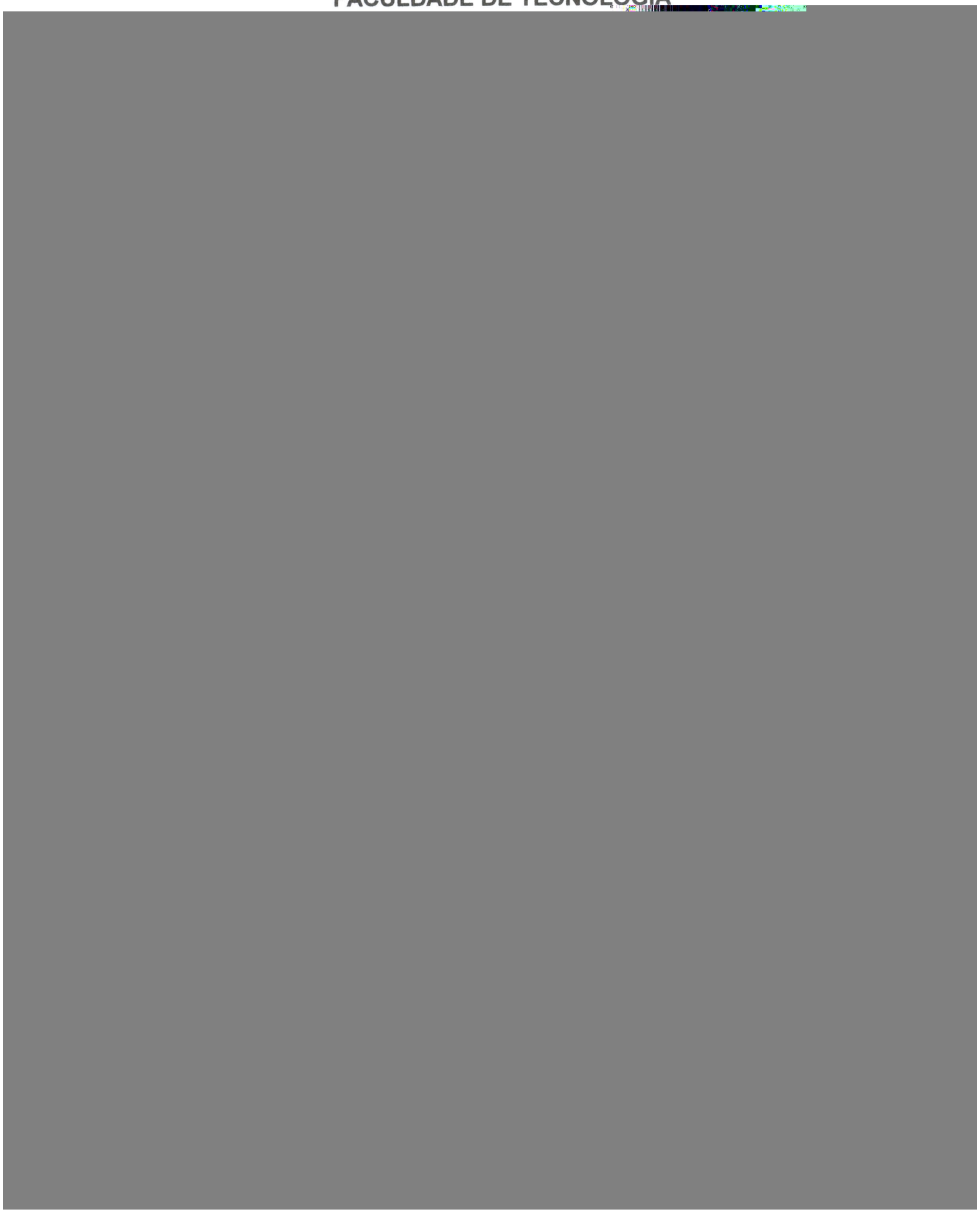
ORIENTADORA: Profª. Drª. CÉLIA GHEDINI RALHA

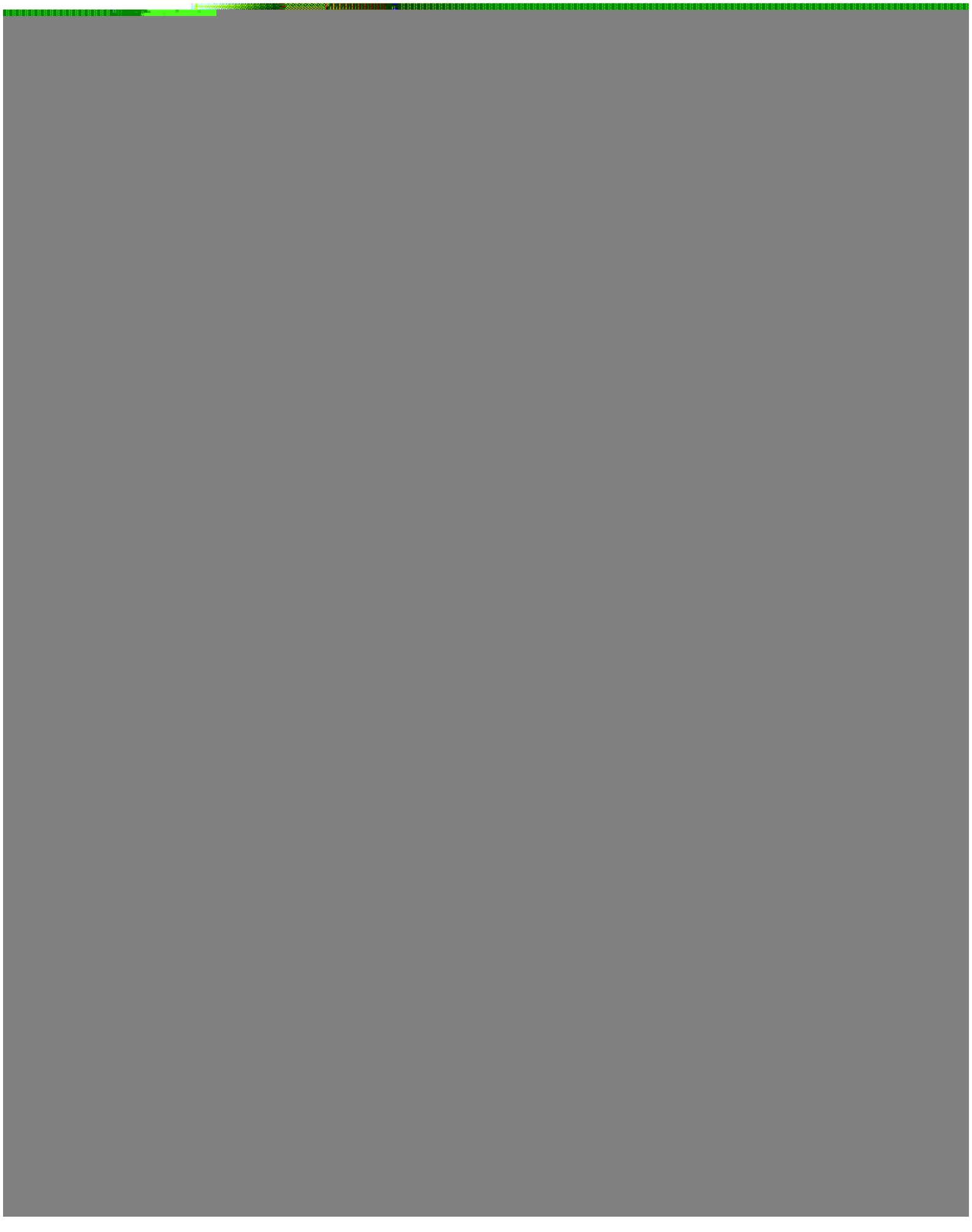
**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
ÁREA DE CONCENTRAÇÃO INFORMÁTICA FORENSE E
SEGURANÇA DA INFORMAÇÃO**

PUBLICAÇÃO: ENE/PG.DM – 082 A/2011

BRASÍLIA/DF: NOVEMBRO/2011

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA





Para Maria Júlia e Déborah, que fazem tudo valer a pena.

AGRADECIMENTOS

A Deus.

À minha orientadora Profª. Drª. Célia Ghedini Ralha, pelo constante apoio, incentivo, dedicação e amizade essenciais para o desenvolvimento deste trabalho e para o meu desenvolvimento como pesquisador.

Ao colega Nassif, pelas sugestões e contribuições fundamentais.

Ao colega Werneck, pelas discussões construtivas, palavras sábias e pela paciência.

Aos colegas Cristiano, Augusto, Jean, Weder e Dalben, pela ajuda nos experimentos.

A todos os colegas do Mestrado em Informática Forense, pela amizade.

Aos Professores Riverson e Araújo, e aos amigos Jesus e Renilton, pela confiança demonstrada através das cartas de recomendação.

A Tomáš Hudík, autor do pacote de programas ART para aprendizado não-supervisionado; a Holger Arndt, autor da biblioteca Universal Java Matrix Package (UJMP); à Apache Software Foundation (ASF), mantenedora dos projetos Tika e Lucene; e à Oracle Corporation, patrocinadora do projeto NetBeans, que foram fundamentais para a concretização deste trabalho.

O presente trabalho foi realizado com o apoio do Departamento de Polícia Federal – DPF, com recursos do Programa Nacional de Segurança Pública com Cidadania – PRONASCI, do Ministério da Justiça.

RESUMO

AGRUPAMENTO DE DOCUMENTOS FORENSES UTILIZANDO REDES NEURAIS ART1

Autor: GEORGER ROMMEL FERREIRA DE ARAÚJO
Orientadora: Profª. Drª. CÉLIA GHEDINI RALHA
Programa de Pós-graduação em Engenharia Elétrica
Brasília, novembro de 2011

Coleções textuais de Informática Forense são normalmente muito heterogêneas. Embora técnicas de classificação, por tipo de arquivo ou outros critérios, possam auxiliar na exploração dessas coleções textuais, elas não ajudam a agrupar documentos com conteúdo assemelhado. A Teoria da Ressonância Adaptativa (*Adaptive Resonance Theory* – ART) descreve várias Redes Neurais Artificiais auto-organizáveis que utilizam um processo de aprendizado não-supervisionado e são especialmente projetadas para resolver o dilema da estabilidade/plasticidade. Este trabalho aplica o algoritmo ART1 (ART com vetores de entrada binários) para agrupar tematicamente documentos retornados de uma ferramenta de busca utilizada com coleções textuais forenses. Documentos que antes seriam apresentados em uma lista desorganizada e frequentemente longa passam a ser agrupados por conteúdo, oferecendo ao perito uma forma organizada de obter uma visão geral do conteúdo dos documentos durante o exame pericial. Os resultados experimentais são indicativos da validade da abordagem proposta, obtendo uma correspondência adequada entre a solução de agrupamento processada com o protótipo de aplicação desenvolvido e as classes-padrão definidas por um especialista.

ABSTRACT

COMPUTER FORENSIC DOCUMENT CLUSTERING WITH ART1 NEURAL NETWORKS

Author: GEORGER ROMMEL FERREIRA DE ARAÚJO
Advisor: Profª. Drª. CÉLIA GHEDINI RALHA
Programa de Pós-graduação em Engenharia Elétrica
Brasília, November of 2011

Computer forensic text corpora are usually very heterogeneous. While classification, by file type or other criteria, should be an aid in the exploration of such corpora, it does not help in the task of grouping together documents thematically. Adaptive Resonance Theory (ART) describes a number of self-organizing artificial neural networks that employ an unsupervised learning process and are specially designed to learn new patterns without forgetting what they have already learned, overcoming the important restriction defined by the stability/ plasticity dilemma. This work applies the ART1 algorithm (ART with binary input vectors) to thematically cluster documents returned from a query tool used with forensic text corpora. Documents that would previously be presented in a disorganized and often long list are thematically clustered, giving the examiner an organized way of obtaining a general picture of document content during forensic examinations. Experimental results validated the approach, achieving adequate agreement between the clustering solution processed with the developed prototype software package and the gold standard defined by a domain specialist.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE SÍMBOLOS, NOMENCLATURA E ABREVIAÇÕES

ART *Adaptive Resonance Theory*. 53, 55, 59, 85, 124

ART1 ART com padrões de entrada binários. 3–5, 38, 40, 55, 57–59, 84, 85, 119, 120, 122–125

Autopsy Interface gráfica para o TSK. 1, 28, 32, 37

bookmark Conjuntos temáticos onde o perito organiza os arquivos considerados relevantes. 3, 23, 106

BOW *Bag-of-words* (literalmente "saco de palavras"). Modelo que representa um documento ou uma coleção de documentos como um conjunto não-ordenado de palavras, desconsiderando a ordem e o contexto destas últimas. 39, 40, 68, 70, 73, 76, 121

EnCase Ferramenta comercial de Informática Forense da Guidance Software. 1, 24, 28, 32, 37

FTK *Forensic Toolkit*. Ferramenta comercial de Informática Forense da AccessData. 1, 24, 28, 29, 37, 68

hit Resultado retornado por uma busca textual. 1, 2, 4–6, 8, 25, 28, 32, 34, 37, 45, 63, 80, 82–85, 87, 88, 94, 101, 105, 106, 120–123

precisão Métrica usada em Recuperação de Informação que mede a porcentagem dos documentos relevantes para a consulta, entre aqueles retornados pela consulta. 39

recall Métrica usada em Recuperação de Informação que mede a porcentagem dos documentos relevantes para a consulta, de toda a coleção, que são efetivamente retornados pela consulta. 2, 39, 67, 123

RNAs Redes Neurais Artificiais. 2–5, 51–53, 55, 57, 82, 84, 85, 119, 120, 122, 124, 125

SOM *Self-Organizing Map*. 2, 4, 53, 82, 84, 85, 123

TSK *The Sleuth Kit.* Biblioteca e coleção de ferramentas de Informática Forense de código aberto. 1, 28, 32, 37

WWW *World Wide Web.* 37, 53

CAPÍTULO 1

INTRODUÇÃO

Este capítulo visa apresentar as principais características do trabalho de pesquisa realizado. Na Seção 1.1 são apresentados o problema tratado e as justificativas do tema com a abordagem adotada; na Seção 1.2 são descritos a hipótese e os objetivos do trabalho; na Seção 1.3 é descrito o método empregado; na seção 1.4 são elencados os resultados esperados e as limitações existentes na pesquisa.

1.1 PROBLEMA TRATADO E JUSTIFICATIVA

Esta dissertação aborda o problema da organização dos *hits* (resultados retornados) por buscas em coleções textuais forenses através de grupos de documentos com conteúdo assemelhado. A técnica proposta é a de agrupamento de documentos, descrita em detalhe na Seção 3.3.1.

A área de estudo de Informática Forense, embora em expansão, ainda pode ser considerada uma atividade de nicho, praticada principalmente por forças policiais e militares e agências de inteligência. Nesta direção, uma pesquisa realizada entre examinadores da área de Informática Forense publicada por Kuncik (2010) apontou que as ferramentas proprietárias mais utilizadas pelos participantes eram FTK (AccessData, 2011) e *EnCase* (Guidance, 2011). Uma ferramenta livre popular é a combinação TSK/*Autopsy* (Carrier, 2011), cujos *downloads* somados totalizavam 316.037 em 17 de agosto de 2011[1]. Todas estas ferramentas permitem ao usuário efetuar buscas textuais nas evidências examinadas, mas não oferecem nenhum recurso para organizar em grupos os *hits* que possuam conteúdo assemelhado. O

[1] `http://sourceforge.net/projects/sleuthkit/files/stats/timeline?dates=2002-06-12+to+2011-08-17`, acesso em 18/08/2011.

FTK categoriza os *hits* por tipo de arquivo (documentos produzidos por processadores de texto, planilhas eletrônicas e mensagens de correio eletrônico, entre outros), e as demais ferramentas apenas listam os *hits* para que o perito os revise.

As buscas retornam todos os arquivos que contêm os termos de busca. Dada a natureza tipicamente vasta e não-estruturada das coleções textuais encontradas nas mídias apreendidas, cada busca retorna grande quantidade de *hits*, não raro na casa das centenas e dos milhares, o que caracteriza sobrecarga de informação (Beebe e Clark, 2007). Os *hits* não são classificados nem ordenados com base em seu conteúdo. Quando o perito submete uma busca à aplicação e, ao examinar a lista de *hits*, localiza um documento de interesse, não tem alternativa senão examinar todos os demais *hits* para que possa localizar outros documentos de conteúdo semelhante que porventura existam.

Uma possível solução seria descartar um certo número de *hits* e assim diminuir a quantidade de arquivos a examinar, ou seja, reduzir o valor do *recall*[2] intencionalmente. Essa solução, entretanto, não se presta ao domínio da Informática Forense, onde um único *hit* pode se mostrar decisivo para o sucesso do exame pericial. Por esse motivo, todos os *hits* devem ser retornados. Torna-se necessário, portanto, desenvolver técnicas e funcionalidades que permitam ao perito revisar os *hits* de forma mais rápida, sem que para isso necessitem reduzir o *recall*.

O agrupamento dos *hits* retornados por buscas textuais foi testado, com resultados considerados positivos, em experimentos realizados com sistemas tradicionais de Recuperação de Informação (Leuski, 2001; Leuski e Allan, 2000) e em ambiente *World Wide Web* (Zeng et al., 2004). Assim, seu uso parece promissor no domínio da Informática Forense.

1.1.1 AGRUPAMENTO OU CLASSIFICAÇÃO

Uma forma de materializar a abordagem acima descrita seria através do agrupamento dos *hits*. Se o perito, ao submeter uma busca textual na aplicação, receber como retorno grupos de *hits* com conteúdo assemelhado, poderá avaliar apenas uma pequena quantidade de *hits* em um determinado grupo e julgar previamente sua relevância. Se o grupo for considerado relevante, poderá ser marcado como importante para uma análise mais aprofundada; caso contrário, poderá receber uma menor prioridade para que seja revisado depois dos grupos julgados mais importantes, podendo ser até mesmo descartado. Como trabalho de pesquisa foi encontrado Beebe (2007), que aplicou Redes Neurais Artificiais (RNAs) *Self-Organizing*

[2]Métrica usada em Recuperação de Informação que mede a porcentagem dos documentos relevantes para a consulta, de toda a coleção, que são efetivamente retornados pela consulta.

Map (SOM) e considerou os resultados promissores, embora tenha também relatado uma série de problemas e limitações de sua abordagem.

Segundo Andrews e Fox (2007), agrupamento de documentos (*document clustering*) é baseado em aprendizado de máquina não-supervisionado, e tem por objetivo descobrir grupamentos naturais de documentos assemelhados com o fim de apresentar uma visão geral das classes (tópicos) presentes em uma coleção de documentos. Não se deve confundir agrupamento com classificação, que é uma técnica baseada em aprendizado de máquina supervisionado, na qual as classes são conhecidas previamente. Assim, classificação não é aplicável ao domínio de aplicação desta pesquisa porque os *bookmarks* (conjuntos temáticos) que conterão os arquivos considerados relevantes para o exame não são conhecidos de antemão, e cada exame tem suas particularidades; os *bookmarks* obtidos em um caso não servem para outro caso não-relacionado. Por esse motivo, este trabalho investigará a técnica de agrupamento.

Não é possível a um algoritmo de agrupamento gerar os "melhores grupos possíveis" porque podem existir várias formas corretas de dispor em grupos os arquivos de uma coleção textual, e nenhuma delas é necessariamente melhor que a outra; são simplesmente diferentes, e a mesma disposição pode ser considerada boa em um dado contexto, porém ruim em outro. Além disso, o mesmo exame pericial realizado por peritos diferentes pode resultar em *bookmarks* diferentes, mas ainda assim satisfatórios ao esclarecimento do fato em apuração. A função do agrupamento limita-se a oferecer grupos de arquivos com contéudo assemelhado ao perito, a quem cabe a tarefa de avaliar se os arquivos contidos em cada grupo são ou não relevantes para o exame. Desse modo, embora seja difícil ou mesmo impossível obter uma solução de agrupamento ótima, o perito pode fazer uso de uma solução considerada satisfatória, que o ajude a responder de forma esclarecedora e juridicamente aceitável os questionamentos que lhe são apresentados.

Decherchi et al. (2009) realizaram um estudo onde foram agrupadas, através do algoritmo k-médias, partes de uma coleção textual forense composta de mensagens de correio eletrônico. Os termos considerados mais descritivos entre os 20 mais frequentes de cada grupo foram então selecionados, e a partir deles o autor buscou interpretar os principais tópicos de cada grupo. Esse trabalho mostra que o agrupamento pode ser usado como técnica de análise de dados exploratória, na qual as hipóteses são formuladas a partir da análise dos dados (Tukey, 1980).

1.1.2 AGRUPAMENTO COM RNAs ART1

A Teoria da Ressonância Adaptativa (*Adaptive Resonance Theory* – ART) descreve uma série de RNAs auto-organizáveis que utilizam um processo de aprendizado não-supervisionado para executar tarefas de agrupamento de dados. As RNAs ART1 (ART com padrões de entrada binários) especificamente realizam agrupamento de padrões de entrada (vetores) binários, e apresentam propriedades que as tornam particularmente aptas para o objetivo proposto neste trabalho, quais sejam:

- Não exigem que o número desejado de grupos seja informado de antemão, ao contrário de outros algoritmos (k-médias e SOM), e determinam dinamicamente o número de grupos de acordo com a estrutura dos padrões de entrada apresentados à RNA;
- São especialmente projetadas para resolver o dilema da estabilidade/plasticidade, descrito na Subseção 3.4.2.

Há alguns trabalhos publicados sobre agrupamento de texto com RNAs ART1, vários deles de autoria de Massey (2002, 2003, 2005b,a), que dedicou sua tese de doutorado a esse tema e usou coleções de notícias e de resumos de artigos científicos para validar sua proposta.

No entanto, até onde foi possível investigar, a presente pesquisa é a primeira a tratar de agrupamento de *hits* em Informática Forense com uso de RNAs ART1, tornando difícil a comparação de resultados dos experimentos realizados.

1.2 HIPÓTESE E OBJETIVOS

A hipótese desta pesquisa é que o uso de RNAs ART1 pode ser útil para agrupar documentos com conteúdos assemelhados retornados por buscas por palavras-chave em coleções textuais de Informática Forense, utilizando-se para validação uma avaliação dupla, envolvendo aspectos quantitativos e qualitativos.

O objetivo geral deste trabalho é investigar a aplicabilidade de RNAs, com uso de Teoria da Ressonância Adaptativa (*Adaptive Resonance Theory*, ou ART) para agrupamento dos *hits* com conteúdos assemelhados retornados por buscas em coleções textuais forenses. A proposta visa oferecer ao perito uma forma de obter uma visão geral do conteúdo dos documentos retornados pelas buscas – facilitando, em última análise, a obtenção de informações que possam demonstrar ou refutar a materialidade, a autoria e a dinâmica de

uma suposta conduta criminosa. Para alcançar o objetivo geral, definem-se cinco objetivos específicos:

1. Mostrar as limitações dos atuais métodos para revisão de *hits* em coleções textuais forenses;

2. Implementar um método de pré-processamento para transformar os arquivos das coleções textuais em representações tratáveis pelas RNAs ART1;

3. Desenvolver um protótipo de aplicação de busca e agrupamento de *hits* com conteúdo assemelhado em coleções textuais forenses que utilize o método proposto;

4. Realizar experimentações com o uso do protótipo e dados provenientes de casos reais para validar o método de agrupamento de conteúdos assemelhados em coleções textuais forenses;

5. Validar os resultados das experimentações através de parâmetros objetivos (análise quantitativa), além das intrínsecas características subjetivas de conhecimento dos especialistas (análise qualitativa).

1.3 MÉTODO

O método definido para o desenvolvimento desta pesquisa envolveu diversas atividades. A primeira delas foi a revisão da literatura relacionada à Informática Forense e tratamento de informação com foco no processamento de texto, o que permitiu identificar o estado da arte, os trabalhos relacionados e as possíveis soluções para o problema de agrupamento de conteúdos assemelhados em coleções textuais. Com base na revisão, foram elencadas as alternativas existentes, dentre as quais foram selecionadas as RNAs ART1, por serem auto-organizáveis e resolverem o dilema da estabilidade/plasticidade de forma coerente, conforme será apresentado em maior detalhe na Seção 3.4.2.

Após a seleção do algoritmo ART1, foi estudada a forma de como aplicá-lo ao domínio da Informática Forense. Para tanto, foi desenvolvido um protótipo de aplicação composto de dois módulos. O primeiro deles, chamado *indexador*, teve o propósito de executar as etapas de pré-processamento: filtragem dos arquivos textuais por tipo e idioma, remoção de *stopwords*, *stemming* e indexação do conteúdo textual e metadados.

Concluído o estudo e a definição do método, em seguida foi desenvolvido o protótipo da aplicação de busca e agrupamento de *hits*. Os primeiros testes evidenciaram a necessidade de melhorar o desempenho, que resultou na adaptação do algoritmo para utilizar uma estrutura de dados que reduziu o tempo de execução, tornando viável sua aplicação no domínio da Informática Forense, que é composto por grandes volumes de conteúdos textuais.

O protótipo com o algoritmo adaptado foi aplicado a coleções forenses de casos reais do Departamento de Polícia Federal nos quais foram investigadas supostas fraudes em licitações públicas e saques do Programa Bolsa-Família[3]. Foram calculados índices de validação e colhidas avaliações subjetivas para validar os grupos obtidos. Os resultados obtidos nesses experimentos foram apresentados em um artigo (de Araújo e Ralha, 2011) que foi escolhido como *runner-up paper* na *Sixth International Conference on Forensic Computer Science* (ICoFCS 2011).

1.4 RESULTADOS ESPERADOS E LIMITAÇÕES

O principal resultado esperado deste trabalho é fomentar a pesquisa em exploração de coleções textuais no âmbito da Informática Forense, contribuindo para agilizar os exames periciais que envolvam buscas em coleções textuais.

Nesta pesquisa considera-se como premissa básica que arquivos textuais com conteúdo assemelhado são arquivos que possuem um determinado número de termos em comum, não sendo utilizado processamento semântico dos conceitos envolvidos. Assim, um arquivo que contém o termo "droga" como substantivo e outro que contém o mesmo termo como interjeição podem ser considerados semelhantes; já um arquivo que contém o termo "entorpecente" e outro que contém o termo "tóxico" não são considerados assemelhados. Essa limitação é decorrente do método de agrupamento utilizado, o qual não incorpora características semânticas de Processamento de Linguagem Natural.

O suporte à extração de texto de todos os possíveis formatos de arquivos textuais do universo da Informática Forense foge ao escopo desta pesquisa. Há uma vasta gama de tipos de arquivos comumente encontrados, a partir da qual serão elencados alguns exemplos:

- Documentos produzidos por processadores de texto e planilhas eletrônicas;
- Mensagens de correio eletrônico;

[3]Os dados reais dos casos, quais sejam, nomes de indivíduos e organizações, foram trocados por identificadores genéricos para não expor os investigados.

- Históricos de programas de mensagens instantâneas;
- Fragmentos de arquivos apagados, os quais ainda não foram sobrepostos e se encontram na área livre da mídia de armazenamento.

Dessa forma, este trabalho se limita a investigar o agrupamento dos *hits* obtidos dos seguintes tipos de arquivos: arquivos de usuário do pacote Microsoft Office (.doc, .docx, .xls, .xlsx, ppt, .pptx, .pps e .ppsx), arquivos de texto plano (.txt), arquivos de texto estruturado (.pdf e .rtf), e arquivos HTML e XML (.htm, .html e .xml).

Também está fora do escopo deste trabalho o processamento de arquivos apagados, cifrados, esteganografados, com extensão incorreta, comprimidos, danificados ou incompletos.

Por fim, o protótipo utiliza, na fase de pré-processamento, um algoritmo de detecção de idioma com a finalidade de somente processar arquivos que possam conter texto em língua portuguesa. Dessa forma, não são processados arquivos em cujo texto o algoritmo detectar com alta probabilidade que não há trechos em língua portuguesa. O teste de idioma é descrito em detalhes na Seção 4.1.1.

A literatura menciona as expressões "agrupamento de documentos" e "agrupamento de texto", mas não menciona "agrupamento de arquivos", e por isso o termo "documento" como sinônimo para arquivo será usado neste trabalho.

O restante desta dissertação está organizado da seguinte forma: os Capítulos 2 e 3 apresentam a área de Informática Forense e a fundamentação teórica do trabalho, respectivamente; o Capítulo 4 descreve a proposta e o método empregado neste projeto de pesquisa; o Capítulo 5 apresenta os experimentos e analisa seus resultados; e o Capítulo 6 traz as conclusões e sugere trabalhos futuros.

CAPÍTULO 2

INFORMÁTICA FORENSE

Este capítulo se inicia com um breve resumo da nomenclatura e dos aspectos legais relacionados à área de Informática Forense na Seção 2.1. A Seção 2.2 mostra uma visão geral da Informática Forense, para a qual definições importantes são apresentadas na Seção 2.3 e modelos de processo são discutidos na Seção 2.4. A Seção 2.5 discorre sobre exames periciais em Informática Forense, e a Seção 2.6 aborda ferramentas de Informática Forense, com ênfase em suas funcionalidades de buscas em coleções textuais e apresentação dos respectivos *hits*.

2.1 PERÍCIA CRIMINAL

O dicionário Michaelis oferece as seguintes definições para as palavras *perícia*[1], *perito*[2] e *vestígio*[3]:

- Perícia: *sf (lat peritia)* **1** Qualidade de perito. **2** Destreza, habilidade, proficiência. **3** *Dir* Exame de caráter técnico, por pessoa entendida, nomeada pelo juiz, de um fato, estado ou valor de um objeto litigioso. *P. grafoscópica:* a que se efetua por comparação de letras.

- Perito: *adj (lat peritu)* **1** Que tem perícia. **2** Experiente, hábil, prático, sabedor, versado. *sm* **1** Aquele que é prático ou sabedor em determinados assuntos. **2** Aquele

[1] `http://michaelis.uol.com.br/moderno/portugues/index.php?lingua=portugues-portugues&palavra=per%EDcia`, acesso em 25/08/2011.

[2] `http://michaelis.uol.com.br/moderno/portugues/index.php?lingua=portugues-portugues&palavra=perito`, acesso em 25/08/2011.

[3] `http://michaelis.uol.com.br/moderno/portugues/index.php?lingua=portugues-portugues&palavra=vest%EDgio`, acesso em 25/08/2011.

que é judicialmente nomeado para uma avaliação, exame ou vistoria. *P.-contador:* contador especializado ou judicialmente habilitado a resolver questões de contabilidade.

- Vestígio: *adj (lat vestigiu)* **1** Sinal deixado pela pisada ou passagem, tanto do homem como de qualquer outro animal; pegada, rasto. **2** Indício ou sinal de coisa que sucedeu, de pessoa que passou. **3** Ratos, resquícios, ruínas. *Seguir os vestígios de alguém:* fazer o que ele fez ou faz; imitá-lo.

O ordenamento jurídico brasileiro prevê perícias no interesse de processos judiciais cíveis (relacionados a conflitos de interesses) e criminais (relacionados à pretensão punitiva do Estado). O Código de Processo Penal[4] (CPP) rege estas últimas, e determina que será indispensável o exame de corpo de delito quando a infração penal deixar vestígios. É necessário interpretar o conceito de *corpo de delito* contido no CPP; embora o termo seja naturalmente associado aos exames médico-legais em pessoas vivas e mortas, também necessita abranger os vestígios relacionados a todas as modalidades de infrações penais. O exame será realizado por perito oficial, portador de diploma de curso superior, que deverá elaborar um laudo pericial onde descreverá minuciosamente o que examinar e responderá aos quesitos (perguntas) formulados pelo requisitante.

Todos os envolvidos na apuração de uma infração – juiz, membro do Ministério Público, delegado, advogado, ofendido, acusado – podem formular quesitos e requisitar esclarecimentos ao perito. O laudo é juntado aos autos (registros oficiais escritos) do processo. O juiz não fica adstrito ao laudo, podendo aceitá-lo ou rejeitá-lo, no todo ou em parte; e, no caso de inobservância de formalidades, ou no caso de omissões, obscuridades ou contradições, mandará suprir a formalidade, complementar ou esclarecer o laudo.

O mesmo dicionário Michaelis oferece a seguinte tradução para a expressão em inglês *forensic science*[5]:

- *Forensic science*: *n* métodos científicos utilizados pela polícia.

Criminalística é um termo muito utilizado em língua portuguesa para fazer referência à ciência forense e à perícia criminal.

No Departamento de Polícia Federal, as perícias criminais são realizadas pelos policiais ocupantes do cargo de Perito Criminal Federal, que é subdividido em áreas especializadas.

[4] `http://www.planalto.gov.br/ccivil_03/decreto-lei/Del3689Compilado.htm`, acesso em 24/08/2011.

[5] `http://michaelis.uol.com.br/moderno/ingles/index.php?lingua=ingles-portugues&palavra=forensic%20science`, acesso em 26/08/2011.

Os profissionais são selecionados em concurso público que tem como requisito de investidura o diploma de curso superior específico para a área de perícia na qual deverão atuar, e sua primeira atribuição específica é a execução de exames periciais[6]. Nos Institutos de Criminalística e órgãos congêneres estaduais, cuja esfera de atuação é circunscrita às Unidades da Federação onde funcionam, as perícias criminais são realizadas por peritos cujos cargos têm uma variedade de nomenclaturas: Peritos Criminalísticos, Peritos Criminais, ou simplesmente Peritos.

Na literatura em inglês é comum encontrar os termos *expert* (especialista) e *examiner* (examinador). A característica comum a todos esses profissionais, independente da nomenclatura, é que realizam exames técnico-científicos (perícias) no interesse de processos criminais.

2.2 VISÃO GERAL

Segundo Huebner et al. (2008), a partir da popularização dos microcomputadores pessoais que se deu nas décadas de 1970 e 1980, as agências de manutenção da lei se depararam com uma nova classe de crimes, os *crimes digitais*[7]. Os mesmos autores definem crimes digitais da seguinte forma (tradução livre):

> *Crimes digitais são entendidos de forma ampla como atos criminosos nos quais um computador é o objeto da transgressão, ou a ferramenta utilizada para o seu cometimento.*[8]

Segundo os mesmos autores, na década de 1990, as forças policiais de praticamente todos os países tecnologicamente desenvolvidos já tinham conhecimento e possuíam sistemas para investigar crimes digitais. Centros de pesquisa e universidades passaram a estudar a área recém-surgida e a indústria de *software* desenvolveu e ofereceu ferramentas para auxiliar nas investigações.

No período inicial da Informática Forense, os dispositivos de armazenamento tinham capacidades relativamente pequenas e continham quantidades de informação também peque-

[6] `https://conlegis.planejamento.gov.br/conlegis/Downloads/file?Prt.%20523-1989.pdf`, acesso em 27/08/2011.

[7] Tradução livre de *computer crime*, generalizada pelo tradutor para abranger outros dispositivos e tecnologias além de computadores.

[8] O termo "computador" deve ser interpretado de forma ampla para abranger outros tipo de dispositivos e tecnologias digitais.

nas (Huebner et al., 2008). A maioria das redes de computadores encontrava-se implantada em empresas e órgãos governamentais, sendo pouco comum encontrá-las em residências.

A evolução das tecnologias de informática e telecomunicações modificou esse cenário e introduziu novos desafios. As redes de computadores tornaram-se acessíveis e populares, e o uso da Internet cresceu muito. De acordo com o IBGE, a porcentagem dos domicílios que possuíam microcomputador com acesso à Internet no Brasil, que era de 8,5% em 2001, saltou para 27,7% em 2009[9].

Também foram lançados novos equipamentos computacionais, tais como *smartphones* (telefones celulares com aplicativos de produtividade e recursos avançados de acesso à Internet) e *tablets*. A introdução e a popularização dessas tecnologias trouxeram para o dia-a-dia facilidades como o comércio eletrônico e o *internet banking*, mas também abriram caminho para o aumento da incidência de crimes de computador. Segundo a Federação Brasileira de Bancos (FEBRABAN), os bancos brasileiros perderam 685 milhões de reais com fraudes eletrônicas no 1º semestre de 2011[10]. A Gartner, empresa norte-americana de pesquisa e consultoria, afirma que 3,6 milhões de adultos perderam 3,2 bilhões de dólares em ataques de *phishing*[11] no período de 12 meses que findou em 12 de agosto de 2007[12]. A tecnologia continua a evoluir e apresentar novos desafios, dentre os quais podem ser citados o aumento da complexidade dos sistemas computacionais, o crescimento da capacidade das mídias de armazenamento e a evolução ininterrupta das técnicas de criptografia.

A importância e os desafios da Informática Forense motivaram a realização da primeira edição do *Digital Forensic Research Workshop* (DFRWS), que aconteceu em 2001 em Utica, no estado norte-americano de New York, e reuniu mais de 50 participantes entre pesquisadores acadêmicos, especialistas e analistas em Informática Forense. O objetivo do evento era estimular a discussão entre acadêmicos e praticantes com experiência e interesse em Informática Forense. Palmer (2001) afirma, no relatório final da edição inicial do evento (tradução livre):

> *A meta maior foi estabelecer uma comunidade de pesquisa que iria aplicar o método científico para encontrar soluções focadas de curto prazo orientadas pelos*

[9] `http://www.ibge.gov.br/home/estatistica/populacao/trabalhoerendimento/pnad2009/graficosdinamicos/`, acesso em 24/08/2011.

[10] `http://economia.uol.com.br/ultimas-noticias/redacao/2011/08/19/bancos-perderam-r-685-milhoes-com-fraudes-eletronicas-no-1-semestre-segundo-febraban.jhtm`, acesso em 24/08/2011.

[11] Modalidade de ataque onde o atacante se faz passar por uma entidade confiável em uma comunicação eletrônica, a exemplo de uma mensagem de correio eletrônico, com o fim de obter informações confidenciais tais como nomes de usuário, senhas e números de cartões de crédito.

[12] `http://www.gartner.com/it/page.jsp?id=565125`, acesso em 24/08/2011.

requisitos dos praticantes e abordar necessidades de longo prazo, considerando os paradigmas correntes mas não se deixando limitar por eles.

O evento continua a ser realizado anualmente e passou a se chamar *Digital Forensics Research Conference*.

A Informática Forense é, portanto, uma área recente de pesquisa, em constante transformação e cada vez mais demandada. As definições pertinentes ao domínio da Informática Forense que são utilizadas nesta dissertação são apresentadas na Seção 2.3.

2.3 DEFINIÇÕES

Farmer e Venema (2005) oferecem a seguinte definição para Informática Forense (tradução livre):

(...) reunir e analisar dados de uma forma tão livre de distorção e viés quanto possível para reconstruir dados ou o que aconteceu no passado em um sistema.

Embora seja uma definição adequada do ponto de vista técnico, não menciona que as técnicas e os procedimentos adotados, assim como o resultado obtido, necessitam ser admissíveis como elementos probatórios em juízo. Uma outra definição que leva em conta esse aspecto é a de McKemmish (1999) (tradução livre):

(. . .) o processo de identificar, preservar e analisar evidências digitais de uma maneira legalmente aceitável.

Essa definição é mais rica e completa porque vai além dos aspectos técnicos. Não é suficiente identificar, preservar e analisar as evidências digitais; também é necessário cumprir os requisitos para que as descobertas obtidas a partir da análise das evidências no exame pericial possam ser aceitas no mundo jurídico, momento a partir do qual passam a ser consideradas provas. É oportuno citar a definição do Dicionário Michaelis para o termo *forense*[13]:

- Forense: *adj (lat forense)* **1** Que se refere ao foro judicial. **2** Relativo aos tribunais.

[13] `http://michaelis.uol.com.br/moderno/portugues/index.php?lingua=portugues-portugues&palavra=forense`, acesso em 25/08/2011.

A definição acima é suficiente para o objetivo desta dissertação, cujo foco é a atividade pericial regida pelo CPP. Ainda assim, é útil citar a definição de Palmer (2001) (tradução livre):

> *Ciência Forense Digital – O uso de métodos cientificamente elaborados e demonstrados para a preservação, coleta, validação, identificação, análise, interpretação, documentação e apresentação das evidências digitais obtidas a partir de fontes digitais para o propósito de facilitar ou promover a reconstrução de eventos que foram considerados criminosos, ou para ajudar a antecipar ações não-autorizadas para as quais se demonstrou que interrompem operações planejadas.*

Esta definição não menciona explicitamente o aspecto legal, porém é adequada para mostrar que a Informática Forense também pode ser aplicada a cenários nos quais as condutas criminosas ainda não foram praticadas. Além disso, busca alçar a Informática Forense ao *status* de ciência, indicando que o método científico deve ser aplicado a todas as suas atividades.

No cerne das definições de Informática Forense de McKemmish (1999) e Palmer (2001) está o conceito de *evidência digital*, para o qual esta pesquisa adota a definição de Whitcomb (2002) (tradução livre):

> *Informação de valor probatório armazenada ou transmitida de forma digital.*

Assim, a abrangência da Informática Forense vai além dos computadores e suas mídias de armazenamento, e também compreende todo tipo de dispositivo digital que armazene ou transmita informações, a exemplo de telefones celulares, *tablets* e aparelhos de *videogame*. Este trabalho também utilizará o termo *vestígio* como equivalente para *evidência digital*, porque o primeiro é o utilizado pelo CPP.

A investigação dos crimes digitais se utiliza de métodos consagrados de investigação policial e a eles mescla elementos da Ciência da Computação e áreas correlatas. Esse novo ramo das Ciências Forenses recebeu várias nomenclaturas diferentes em inglês, a exemplo de *computer forensics*, *forensic computer science* e *digital investigation*. Este trabalho utiliza o termo *Informática Forense* em português para descrever a área de estudo. O termo *crimes digitais* descrito na Seção 2.2 será utilizado para descrever as condutas criminosas.

Huebner et al. (2008) definem da seguinte forma o objetivo da Informática Forense (tradução livre):

A Informática Forense tem por objetivo solucionar e documentar crimes digitais, e permitir que sejam objeto de processos judiciais.

Os mesmos autores também propuseram a seguinte classificação para os crimes digitais:

- Crimes digitais centrados no computador: atividade criminal que tem por alvo computadores, redes de computadores, mídias de armazenamento ou outros sistemas computacionais (por exemplo, invadir um sítio comercial de Internet e modificar seu conteúdo). Isto pode ser visto como novas ferramentas permitindo uma nova classe de condutas criminosas;

- Crimes auxiliados por computadores: uso de computadores como ferramentas para auxiliar em condutas criminosas nas quais o uso de computadores não é estritamente necessário (por exemplo, distribuição de material de pornografia infantil). Isto pode ser visto como novas formas de praticar condutas criminosas que já existiam antes do uso e da popularização dos computadores;

- Crimes de computador incidentais: conduta criminosa onde o uso do computador é incidental (por exemplo, contabilidade de transações de tráfico de drogas feita por meio de planilhas eletrônicas). Isto pode ser visto como novas ferramentas tomando o lugar de velhas ferramentas (por exemplo, a substituição de cadernos de anotações por planilhas eletrônicas).

Os autores ressaltam que a classificação objetiva unicamente auxiliar na compreensão da área que descreve, e que existem cenários onde atividades criminosas podem abranger mais de uma das classes ou não se enquadrar claramente em nenhuma das classes.

Diversos modelos de processo foram propostos para formalizar as atividades da Informática Forense. A Seção 2.4 apresenta a motivação e os princípios que nortearam alguns dos vários modelos propostos, e descreve alguns desses modelos.

2.4 MODELOS DE PROCESSO

Na literatura em inglês, o termo *framework* é utilizado para descrever modelos de processo que foram propostos e aplicados ao domínio da Informática Forense; em particular, é utilizado para descrever o modelo de processo que será estudado de forma mais aprofundada na Seção 2.4.3. Por esses motivos, o termo *framework* como sinônimo para modelo de processo será usado deste ponto em diante.

Primeiramente é descrita a motivação para a formalização de *frameworks* de Informática Forense. Em seguida, os princípios que devem ser seguidos pelos *frameworks* são discutidos. Três *frameworks* são então descritos de forma superficial, e um outro é estudado em maior detalhe e contextualizado na pesquisa desta dissertação.

Neste ponto é necessário definir *framework*. A definição de Johnson e Foote (1988), embora seja voltada para outro domínio, o da programação orientada a objetos, é adequada para a Informática Forense (tradução livre):

> *Um* framework *é um conjunto de classes que incorpora um modelo abstrato para as soluções de uma família de problemas relacionados.*

No contexto dos *frameworks* de Informática Forense, as *classes* não têm a mesma conotação que na programação orientada a objetos; na verdade, representam as fases da investigação. O *framework* é caracterizado pela formalização do que é feito em cada uma das fases e as interações entre elas. O modelo abstrato resultante pode ser aplicado a uma variedade de cenários de investigações de Informática Forense.

2.4.1 MOTIVAÇÃO

Conforme citado na Seção 2.2, a Informática Forense emprega métodos de investigação policial. Palmer (2001) diz que as atividades do especialista em Informática Forense são de natureza investigativa. Huebner et al. (2008) apontam que todo investigador deve estar atento ao chamado *Princípio da Troca de Locard* (tradução livre):

> *Todo aquele e tudo quanto adentrar uma cena de crime leva consigo algo da cena, ou deixa algo de si para trás ao partir.*

Segundo Hoelz (2009), no contexto da Informática Forense, o princípo tem as seguintes implicações às quais o especialista deve estar atento:

- Toda atividade em um sistema computacional provavelmente produzirá alterações, no mínimo, em sua memória principal, e possivelmente também em seu sistema de arquivos. Essas duas áreas de armazenamento são, portanto, de interesse para o exame pericial;

- Como os materiais digitais, a exemplo de arquivos lógicos e registros em bases de dados, são mais facilmente modificáveis e forjáveis que os objetos físicos, o especialista deve preservar a integridade dos vestígios mantendo-os inalterados, trabalhando

preferencialmente sobre uma cópia integral e exata dos dados originais. As medidas adotadas para manter a integridade dos vestígios devem ser documentadas com clareza.

Conforme mencionado na Seção 2.3, o perito deve proceder de tal forma que suas descobertas sejam admissíveis em juízo. Embora o CPP não determine de forma explícita que a perícia deve ser realizada com rigor científico, deixar de fazê-lo dessa forma embute vários riscos:

- Um exame pericial com conclusões incorretas pode condenar um inocente ou libertar um culpado;
- Um exame pericial com conclusões corretas, porém mal fundamentado, pode ser contestado por uma das partes e, em consequência, ser descartado pelo juiz.

O perito deve realizar seu exame pericial seguindo o método científico para que as conclusões apresentadas no laudo sejam corretas e bem fundamentadas, e possam ser plenamente aproveitadas pelo julgador para formar sua convicção.

2.4.2 PRINCÍPIOS

Faz-se necessário neste momento definir *cadeia de custódia*, um conceito importante não apenas para as investigações policiais, mas para todo o processo penal. Segundo Hoelz (2009), a cadeia de custódia é:

> *(...) o registro de todas as pessoas que manusearam ou locais que mantiveram a custódia de uma evidência durante toda a sua existência, desde a coleta na cena do crime até o seu uso final no processo judicial.*

Beebe e Clark (2005) elencam uma série de princípios que afirmam que devem ser aplicados a todas as atividades da Informática Forense e aprofundam a discussão de dois princípios em particular (tradução livre):

1. Preservação dos vestígios – Os objetivos primários deste princípio são: (*i*) maximizar a disponibilidade e a qualidade dos vestígios; e (*ii*) manter a integridade dos vestígios durante todo o processo;
2. Documentação – identificar os vestígios; registrar as circunstâncias nas quais foram coletados; subsidiar a manutenção da cadeia de custódia; descrever as ferramentas

e técnicas utilizadas; relatar as conclusões de forma clara e objetiva; arquivar e salvaguardar toda a documentação produzida durante o processo.

McKemmish (1999) descreve o que chama de "regras" da Informática Forense (tradução livre):

1. Manuseio mínimo do original – Considerada pelo autor como a mais importante, esta regra determina que a aplicação de procedimentos de Informática Forense ao examinar os dados originais deve ser reduzida ao mínimo absoluto, e que os exames devem ser conduzidos, sempre que possível, em uma cópia integral e exata dos dados originais;

2. Documentar todas as modificações – Esta regra estipula que, sempre que o perito necessitar realizar alguma modificação em um vestígio, a natureza, a extensão e a razão dessa modificação devem ser documentadas adequadamente, para que posteriormente as descobertas obtidas a partir do vestígio sejam admissíveis em juízo;

3. Cumprir as regras que regem as provas judiciais – A aplicação das ferramentas e técnicas da Informática Forense não deve diminuir o valor probatório das descobertas;

4. O especialista não deve exceder seu próprio conhecimento – O especialista somente deve executar exames periciais para os quais detenha a qualificação necessária, sob risco de não ser capaz de documentar adequadamente suas ações, não conseguir relatar de forma convincente suas descobertas, e, o mais grave, contaminar os vestígios.

Os princípios e regras descritos acima podem ser encontrados, sob várias formas, em diversos *frameworks* que foram desenvolvidos e propostos com o fim de formalizar as técnicas e procedimentos da Informática Forense à luz da boa prática investigativa, do cumprimento das normas jurídicas e da obediência ao método científico. Alguns desses *frameworks* são descritos na Seção 2.4.3.

2.4.3 FRAMEWORKS PROPOSTOS

Huebner et al. (2008) discutem três *frameworks* que foram propostos para investigações de Informática Forense, resumidos na Tabela 2.1. Os mesmos autores identificaram características comuns às definições de Informática Forense de Farmer e Venema (2005) e McKemmish (1999), bem como aos *frameworks* discutidos:

Tabela 2.1: *Frameworks* descritos por Huebner et al. (2008)

Farmer e Venema (2005)	McKemmish (1999)	Prosise e Mandia (2003)
1. Assegurar e isolar o local do crime 2. Registrar a cena do crime 3. Conduzir uma busca sistemática por evidências 4. Coletar e embalar as evidências 5. Manter a cadeia de custódia	1. Identificação 2. Preservação 3. Análise 4. Apresentação	1. Preparação pré-incidente 2. Detecção do incidente 3. Resposta inicial 4. Formular a estratégia de resposta 5. Investigar o incidente: coleta de dados seguida por análise de dados 6. Relatório 7. Resolução (medidas de segurança, lições aprendidas, soluções de longo prazo)

1. Todas as definições e *frameworks* se baseiam em abordagens descritas na literatura de investigação de crimes "convencionais", que por sua vez são baseadas no *Princípio da Troca de Locard*, citado na Seção 2.4.1. Os autores argumentam que essa conformidade é necessária para que as conclusões obtidas sejam admissíveis em juízo.

   É válido citar que o *framework* de Farmer e Venema (2005) contempla apenas os passos iniciais da investigação. Os autores não criaram um *framework* original, apenas propuseram aplicar às investigações de Informática Forense, sem alterações, um método básico apresentado em uma outra obra que trata de Ciências Forenses em geral.

2. Os *frameworks* descrevem formalmente passos detalhados, inclusive com fluxogramas e procedimentos adicionais, o que resulta em listas extensas e sequências de passos que devem ser seguidas. Uma das justificativas é a intenção de tornar o processo menos propenso a erros. A outra é o desejo de demonstrar que regras idôneas foram estritamente seguidas, e assim os resultados são válidos e admissíveis em juízo;

3. As definições são amplas e não necessariamente amarradas a cenários de crimes digitais. Se os termos de informática fossem removidos, as definições continuariam válidas.

Vários outros *frameworks* foram propostos na literatura, e um deles será estudado de forma mais aprofundada neste trabalho.

O FRAMEWORK DE BEEBE E CLARK

Beebe e Clark (2005) propuseram um *framework* hierárquico onde buscaram reunir os pontos positivos de outros *frameworks*, bem como preencher o que consideraram lacunas. O *framework* é formado por um conjunto de seis fases de primeiro nível, que por sua vez comportam um número variável de subfases de segundo nível. As subfases de segundo nível são determinadas pelos objetivos buscados na investigação. Os autores argumentam que, de acordo com sua experiência, um *framework* para investigações de Informática Forense deve ser baseado em objetivos, não em tarefas, porque cada caso concreto exige abordagens específicas, e oferecem um exemplo de matriz objetivo-tarefa para o qual um trecho é apresentado na Figura 2.1.

Tarefas de análise de dados	Objetivos de análise de dados							
	Redução de dados	Avaliação do nível de habilidade	Recuperação de arquivos apagados	Detecção e recuperação de dados apagados	Cronologia de atividades	Recuperação de dados ASCII	Recuperação de arquivos por tipo	Recuperação de *emails*
Análise de assinatura	✓	✓		✓				
Análise de *hashes*	✓	✓		✓				
Cronologia de atividade de arquivos	✓				✓			
Análise de chaves de registro					✓			
Identificação de fluxos de dados		✓		✓				
Detecção de esteganografia		✓		✓				
Identificação de utilitários de *wipe*		✓						
R&A* de arquivos apagados			✓			✓		
R&A do histórico de arquivos apagados			✓		✓			
Recuperação de partições		✓	✓					
Pesquisa por palavras-chave e análise			✓	✓		✓		✓
Recuperação de arquivos por assinatura			✓	✓			✓	✓

*R&A = Recuperação & Análise

Figura 2.1: Trecho da matriz de objetivos e tarefas de Beebe e Clark (2005)

As fases de primeiro nível do *framework* proposto são mostradas na Figura 2.2, e descritas a seguir em tradução livre. Como o *framework* é genérico e se presta a uma ampla variedade de cenários, muitos dos quais fogem ao escopo desta dissertação, as descrições

serão estritamente focadas no ponto de vista da atuação da polícia e do perito criminal à luz do CPP.

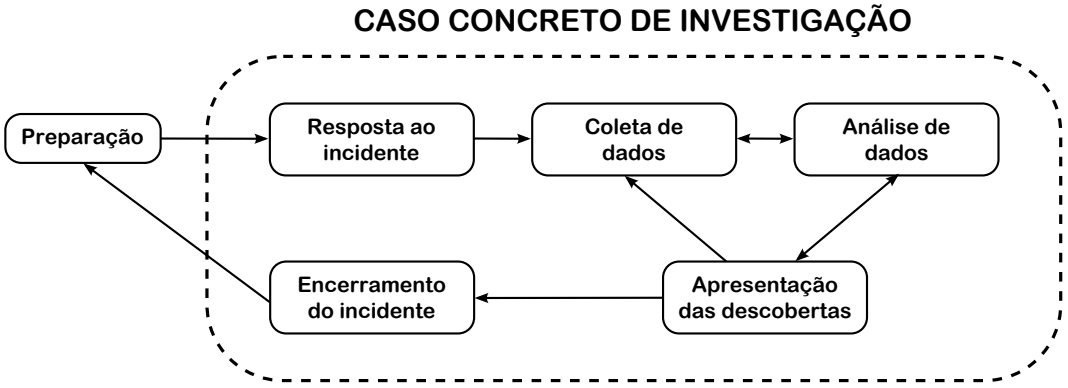


Figura 2.2: Fases de primeiro nível do *framework* proposto por Beebe e Clark (2005)

1. Preparação – Esta fase não faz parte da investigação propriamente dita. O termo "preparação" não se refere às atividades que antecedem imediatamente uma investigação, e sim às atividades que os envolvidos devem realizar para estar preparados para uma investigação. Nesta fase os peritos desenvolvem habilidades técnicas, participam de treinamentos e aperfeiçoam procedimentos de manuseio de vestígios. Os autores mencionam que as organizações que são vítimas em potencial de crimes digitais podem melhorar a qualidade e a disponibilidade das evidências digitais através de uma série de atividades que caracterizam a chamada *forensic readiness* (*prontidão para perícia*).

2. Resposta ao incidente – Esta é a primeira fase da investigação propriamente dita em que o perito poderá atuar, embora em muitos casos isso não aconteça. Isso se deve ao fato de que as equipes policiais que cumprem mandados judiciais de busca e apreensão não necessariamente incluem peritos. Caso o perito participe desta fase, deverá fazer o levantamento do local de crime e avaliar as possíveis fontes de vestígios, bem como adotar medidas para minimizar o risco de perda de vestígios.

3. Coleta de dados – No decorrer do cumprimento de um mandado judicial de busca e apreensão, esta fase começa imediatamente após o término da anterior, sem intervalo temporal; por isso, também nela é possível que a equipe não conte com um perito. É nesta fase que se deve proceder à coleta de vestígios, tais como discos rígidos e

imagens da memória principal de computadores, e mídias removíveis como dispositivos de memória *flash* (popularmente conhecidos como *pen drives*) e discos óticos. O responsável pela coleta deve atentar para a integridade dos vestígios, que devem ser identificados, embalados e lacrados para garantir a cadeia de custódia.

Caso o perito não participe desta fase, é possível que a equipe que executou o mandado de busca e apreensão, por ser formada por policiais que não são especialistas em Informática Forense, tenha coletado todos os computadores e mídias presentes no local e/ou tenha deixado de coletar outros vestígios que não estavam evidentes. Segundo Hoelz (2009), a consequência disso é que muitas vezes, na fase seguinte, o perito terá que examinar um grande volume de material que tem pouca relevância.

4. Análise de dados – Esta é a fase mais complexa e demorada de todo o processo, porque é nela que ocorre o exame pericial. No Brasil, esta fase deve ser obrigatoriamente desempenhada por um perito, conforme descrito na Seção 2.1. Esta fase contempla a análise confirmatória, que se propõe a confirmar ou refutar as alegações de atividade suspeita, e/ou a reconstrução de eventos, para responder às perguntas do tipo "Quem fez?", "O que fez?", "Onde fez?", "Quando fez?", "Como fez?" e "Por que fez?". Segue uma relação exemplificativa e não-exaustiva das atividades desta fase:

   4.1. Transformar (reduzir) as volumosas massas de dados coletadas durante a fase de *Coleta de dados*, de forma que possam ser melhor analisadas;

   4.2. Empregar técnicas de extração de dados, a exemplo de buscas por palavras-chave textuais, obtenção dos dados contidos no espaço não-alocado e descoberta de dados ocultos;

   4.3. Examinar, analisar e reconstruir eventos a partir dos dados para responder questões de importância crítica para a investigação.

   Os autores dividem esta fase de primeiro nível em três subfases de segundo nível, apresentadas na Figura 2.3.

5. Apresentação das descobertas – O propósito desta fase é comunicar as descobertas relevantes obtidas na fase de *Análise de dados* aos interessados, particularmente ao requisitante dos exames. A apresentação deve ser objetiva e detalhada. O perito deve ter em mente que o requisitante frequentemente não é um especialista em Informática Forense. Os aspectos técnicos do exame, portanto, devem ser explanados de forma clara e didática.

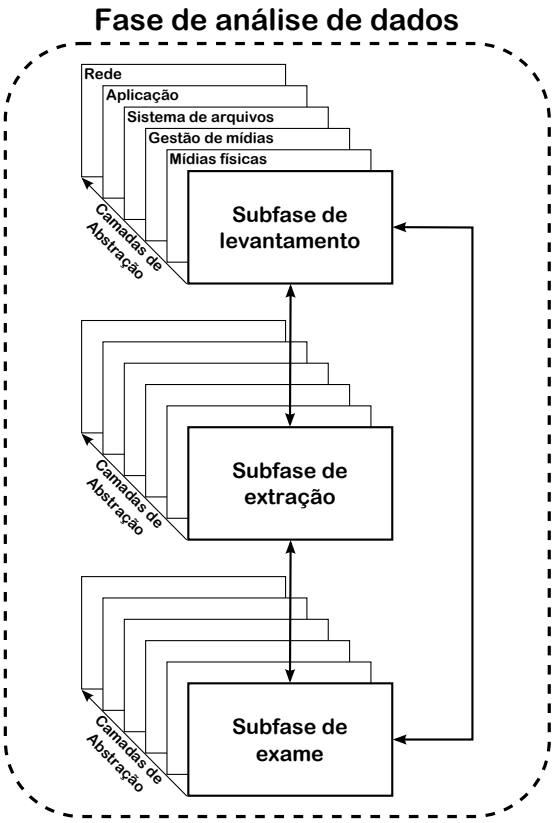


Figura 2.3: Subfases da fase de *Análise de dados*, adaptado de Beebe e Clark (2005)

6. Encerramento do incidente – Nesta fase, os autores argumentam que não se deve apenas adotar os passos mais óbvios, tais como a devolução dos materiais coletados e o arquivamento da documentação; também devem ser tomadas medidas para preservar o conhecimento obtido, para que possa ser reaproveitado em investigações futuras.

A Seção 2.5 que segue apresenta uma visão mais aprofundada da fase de análise, relacionada ao exame pericial propriamente dito.

2.5 EXAMES PERICIAIS

Conforme mencionado na Seção 2.1, os peritos examinam o *corpo de delito* (vestígios) e elaboram o laudo pericial, onde descrevem minuciosamente o que examinaram, com o fim de elucidar a materialidade do fato (o que aconteceu), a autoria (quem o praticou) e a dinâmica (como o fato ocorreu).

Segundo Hoelz (2009), há três tipos bem definidos de exames periciais em Informática Forense:

1. Exames em mídias de armazenamento como discos rígidos, mídias óticas como CD e DVD, discos flexíveis, cartões de memória e dispositivos de memória *flash*, dentre outras mídias;

2. Exames em programas e sistemas como exames de *softwares* maliciosos como cavalos-de-troia, vírus e *keyloggers* e em sistemas de bases de dados;

3. Exames em redes de computadores como, por exemplo, exames de fluxos de dados capturados, mensagens de correio eletrônico, conteúdo de sítios na Internet e *logs* (registros históricos) de dispositivos de rede como roteadores, *firewalls* e sistemas de detecção de intrusão.

Conforme o perito conduz o exame e localiza informações de interesse, cria *bookmarks* (conjuntos temáticos) onde registra e organiza suas descobertas, à luz do princípio *Documentação*. Isto se aplica a todos os tipos de exames em Informática Forense. Todas as ferramentas descritas na Seção 2.6 apresentam a funcionalidade de criação e gerenciamento de *bookmarks*.

Esta dissertação dá ênfase aos exames em mídias de armazenamento, os quais são detalhados na Seção 2.5.1.

2.5.1 EXAMES EM MÍDIAS DE ARMAZENAMENTO

Os exames em mídias de armazenamento têm por objetivo localizar informações de interesse da investigação nas mídias que foram arrecadadas na fase de coleta de dados. É um tipo de exame importante porque é nas mídias de armazenamento que os sistema computacionais guardam informações, tais como documentos de texto e planilhas eletrônicas, registros de bases de dados e históricos de navegação de Internet. A análise dessas informações frequentemente revela vestígios importantes.

O manuseio de mídias de armazenamento no âmbito da Informática Forense requer cuidados específicos. Em observância ao princípio *Preservação dos vestígios* de Beebe e Clark (2005) e à regra *Manuseio mínimo do original* de McKemmish (1999) discutidos na Seção 2.4.2, o conteúdo integral das mídias de armazenamento deve ser, sempre que possível, copiado integralmente para uma outra mídia sob a forma de um *arquivo de ima-*

gem[14]; assim, evita-se a modificação e/ou destruição inadvertida dos dados contidos na mídia original. Os exames periciais devem ser conduzidos, sempre que possível, sobre os arquivos de imagem.

Códigos de integridade criptográficos (*hashes*) calculados durante o processo de cópia indicam, com altíssima probabilidade, que os conteúdos da mídia original e de seu arquivo de imagem são idênticos. O código de integridade de um arquivo de imagem cujo conteúdo tenha sido modificado de um único *bit* será diferente, com altíssima probabilidade, do código calculado a partir do conteúdo original da mídia de armazenamento da qual foi obtido. Após a conclusão da cópia do conteúdo de uma mídia de armazenamento para um arquivo de imagem, este não deve mais ser modificado de nenhuma forma, sob pena de comprometer a cadeia de custódia. Dessa forma, pode-se afirmar que os arquivos de imagem têm natureza estática.

Os exames periciais em arquivos de imagem são chamados por Carvey (2009) de exames *postmortem* de Informática Forense[15]. As mídias apreendidas durante o cumprimento de mandados judiciais de busca são submetidas a esse tipo de exame com o fim de obter informações de interesse da investigação.

Hoelz (2009) cita uma série de procedimentos que podem ser realizados no decorrer dos exames em mídias de armazenamento. Segue uma lista não-exaustiva desses procedimentos:

- Análise da cronologia do sistema de arquivos e de outros registros temporais;
- Exame de chaves de registro e configurações do sistema operacional;
- Identificação de fluxos de dados ocultos e esteganografia;
- Recuperação de arquivos apagados e fragmentos de dados;
- Pesquisas por arquivos contendo palavras-chave;
- Extração de metadados em documentos e imagens;
- Análise do histórico e *cache* dos navegadores de Internet.

As ferramentas comerciais FTK (AccessData, 2011) e *EnCase* (Guidance, 2011) permitem examinar, além de arquivos de imagem de mídias de armazenamento, vários outros tipos de objetos, a exemplo de imagens de memória principal e arquivos individuais. O

[14]Arquivo binário que armazena uma cópia fiel do conteúdo de uma mídia de armazenamento.
[15]Tradução livre de *postmortem computer forensic analysis*.

termo *evidência* (*evidence*) é utilizado de forma ampla, com base na definição apresentada na Seção 2.3, para descrever o objeto examinado.

Esta dissertação dá ênfase nos exames em mídias de armazenamento e na tarefa de pesquisa por arquivos contendo palavras-chave, descrita a seguir.

PESQUISAS POR ARQUIVOS CONTENDO PALAVRAS-CHAVE

Segundo Hoelz (2009), a pesquisa por palavras-chave é um dos métodos mais utilizados para identificar rapidamente arquivos de interesse, sendo que dois métodos principais são utilizados para realizar buscas textuais:

- Na busca exaustiva ou ao vivo (*live search*), uma pesquisa é feita em cada um dos setores ou arquivos da evidência examinada, do início ao fim da área de dados, para cada termo de busca desejado (Forte, 2004);
- Na busca indexada, um índice é criado com todas os termos encontrados na evidência e suas respectivas posições de ocorrência. A pesquisa então é feita de forma rápida por meio da consulta ao índice (Johansson, 2003).

Ainda segundo Hoelz (2009), as pesquisas por palavras-chave apresentam algumas limitações, a saber:

- A busca não é feita com base no contexto, apenas com base na palavra-chave em si;
- A criação do índice para a pesquisa indexada pode ser muito demorada, dependendo do tamanho da mídia examinada, e não pode ser feita de forma distribuída;
- A pesquisa sem o auxílio de um índice é lenta em dispositivos de grande capacidade.

A última limitação citada, a de pesquisas em dispositivos de grande capacidade, tornou-se frequente. Em setembro de 2011, era possível adquirir um disco rígido interno de 3 *terabytes*, um dispositivo de memória *flash* de 128 *gigabytes* e um computador tipo *netbook* com disco rígido de 500 *gigabytes* por cerca de US$ 685 nos EUA[16], conforme mostrado na Figura 2.4.

O crescimento da capacidade das mídias de armazenamento permite que seja armazenada uma quantidade cada vez maior de informações, o que inclui documentos textuais. Segundo Beebe e Clark (2007), as buscas em coleções textuais forenses frequentemente

[16]Valor obtido no sítio `http://www.newegg.com`, acesso em 02/09/2011.

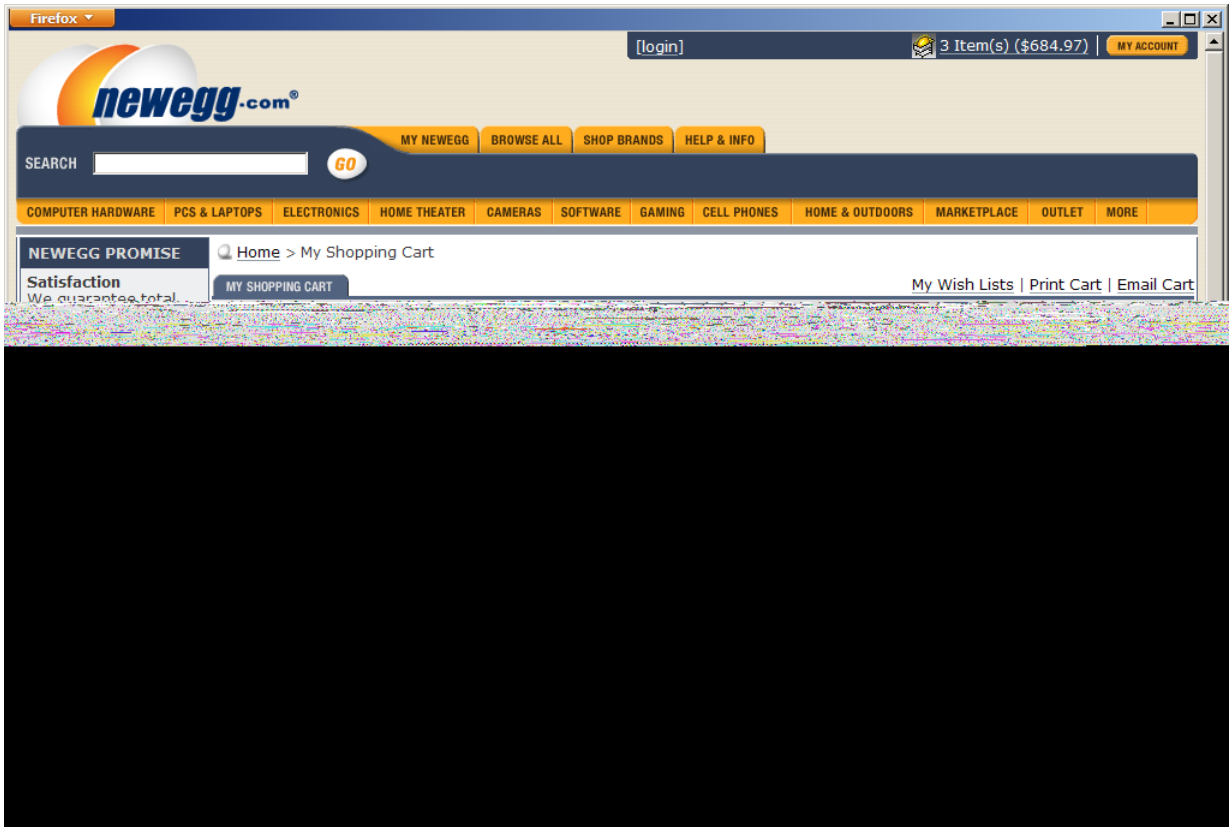


Figura 2.4: 3,628 *terabytes* de armazenamento por cerca de US$ 685.

trazem de centenas a milhares de *hits*. Os *hits* não são classificados nem ordenados com base em seu conteúdo. Quando o perito submete uma busca à aplicação e, ao examinar a lista de *hits*, localiza um documento de interesse, não tem alternativa senão examinar todos os demais *hits* para que possa localizar outros documentos de conteúdo semelhante que porventura existam, conforme exposto na Seção 1.1.

Uma pesquisa realizada entre especialistas em Informática Forense em 2010 revelou que 60,7% dos participantes escolheram "maior eficiência na busca por dados" como uma das três mudanças mais importantes nas ferramentas periciais que lhes permitiriam concluir os exames em menos tempo (Kuncik, 2010). Esta pesquisa se propõe a oferecer uma técnica que permita agilizar os exames periciais que envolvam buscas em coleções textuais.

2.5.2 ANÁLISE DE MEMÓRIA VOLÁTIL E DE SISTEMAS EM OPERAÇÃO

Uma abordagem que vem ganhando impulso, para a qual existem diversas publicações na literatura e também ferramentas especializadas, é a denominada de análise de memória volátil e de sistemas em operação (*live analysis*) (Hoelz, 2009). Lessing e von Solms (2008) argumentam que essa abordagem possui diversas aplicações importantes do ponto de vista da Informática Forense, tais como:

- Capturar arquivos de imagem de mídias de armazenamento que estejam logicamente acessíveis porém fisicamente cifradas;
- Capturar *dumps* (despejos) binários de memória principal, que podem conter informações sobre processos maliciosos e senhas, entre outras informações;
- Capturar o tráfego de rede do sistema computacional questionado.

Lessing e von Solms (2008) também citam uma desvantagem, que é a possibilidade real de modificar os dados armazenados nos sistemas de arquivos e na memória principal, algo que parece incompatível com o princípio *Preservação dos vestígios* e a regra *Manuseio mínimo do original* citados na Seção 2.4.2. A questão foi abordada por Casey (2007) e Lessing e von Solms (2008), que fazem uma analogia entre os vestígios biológicos e as evidências digitais. Os autores argumentam que o perito em Genética Forense, ao submeter vestígios biológicos a processos químicos com o fim de extrair o material genético que porventura esteja lá contido para realizar um exame de DNA, altera ou mesmo destrói os vestígios que servem de suporte para o material genético, sem no entanto destruir nem contaminar o material genético e assim permitindo que o exame seja realizado sem problemas. As provas periciais baseadas em DNA são aceitas regularmente em juízo, porque são obtidas através de procedimentos baseados em métodos científicos sólidos.

Lessing e von Solms (2008) afirmam que, se no campo da Informática Forense forem adotadas medidas para que as alterações sejam as mínimas possíveis e se forem estritamente observados o princípio *Documentação* e a regra *Documentar todas as modificações* de forma que o *significado* do conteúdo da evidência não seja descaracterizado, pequenas alterações são toleráveis. Ou seja, se as modificações forem pequenas, adequadamente documentadas e não levarem o perito a conclusões equivocadas, podem ser aplicadas a cenários que as exijam.

Esta proposta é razoável, porque sem o emprego de técnicas de análise de memória volátil e de sistemas em operação, seria inviável, por exemplo, realizar um exame pericial em uma mídia de armazenamento cujo conteúdo esteja cifrado com um algoritmo de criptografia forte; a decifração do conteúdo poderia tomar mais tempo, que, literalmente, a idade estimada do universo. Se o dispositivo computacional onde está instalada a mídia com conteúdo cifrado estiver ligado e em operação e o conteúdo da mídia estiver acessível durante o cumprimento do mandado judicial de busca e apreensão, essa pode ser a única oportunidade que o perito terá para duplicar o conteúdo da mídia para um arquivo de imagem, a partir do qual o exame pericial *postmortem* poderá ser realizado sem que seja necessário utilizar técnicas de decifração; essa oportunidade deve ser aproveitada, mesmo que isso exija que o perito manipule o dispositivo e execute programas armazenados em mídias removíveis para obter os dados da mídia que será apreendida.

Em estrita observância ao princípio *Documentação* e à regra *Documentar todas as modificações*, deverão ser claramente documentados no laudo pericial a motivação e os aspectos técnicos de um procedimento de obtenção de evidências digitais que implique alteração, para que as descobertas obtidas sejam admissíveis em juízo. Segue o argumento de Casey (2007) em tradução livre:

> *Estabelecer um padrão absoluto que decreta "preservar tudo mas não alterar nada" não apenas é inconsistente com outras disciplinas forenses como também é perigoso em um contexto jurídico. Agir de acordo com um tal padrão pode ser impossível em algumas circunstâncias e, portanto, estipular esse padrão como a "melhor prática" somente expõe as evidências digitais a críticas que não têm relação com as questões sob investigação. Quando se questiona se o item de evidência digital sofreu alteração de* qualquer *espécie, no lugar de questionar se essa alteração compromete a confiabilidade ou a autenticidade dos resultados, isso tira o foco que deveria estar nos aspectos essenciais da evidência. Assim, é importante que se faça distinção entre, de um lado, práticas forenses idôneas e, do outro, paradigmas impraticáveis.*

Há diversas ferramentas à disposição do perito para realizar exames em mídias de armazenamento. Três ferramentas muito utilizadas são descritas na Seção 2.6.

2.6 FERRAMENTAS

Esta seção aborda as ferramentas comerciais FTK (AccessData, 2011) e *EnCase* (Guidance, 2011). Uma pesquisa realizada entre examinadores da área de Informática Forense publicada por Kuncik (2010) revelou que 82,1% dos participantes selecionaram o *EnCase* como uma das quatro ferramentas que mais utilizavam para realizar exames periciais; o FTK foi selecionado por 71,4% dos participantes. Ambas as ferramentas são utilizadas pelo Departamento de Polícia Federal. Também é abordada a ferramenta livre formada pela combinação TSK/*Autopsy* (Carrier, 2011), cujos *downloads* somados totalizavam 316.037 em 17 de agosto de 2011[17]. Relacionadas a estas ferramentas, serão discutidas as funcionalidades de buscas em coleções textuais e apresentação dos respectivos *hits*.

2.6.1 FTK

O FTK (AccessData, 2011) é uma ferramenta forense desenvolvida pela empresa norte-americana AccessData. Permite buscas textuais tanto exaustivas quanto indexadas. Os resultados das buscas são classificados por tipo de arquivo em categorias pré-determinadas. Para cada categoria são exibidos os arquivos que contêm o termo de busca, ordenados pela quantidade de ocorrências de forma decrescente. A ferramenta não agrupa os arquivos por similaridade. A forma de funcionamento descrita pode ser visualizada nas Figuras 2.5 e 2.6. Note-se que as categorias pré-determinadas incluem diferentes tipos de arquivos, tais como documentos, planilhas, gráficos e executáveis, sem contudo apresentar uma visão baseada na similaridade dos conteúdos dos arquivos.

[17] `http://sourceforge.net/projects/sleuthkit/files/stats/timeline?dates=2002-06-12+to+2011-08-17`, acesso em 18/08/2011.

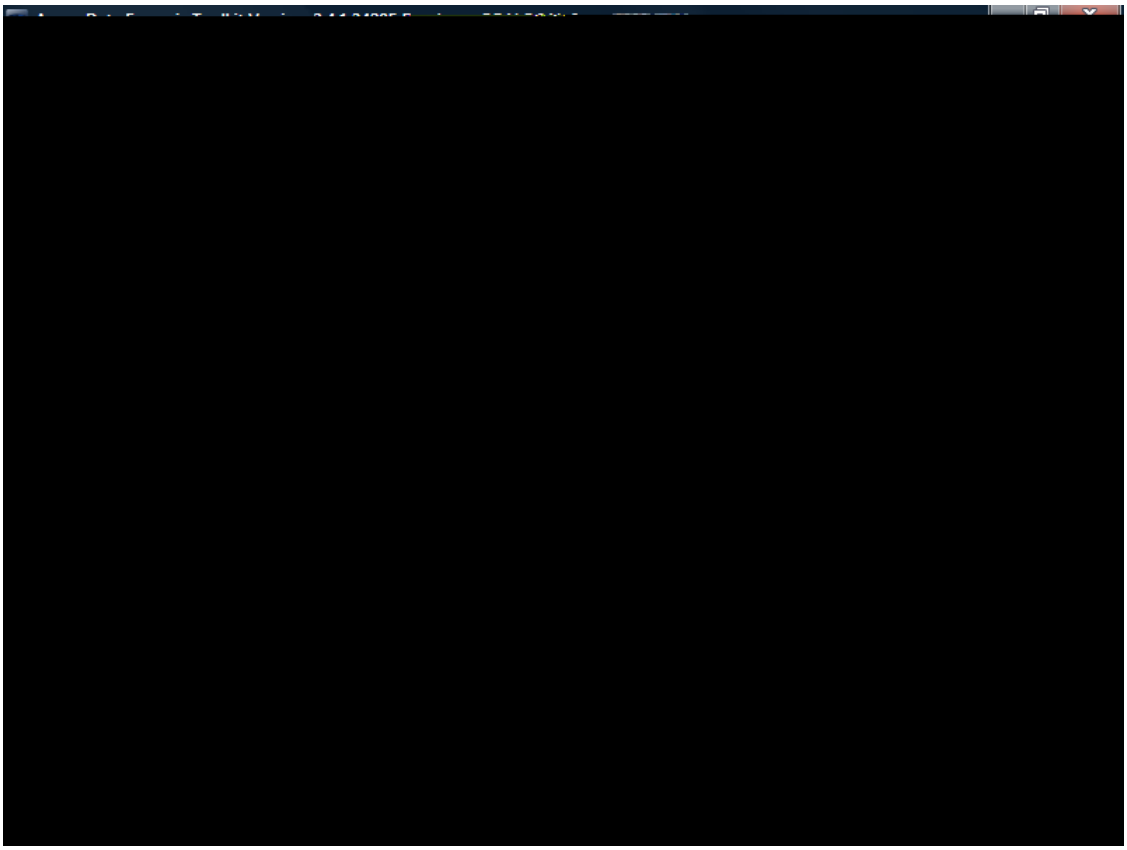


Figura 2.5: Tela de resultado de busca textual no FTK com categorias pré-determinadas.

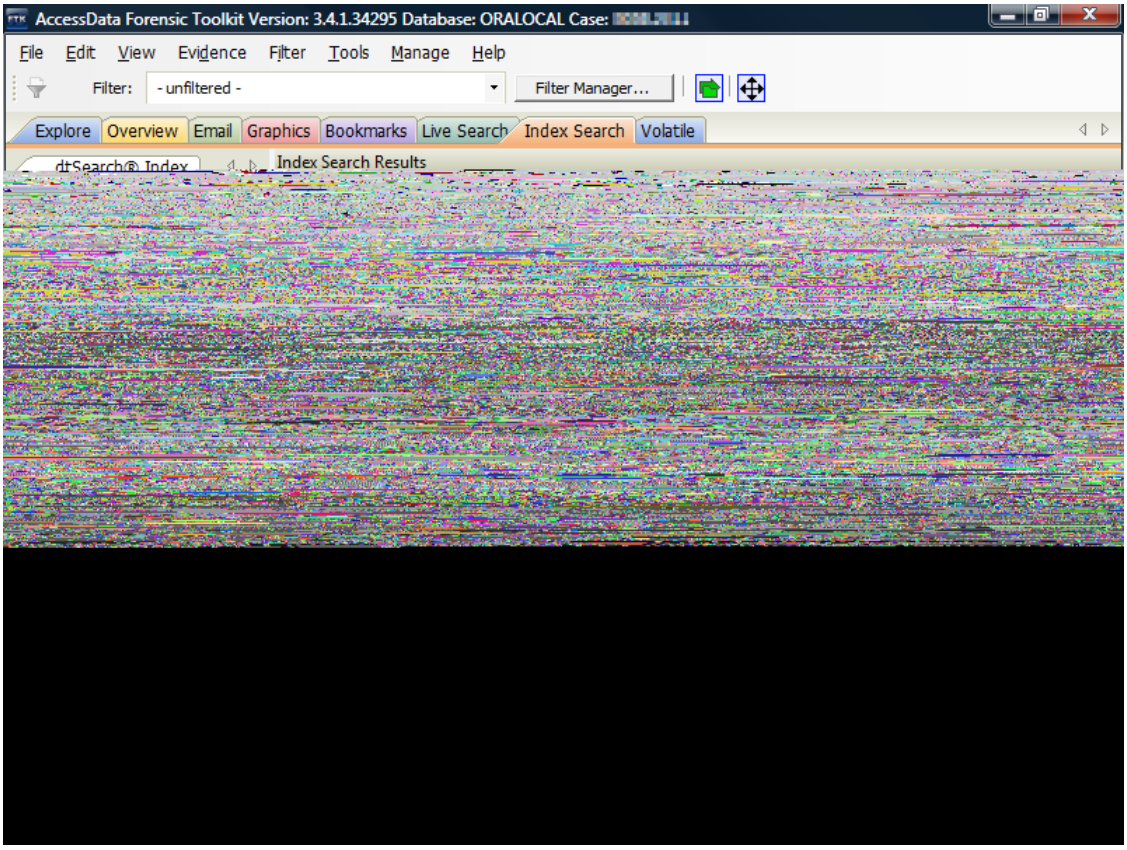


Figura 2.6: Tela de resultado de busca textual no FTK com expansão de uma categoria, mostrando a ordenação pelo número decrescente de *hits* do documento.

2.6.2 ENCASE

O *EnCase* (Guidance, 2011) é uma ferramenta forense desenvolvida pela empresa norte-americana Guidance Software, a qual permite buscas textuais exaustivas e indexadas. Os resultados das buscas podem ser ordenados por uma das colunas de atributos dos arquivos. A ferramenta não agrupa os arquivos por similaridade. A forma de funcionamento descrita pode ser visualizada na Figura 2.7.

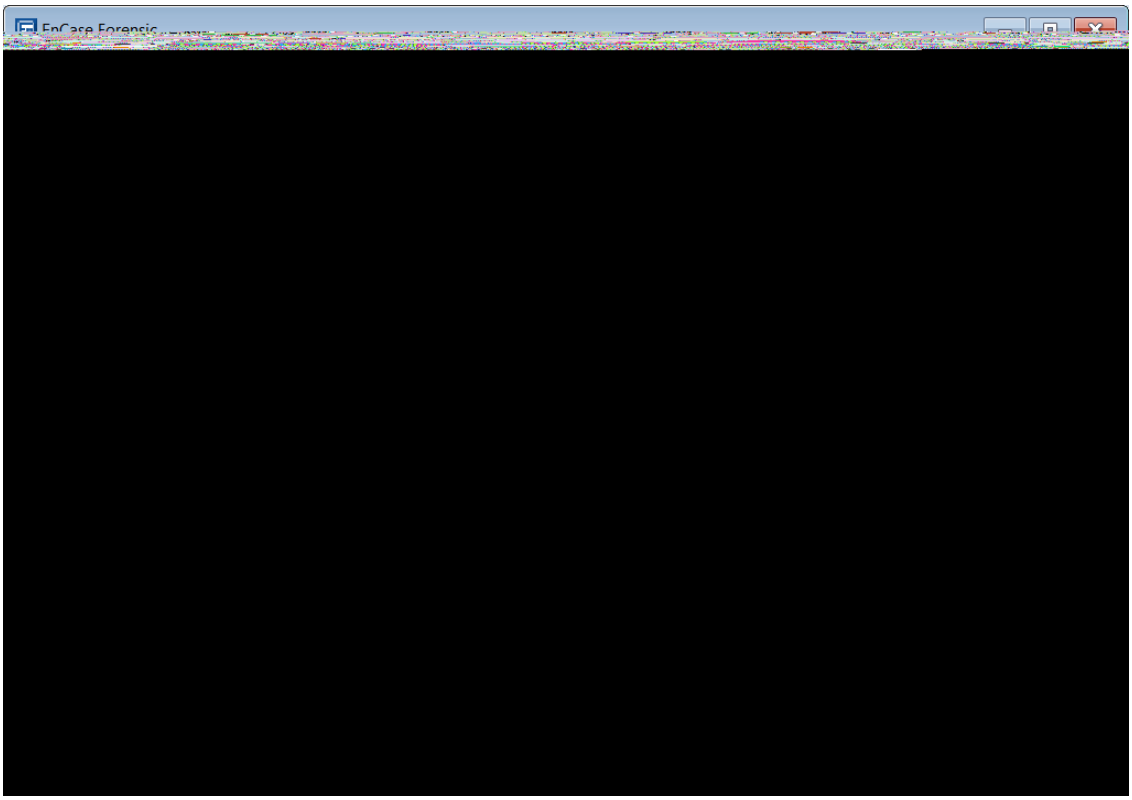


Figura 2.7: Tela de resultado de busca textual no *EnCase*.

2.6.3 THE SLEUTH KIT

A ferramenta livre formada pela combinação TSK/*Autopsy* (Carrier, 2011) é gratuita e seu código-fonte é aberto. Permite buscas textuais tanto exaustivas quanto indexadas. Os resultados das buscas são ordenados pelo número do setor do arquivo de imagem onde o *hit* foi localizado. A ferramenta não aponta diretamente qual arquivo contém cada *hit*, e não agrupa os *hits* por similaridade. A forma de funcionamento descrita pode ser visualizada na Figura 2.8.

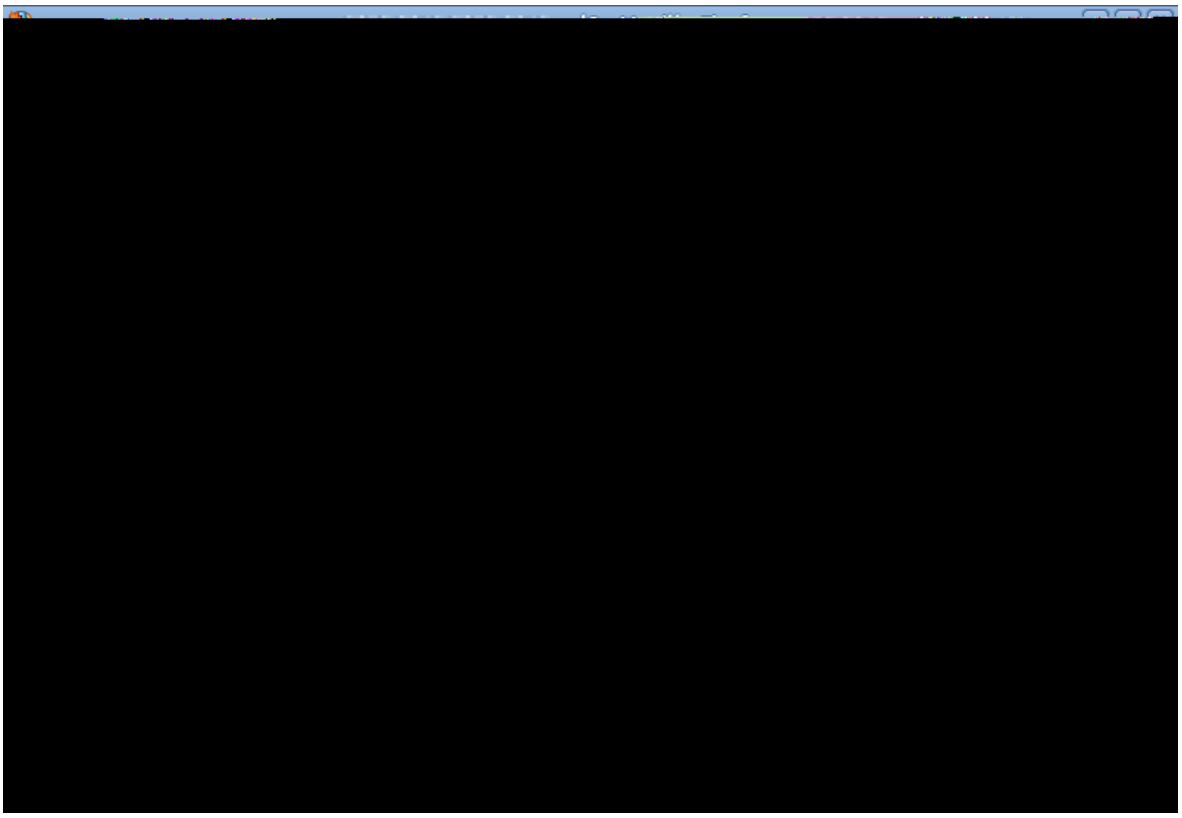


Figura 2.8: Tela de resultado de busca textual no TSK/*Autopsy*.

Neste Capítulo foram apresentados os principais conceitos, definições, *frameworks* e ferramentas relacionados ao domínio da Informática Forense, no qual esta pesquisa é aplicada. O Capítulo 3 discute os fundamentos do método escolhido para o agrupamento de *hits* com conteúdos assemelhados em coleções textuais forenses.

CAPÍTULO 3

TRATAMENTO DE INFORMAÇÃO E REDES NEURAIS ARTIFICIAIS

A proposta deste trabalho consiste em agrupar *hits* com conteúdo assemelhado em coleções textuais forenses, e para tanto utiliza métodos e conceitos de várias disciplinas. São elas: Recuperação de Informação, Mineração de Texto, Agrupamento de Dados e Redes Neurais Artificiais. Essas áreas do conhecimento não são isoladas e por vezes não possuem fronteiras bem definidas; as Redes Neurais Artificiais, por exemplo, embora componham um ramo estabelecido da Inteligência Artificial, são encapsuladas pela Mineração de Texto como um dos mecanismos que podem ser usados na tarefa de agrupamento de documentos. Além disso, todas as áreas se beneficiam de técnicas de aprendizado de máquina. Conhecimentos de todas essas áreas serão aplicados para atingir os objetivos descritos na Seção 1.2.

Neste capítulo são apresentados os fundamentos teóricos desta pesquisa. A Seção 3.1 aborda a Recuperação de Informação e as peculiaridades de sua aplicação ao domínio da Informática Forense. A Seção 3.2 discorre sobre Mineração de Texto. A Seção 3.3 apresenta conceitos de Agrupamento de Dados. Por fim, a Seção 3.4 introduz de forma geral as Redes Neurais Artificiais, e de forma mais específica a Teoria da Ressonância Adaptativa, cuja arquitetura de rede ART1 utilizada nesta pesquisa é estudada em detalhes na Seção 3.5.

3.1 RECUPERAÇÃO DE INFORMAÇÃO

Recuperação de Informação, de forma geral, consiste em buscar documentos que possam satisfazer necessidades de informação. É uma área de estudo multidisciplinar que utiliza conceitos e técnicas de diversas ciências, dentre as quais podem ser citadas Computação,

Matemática, Estatística e Biblioteconomia. Manning et al. (2008) assim definem Recuperação de Informação (tradução livre):

> *Recuperação de Informação (RI) é a busca de material (normalmente documentos) de natureza não-estruturada (normalmente texto) no interior de coleções volumosas (normalmente armazenadas em computadores) que satisfaz uma necessidade de informação.*

Também deve ser citada a definição de Lancaster *apud* van Rijsbergen (1979) (tradução livre):

> *Recuperação de Informação é o termo comumente, embora um tanto incorretamente, aplicado ao tipo de atividade discutido neste livro. Um sistema de recuperação de informação não informa (no sentido de modificar o grau de conhecimento) o usuário sobre o assunto de sua consulta. Ele apenas informa sobre a existência (ou inexistência) e localização dos documentos relacionados à sua consulta.*

De posse das duas definições, é possível mapear os elementos principais das definições de Recuperação de Informação e dos exames periciais *postmortem* de Informática Forense, conforme a Tabela 3.1.

Tabela 3.1: Recuperação de Informação aplicada à Informática Forense

Recuperação de Informação	Informática Forense
Documentos	Arquivos e outros artefatos textuais
Coleções volumosas	Mídias de armazenamento
Necessidades de informação	Quesitos

Hearst *apud* Baeza-Yates e Ribeiro-Neto (1999, Cap. 10), descreveu um modelo padrão do processo de acesso à informação composto por oito atividades básicas, a saber:

1. Comece com a necessidade de informação.
2. Selecione um sistema e as coleções nas quais deseja efetuar a busca.
3. Formule a consulta.
4. Submeta a consulta ao sistema.
5. Receba o resultado na forma de itens de informação.
6. Percorra, avalie e interprete os resultados.
7. Pare, ou,
8. Reformule a consulta e vá para o passo 4.

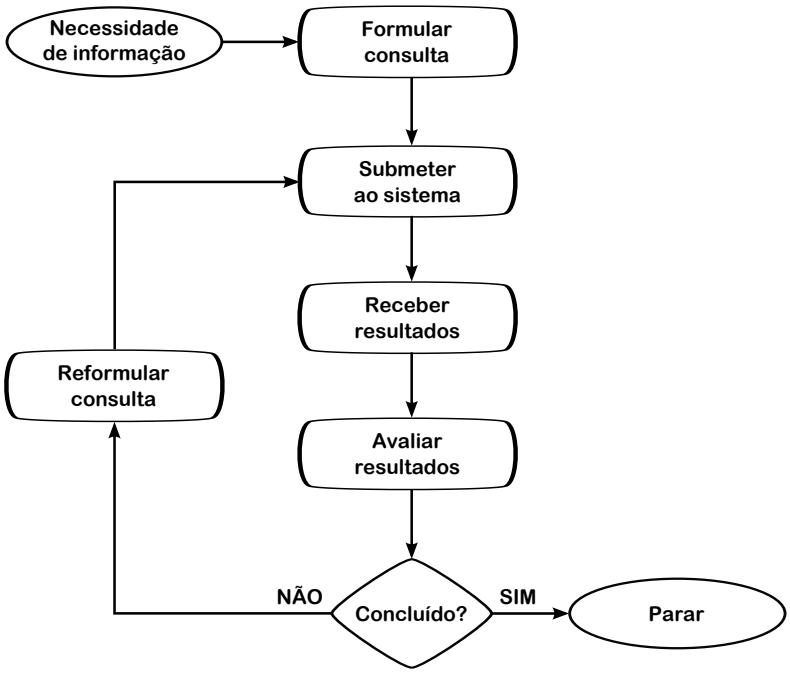


Figura 3.1: Modelo padrão de acesso à informação, adaptado de Hearst *apud* Baeza-Yates e Ribeiro-Neto (1999, Cap. 10)

O fluxograma da Figura 3.1 ilustra as atividades básicas do modelo, as quais podem ser encontradas em buscadores de Internet (Google, Bing), sistemas operacionais (Windows, Linux) e em ferramentas de Informática Forense. Hearst *apud* Baeza-Yates e Ribeiro-Neto (1999, Cap. 10) aponta uma deficiência nesse modelo, a saber (tradução livre):

> *Este modelo não leva em conta o fato que muitos usuários não gostam de ser confrontados com uma lista longa e desorganizada de resultados que não tratam diretamente suas necessidades de informação* [as do usuário]. (esclarecimento adicionado pelo tradutor)

Buscadores de Internet contornam essa deficiência do modelo através de algoritmos de priorização (*ranking*). Dessa forma, embora uma consulta possa retornar milhares de *hits*, aqueles considerados com maior probabilidade de ser relevantes são apresentados no topo da lista, e assim o usuário não necessita percorrer de forma exaustiva a lista para que encontre a informação que procura. Um exemplo é o algoritmo de priorização do Google, que a empresa afirma utilizar mais de duzentas variáveis[1]. Uma variável citada explicitamente é o *PageRank* (Page et al., 1999), que determina a importância de uma página examinando as outras páginas que se interligam a ela; e há variáveis para as quais é dada apenas uma descrição vaga: *"... processamos informações incluídas nos principais atributos e tags de conteúdo, como tags Title e atributos ALT."*[2].

O buscador Blekko[3] usa, entre outras, uma variável comparável com o *PageRank* do Google incrementada pela distribuição geográfica das páginas que se interligam a uma dada página[4]. A arquitetura da *World Wide Web* (WWW) contempla diversos recursos que os buscadores podem e efetivamente utilizam para oferecer melhores resultados de busca, a exemplo de análise de interligações entre páginas, geolocalização e recomendações feitas pelos usuários. Esses recursos não existem no domínio dos exames *postmortem* de Informática Forense, que trata de coleções estáticas presentes em mídias de armazenamento.

A deficiência apontada por Hearst *apud* Baeza-Yates e Ribeiro-Neto (1999, Cap. 10) está presente nas ferramentas FTK (AccessData, 2011), *EnCase* (Guidance, 2011) e TSK/ *Autopsy* (Carrier, 2011), conforme discutido na Seção 2.6. Nesse contexto, o problema torna-se particularmente agudo porque as buscas em coleções textuais forenses frequentemente trazem de centenas a milhares de *hits*. Torna-se desejável, portanto, desenvolver técnicas e mecanismos que permitam ao perito revisar com maior agilidade os *hits* retornados pelas buscas em coleções textuais forenses.

[1] `http://sites.google.com/site/webmasterhelpforum/en/faq--crawling--indexing---ranking#pagerank`, acessado em 03/08/2011

[2] `http://www.google.com/support/webmasters/bin/answer.py?answer=70897`, acesso em 03/08/2011.

[3] `http://www.blekko.com`

[4] `http://help.blekko.com/index.php/what-information-is-available-on-the-seo-pages/`, acesso em 03/08/2011.

3.2 MINERAÇÃO DE TEXTO

Segundo Hotho et al. (2005), Mineração de Texto é uma área multidisciplinar do conhecimento que agrega técnicas e algoritmos de Recuperação de Informação, Processamento de Linguagem Natural, Mineração de Dados, Aprendizado de Máquina e Estatística, entre outras. Os autores ressaltam que a definição de Mineração de Texto depende da área de pesquisa, e oferecem três possíveis definições:

- Extração de Informação: abordagem que busca extrair fatos a partir de textos;
- Mineração de Dados Textuais (*Text Data Mining*): aplica algoritmos e métodos de aprendizado de máquina e estatística com o objetivo de encontrar padrões úteis em textos;
- Processo de Descoberta de Conhecimento em Bases de Dados (*Knowledge Discovery in Databases*, ou KDD): extração de informações ainda desconhecidas em grandes coleções textuais.

Esta dissertação utiliza a definição de Mineração de Texto apresentada por Hotho et al. (2005) como Mineração de Dados Textuais. Primeiramente serão mostradas as tarefas de pré-processamento que precedem a criação de um índice textual, e em seguida será apresentada uma técnica de representação de documentos. Todas essas tarefas e técnicas serão utilizadas para preparar os documentos que serão apresentados ao algoritmo ART1.

3.2.1 PRÉ-PROCESSAMENTO

Navarro *apud* Baeza-Yates e Ribeiro-Neto (1999, Cap. 8) afirma que as buscas em sistemas de Recuperação de Informação podem ser executadas de duas formas:

- Sequencialmente, em todos os documentos;
- Através da consulta a um índice construído previamente.

O autor afirma que é válido construir um índice quando a coleção de documentos é grande e *semi-estática*, isto é, pode ser atualizada em intervalos regulares. Dessa forma, é válido construir índices para as coleções textuais tratadas nos exames periciais *postmortem* de Informática Forense, que são completamente estáticas (vide Seção 2.5.1) porque os arquivos de imagem que as contêm não necessitam (nem devem) ser modificados. Essa abordagem

torna mais rápidas as buscas no domínio da Informática Forense, conforme demonstrado por Johansson (2003).

Segundo Ziviani *apud* Baeza-Yates e Ribeiro-Neto (1999, Cap. 7), nem todas as palavras de um documento são igualmente significativas para representar sua semântica. Assim, é considerado válido pré-processar os documentos para determinar que palavras ou termos serão indexados. Três tarefas de pré-processamento discutidas pelo autor são:

- Análise léxica – consiste em converter um fluxo de caracteres (o texto de um documento) em um fluxo de palavras (termos candidatos a ser indexados). Esta tarefa deve levar em conta não apenas espaços, mas também números, hífens, sinais de pontuação, e letras maiúsculas e minúsculas. O autor argumenta que números geralmente não são bons termos para indexar, porque ficam vagos ao ser retirados de seu contexto;

- Eliminação de *stopwords* – consiste em eliminar palavras que são muito frequentes na coleção e, por isso, pouco discriminantes. Artigos, preposições e conjunções são candidatos naturais a estar presentes em uma lista de *stopwords*. Um dos benefícios desta técnica é a redução do tamanho do índice. Um possível malefício é a redução do *recall*[5] nos casos em que o termo buscado é uma *stopword*.

- *Stemming* – consiste em extrair de uma palavra as variações afixais (prefixos e/ou sufixos) para que esta seja reduzida à sua "raiz" (*stem*) ou radical. Por exemplo, as palavras *música* e *músicos* poderiam ser reduzidas ao radical *music*. Esta técnica ajuda a reduzir o tamanho do índice, pois palavras com o mesmo *stem* são representadas por um único termo no índice. Outro possível benefício é que documentos que contenham formas flexionadas (gênero e número para substantivos, conjugações para verbos) do termo de busca serão recuperados, aumentando o *recall*. Por esse mesmo motivo, um possível malefício é a redução da precisão[6].

A proposta deste trabalho, descrita em detalhes no Capítulo 4, inclui o uso de um índice, em cuja construção são realizadas as tarefas de pré-processamento elencadas acima.

[5]Métrica usada em Recuperação de Informação que mede a porcentagem dos documentos relevantes para a consulta, de toda a coleção, que são efetivamente retornados pela consulta.

[6]Métrica usada em Recuperação de Informação que mede a porcentagem dos documentos relevantes para a consulta, entre aqueles retornados pela consulta.

3.2.2 REPRESENTAÇÃO DE DOCUMENTOS

Hotho et al. (2005) citam que, para minerar coleções textuais, é necessário pré-processar os documentos e armazenar suas informações em estruturas de dados mais apropriadas que arquivos de texto plano para que os algoritmos possam processá-los. Uma abordagem muito utilizada é a que representa um documento com base no conjunto de palavras que ele contém: todas as palavras são jogadas em um "saco" (*bag-of-words*, ou BOW), sem levar em conta a ordem e o contexto em que aparecem (Manning et al., 2008). O conjunto de palavras distintas extraídas de todos os documentos da coleção é chamado de *dicionário* da coleção.

A partir da definição da BOW é possível montar uma representação atributo-valor e gerar uma tabela cujas entradas contêm informações relacionadas à presença ou freqüência de cada palavra (Soares et al., 2008). Uma coleção de documentos pode então ser representada por uma matriz onde cada documento é uma linha e cada palavra (termo) é uma coluna (dimensão). Desta forma, uma coleção que contém N documentos e M termos corresponde a uma matriz $N \times M$ onde a entrada a_{nm} é um atributo que representa a presença ou freqüência do termo t_m no documento d_n, conforme apresentado na Tabela 3.2.

Tabela 3.2: Representação da *bag-of-words* (BOW).

	t_1	t_2	$\ldots$	t_M
d_1	a_{11}	a_{12}	$\ldots$	a_{1M}
d_2	a_{21}	a_{22}	$\ldots$	a_{2M}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
d_N	a_{N1}	a_{N2}	$\ldots$	a_{NM}

Ainda segundo Soares et al. (2008), cada entrada a_{nm} associada ao documento d_n e ao termo t_m pode ser preenchida de diversas maneiras: presença ou ausência da palavra, número absoluto de aparições da palavra (freqüência absoluta), ou a freqüência relativa da palavra na coleção.

Este trabalho utiliza a abordagem da BOW para representar os documentos que serão processados pelo algoritmo ART1. A medida utilizada para representar os atributos (termos) é a binária, na qual a entrada a_{nm} da BOW recebe o valor 1 (verdadeiro) se o

documento d_n contém o termo t_m, e 0 (falso) caso contrário. A BOW é representada por uma matriz de incidência binária. A representação é mostrada na Equação 3.1:

$$a_{nm} = \begin{cases} 1 & \text{se } t_m \text{ ocorre em } d_n \\ 0 & \text{caso contrário} \end{cases} \tag{3.1}$$

A Tabela 3.3 apresenta um exemplo de BOW.

Tabela 3.3: Exemplo de BOW.

	pedra	papel	tesoura
d_1	1	0	0
d_2	1	1	0
d_3	0	1	0
d_4	0	0	1

O conjunto de todos os documentos (ou seja, a coleção) é definido como D, e o conjunto $T = \{t_1, \ldots, t_m\}$ é definido como o dicionário de D, isto é, todas as diferentes palavras que ocorrem em todos os documentos $d_i \in D$.

3.3 AGRUPAMENTO DE DADOS

Jain e Dubes (1988) afirmam que a base de grande parte da ciência consiste na prática de classificar objetos de acordo com suas similaridades observadas, e que organizar dados em agrupamentos naturalmente perceptíveis é um dos modos mais fundamentais de compreender e aprender. Os autores oferecem a seguinte definição, em tradução livre, para análise de agrupamentos (*cluster analysis*):

> *Análise de agrupamentos é o estudo formal de algoritmos e métodos para agrupar, ou classificar, objetos.*

Jain (2010) define o objetivo do agrupamento de dados, também chamado na literatura de *análise de agrupamentos*, da seguinte forma (tradução livre):

> *O objetivo do agrupamento de dados, também conhecido como análise de agrupamentos, é descobrir grupamentos naturais de um conjunto de padrões, pontos, ou objetos.*

Pela definição acima, os objetos a agrupar podem ser pontos em um espaço. Um exemplo de agrupamento de pontos em um espaço é ilustrado na Figura 3.2.

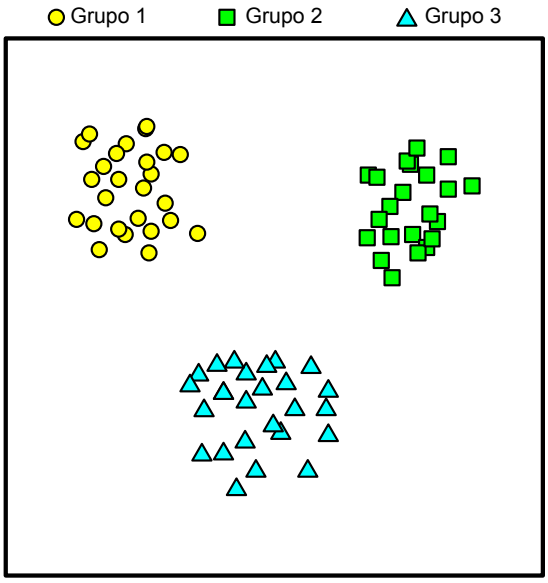


Figura 3.2: Exemplo de agrupamento de dados: pontos em um espaço bidimensional

Segundo Everitt *apud* Jain e Dubes (1988), a definição de *grupo* é (tradução livre):

> *Um grupo é um conjunto de entidades assemelhadas, e entidades de grupos diferentes não são assemelhadas.*

Conforme descrito na Seção 1.1.1, agrupamento de dados não conta com nenhum tipo de auxílio externo para realizar sua tarefa, portanto, é uma técnica de aprendizado de máquina não-supervisionado. Como o agrupamento busca encontrar estrutura em conjuntos de dados sobre os quais se possui pouco ou nenhum conhecimento, pode ser descrito como uma técnica de análise de dados exploratória (Tukey, 1980).

A disciplina de agrupamentos de dados tem décadas de existência e uma literatura extensa. Os conceitos que seguem serão apresentados sem aprofundar além do escopo desta pequisa.

Jain e Dubes (1988) descrevem dois grandes métodos de análise de agrupamentos: hierárquicos e particionais.

- Métodos hierárquicos organizam os dados em uma sequência aninhada de grupos;
- Métodos particionais determinam uma partição dos padrões de entrada em um certo número de grupos, tal que os padrões contidos em um grupo são mais similares uns aos outros que aos padrões contidos em outros grupos.

Jain (2010) divide os métodos particionais em dois tipos:

- Rígidos (*hard assignment*): cada um dos pontos é associado a um único grupo;
- Sobrepostos (*soft assignment*): cada um dos pontos pode pertencer a mais de um grupo.

Jain e Dubes (1988) afirmam que (tradução livre)

> *Partições resultam de algoritmos de agrupamento particionais (...) Partições também são obtidas a partir de informações de categorias.*

Dessa forma, é possível descrever soluções de agrupamento através de partições do conjunto de objetos que de deseja agrupar. A descrição a seguir é adaptada de Vinh et al. (2009).

Sejam $S = \{s_1, s_2, \ldots, s_N\}$ um conjunto de N objetos, e $U = \{u_1, u_2, \ldots, u_R\}$ e $V = \{v_1, v_2, \ldots, v_C\}$ duas partições dos objetos de S tal que $\bigcup_{i=1}^{R} u_i = S = \bigcup_{j=1}^{C} v_j$ e $u_i \cap u_{i'} = \emptyset = v_j \cap v_{j'}$ para $1 \leq i \neq i' \leq R$ e $1 \leq j \neq j' \leq C$.

As duas partições U e V possuem R grupos e C classes, respectivamente. Estão descritas duas soluções de agrupamento: $U = \{u_1, u_2, \ldots, u_R\}$, obtida através de um algoritmo de agrupamento; e $V = \{v_1, v_2, \ldots, v_C\}$, obtida a partir da classificação dos objetos em classes padrão por um especialista, que será discutida em maior detalhe na Seção 3.3.2.

Um algoritmo particional considerado clássico na literatura de agrupamento é o k-médias (Jain, 2010), para o qual existem diversas implementações e variações; um algoritmo muito utilizado é o de Lloyd (1982). O algoritmo recebe como parâmetro o número de grupos k nos quais os padrões de entrada devem ser alocados. O exemplo mostrado na Figura 3.2 pode ser obtido com o valor $k = 3$.

Os autores afirmam que métodos particionais são especialmente apropriados para a representação eficiente e a compressão de grandes bases de dados. É exatamente o cenário encontrado nos exames periciais *postmortem* de Informática Forense, conforme apresentado na Tabela 3.1. Este trabalho utiliza um método particional rígido, onde os objetos pertencem a um único grupo. Métodos hierárquicos não serão abordados nesta dissertação.

3.3.1 AGRUPAMENTO DE DOCUMENTOS

Quando o conjunto de objetos a agrupar é uma coleção de documentos, o conceito de agrupamento de dados pode ser estendido para agrupamento de documentos (*document clustering*), definido por Andrews e Fox (2007) como (tradução livre):

> *O processo de agrupamento tem por objetivo descobrir grupamentos naturais, e assim apresentar uma visão geral das classes (tópicos) presentes em uma coleção de documentos.*

Conforme apresentado na Seção 3.3, faz sentido utilizar agrupamento para comprimir o espaço de informação. Um exemplo conceitual de agrupamento de documentos é ilustrado na Figura 3.3.

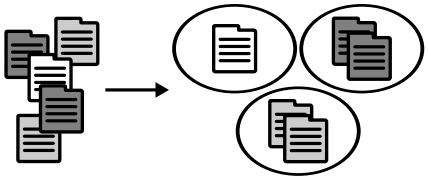


Figura 3.3: Exemplo conceitual de agrupamento de documentos

van Rijsbergen (1979) enunciou a hipótese de agrupamento (*cluster hypothesis*), que segue em tradução livre:

> *Documentos estreitamente associados tendem a ser relevantes para as mesmas requisições* [buscas]. (esclarecimento adicionado pelo tradutor)

Conforme discutido na Seção 2.5.1, a pesquisa por palavras-chave é um dos métodos mais utilizados para identificar rapidamente arquivos de interesse durante o exame pericial, porém está sujeita ao problema apresentado por Hearst *apud* Baeza-Yates e Ribeiro-Neto (1999, Cap. 10), de que *"(...) muitos usuários não gostam de ser confrontados com uma lista longa e desorganizada de resultados que não tratam diretamente suas necessidades de informação."*. Não é incomum que o perito, ao conduzir suas buscas, seja confrontado com listas de centenas ou até milhares de *hits*. Esse é um cenário em que o agrupamento de documentos pode ser aplicado para organizar os *hits* em grupos com conteúdos assemelhados.

Supondo que a hipótese de agrupamento de van Rijsbergen (1979) seja verdadeira, se o perito, ao submeter uma busca textual na aplicação, receber como retorno grupos de *hits* com conteúdo assemelhado, poderá avaliar apenas uma pequena quantidade de *hits* em um determinado grupo e julgar previamente sua relevância. Desta forma, o grupo poderá ser marcado como importante para uma análise mais aprofundada, ou poderá receber uma menor prioridade, para ser revisado depois dos grupos julgados mais importantes, podendo ser até mesmo descartado. Dessa forma, supõe-se que o agrupamento de documentos

pode tornar mais breve o exame pericial, porque quando o perito localizar um documento relevante, não necessitará percorrer exaustivamente a lista de *hits* para localizar todos os outros documentos com conteúdos assemelhados, bastando-lhe examinar os documentos contidos no grupo onde foi localizado o documento relevante.

3.3.2 VALIDAÇÃO

Segundo Jain (2010), os algoritmos de agrupamento tendem a encontrar grupos nos padrões de entrada, estejam esses grupos presentes ou não. Dessa forma, é necessário validar as análises de agrupamento produzidas. Uma abordagem muito utilizada e discutida na literatura é a da validação objetiva através de índices. Os índices devem seguir critérios bem definidos.

Jain e Dubes (1988) descrevem a relação entre grupos, critérios, índices e validação da seguinte forma (tradução livre):

> *(...) Um critério expressa a estratégia pela qual uma estrutura de agrupamento será validada, enquanto um índice é uma estatística nos termos da qual a validade será testada.*

Neste ponto é necessário definir o que é um índice e descrever os critérios nos quais os índices se baseiam. Jain e Dubes (1988) descrevem um índice como uma estatística que deve possuir determinadas propriedades (tradução livre):

> *O índice deve fazer sentido intuitivamente, deve ter uma base teórica, e deve ser prontamente calculável.*

Jain e Dubes (1988) descrevem um critério como uma estratégia de validação. Jain (2010) elenca três tipos de critérios:

- Internos: medem a qualidade de um agrupamento com base apenas nos próprios dados;
- Externos: medem a correspondência entre o agrupamento gerado pelo algoritmo e as classes padrão pré-estabelecidas por especialistas (*ground truth*);
- Relativos: comparam agrupamentos entre si, produzidos a partir de algoritmos ou parâmetros diferentes, para decidir qual deles é o melhor de acordo com algum critério.

O índice utilizado para validação nesta pesquisa é baseado em um critério externo, cuja definição deve ser discutida com maior profundidade.

Segundo Jain e Dubes (1988), a definição de *critério externo* para expressar a validade de uma estrutura de agrupamento pode ser dada como (tradução livre):

> *Critérios externos medem o desempenho através da correspondência de uma estrutura de agrupamento com informação* a priori. *Por exemplo, um critério externo mede o grau de correspondência entre números de grupos, obtidos a partir de um algoritmo de agrupamento, e rótulos de categorias, determinados* a priori.

Conforme exposto, neste trabalho foi utilizado um índice baseado em critério externo, conforme definido por Jain e Dubes (1988), para avaliar objetivamente as soluções de agrupamento obtidas. Os grupos obtidos através do algoritmo de agrupamento de forma não-supervisionada (sem ajuda externa) serão comparados às classes padrão pré-estabelecidas por um especialista (*ground truth*). As classes são consideradas "corretas", isto é, refletem uma partição "perfeita" ou "ótima" dos documentos da coleção, conforme definido por Manning et al. (2008). Desta forma, do ponto de vista objetivo, o agrupamento será considerado tão bom quanto mais os grupos obtidos se aproximarem das classes padrão.

Índices internos obtêm valores tão mais altos quanto é maior a similaridade intra-grupo e menor a similaridade entre grupos diferentes. Esta pesquisa trabalha com conjuntos de dados de alta dimensionalidade, para os quais o conceito de distância tem pouco poder discriminante. Segundo Beyer et al. (1999), em cenários de alta dimensionalidade, um dado ponto no espaço tende a ser quase equidistante de seus vizinhos mais próximo e mais distante; desta forma, com a alta dimensionalidade envolvida nos conjuntos de dados tratados pela Informática Forense, este critério é inadequado. Além disso, Manning et al. (2008) afirmam que (tradução livre):

> *(...) boas pontuações em um critério interno não necessariamente se traduzem em boa efetividade em uma aplicação.*

Portanto, não serão calculados índices internos.

Como neste trabalho foi utilizado um único algoritmo de agrupamento (ART1), para cujos parâmetros não foi feita uma pesquisa exaustiva em busca de valores considerados ótimos, não serão calculados índices relativos.

Uma forma de expressar soluções de agrupamento é descrita e apresentada na Seção 3.3.3.

3.3.3 ÍNDICE EXTERNO – NMI

A descrição a seguir é adaptada de Vinh et al. (2009). Sejam $S = \{s_1, s_2, \ldots, s_N\}$ um conjunto de N objetos, e $U = \{u_1, u_2, \ldots, u_R\}$ e $V = \{v_1, v_2, \ldots, v_C\}$ duas partições dos objetos de S tal que $\bigcup_{i=1}^{R} u_i = S = \bigcup_{j=1}^{C} v_j$ e $u_i \cap u_{i'} = \emptyset = v_j \cap v_{j'}$ para $1 \leq i \neq i' \leq R$ e $1 \leq j \neq j' \leq C$. As duas partições U e V possuem R grupos e C classes, respectivamente. Seja $a_i = |u_i|$ a quantidade de objetos do grupo $u_i \in U$ e, de forma análoga, $b_j = |v_j|$ a quantidade de objetos da classe $v_j \in V$. É possível representar a sobreposição entre U e V através de uma tabela de contingência, como apresentado na Tabela 3.4.

Tabela 3.4: Tabela de contingência

	v_1	v_2	$\ldots$	v_C	
u_1	N_{11}	N_{12}	$\ldots$	N_{1C}	a_1
u_2	N_{21}	N_{22}	$\ldots$	N_{2C}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
u_R	N_{R1}	N_{R2}	$\ldots$	N_{RC}	a_R
	b_1	b_2	$\ldots$	b_C	N

O próximo passo é calcular um índice externo para comparar os grupos calculados pelo algoritmo e as classes padrão definidas por especialistas. O índice externo utilizado neste trabalho é a *Normalized Mutual Information*, ou NMI, que possui uma sólida fundamentação teórica em Teoria da Informação (Strehl et al., 2002). A NMI foi aplicada em um estudo de caso de agrupamento de documentos por Ghosh (2003, Cap. 10), que justificou a escolha sob o argumento de que a NMI é uma medida não-tendenciosa da utilidade do conhecimento obtido através do agrupamento em predizer rótulos de classe; isto é, a NMI mede a utilidade dos elementos dos grupos como preditores dos elementos correspondentes das classes.

Serão necessários alguns conceitos elementares de Teoria da Informação, cujas definições que seguem são adaptadas de Cover e Thomas (2006) e Vinh et al. (2009).

A entropia de informação $H(X)$ de uma variável aleatória discreta X, que pode assumir apenas valores pertencentes ao seu domínio $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ com uma função de probabilidade $p(x)$, é definida conforme apresentado na Equação 3.2:

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) \tag{3.2}$$

É comum usar logaritmos de base 2 para calcular a entropia, que nesse caso é medida em *bits* (Cover e Thomas, 2006). A entropia pode ser entendida como uma medida da incerteza de uma variável aleatória. Também representa o número de *bits* necessários, em média, para descrever uma variável aleatória. Por exemplo, duas variáveis aleatórias X e Y de mesmo domínio $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ mas com funções de probabilidade respectivas $p(x)$ e $q(x)$ diferentes, onde a distribuição de $p(x)$ é uniforme (todos os valores são igualmente prováveis) e a distribuição de $q(x)$ é não-uniforme (os valores têm probabilidades diferentes, alguns mais prováveis e outros menos), têm entropias diferentes; a incerteza associada à variável aleatória X é maior.

A entropia condicional $H(X|Y)$ é uma medida da incerteza de uma variável aleatória X quando condicionada ao conhecimento de outra variável aleatória Y. É a entropia média de X para todos os valores de Y. A probabilidade conjunta das duas variáveis é expressa por $p(x, y)$, conforme apresentado na Equação 3.3:

$$\begin{aligned} H(X|Y) &= \sum_{y \in \mathcal{Y}} p(y) H(X|Y = y) \\ &= -\sum_{y \in \mathcal{Y}} p(y) \sum_{x \in \mathcal{X}} p(x|y) \log p(x|y) \\ &= -\sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p(x, y) \log p(x|y) \end{aligned} \tag{3.3}$$

A Informação Mútua (*Mutual Information* – MI) $I(X, Y)$ de duas variáveis discretas X e Y com domínios respectivos $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ e $\mathcal{Y} = \{y_1, y_2, \ldots, y_n\}$ e funções de probabilidade $p(x)$ e $p(y)$ é definida conforme apresentado na Equação 3.4:

$$I(X, Y) = H(X) - H(X|Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{3.4}$$

A informação mútua entre duas variáveis aleatórias representa a redução que uma delas provoca na incerteza (entropia) da outra; ou seja, é uma medida da quantidade de informação que uma variável aleatória contém sobre outra.

A informação mútua é uma medida simétrica, isto é, $I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$. Ou seja, Y oferece tanta informação sobre X quanto X oferece sobre Y. Esta propriedade sugere que a informação mútua pode ser usada para medir a informação compartilhada por dois agrupamentos e, portanto, avaliar a similaridade entre eles. Para tanto, é necessário descrever as soluções de agrupamento em um contexto probabilístico.

Novamente, sejam $S = \{s_1, s_2, \ldots, s_N\}$ um conjunto de N objetos, e $U = \{u_1, u_2, \ldots, u_R\}$ e $V = \{v_1, v_2, \ldots, v_C\}$ duas partições dos objetos de S. Ao escolher aleatoriamente um objeto de S, a probabilidade que esse objeto pertença ao grupo u_i é definida conforme a Equação 3.5.

$$p(i) = \frac{|u_i|}{N} \tag{3.5}$$

Através das Equações 3.2 e 3.5, é possível obter a entropia associada à partição U, expressa na Equação 3.6.

$$H(U) = -\sum_{i=1}^{R} p(i) \log p(i) \tag{3.6}$$

$H(U)$ é não-negativa e assume o valor 0 quando não há incerteza em determinar a qual grupo um objeto pertence, que é quando existe apenas um único grupo. A entropia $H(V)$ da partição V é obtida de maneira análoga. A partir desses elementos e da Equação 3.4, é possível calcular a Informação Mútua $I(U, V)$ entre as partições U e V de acordo com a Equação 3.7.

$$I(U, V) = \sum_{i=1}^{R} \sum_{j=1}^{C} p(i, j) \log \frac{p(i, j)}{p(i)p(j)} \tag{3.7}$$

A probabilidade de que o objeto pertença ao grupo $u_i \in U$ e à classe $v_j \in V$ é expressa por $p(i, j)$, conforme apresentado na Equação 3.8.

$$p(i, j) = \frac{|u_i \cap v_j|}{N} \tag{3.8}$$

Segundo Strehl et al. (2002), a Informação Mútua é uma medida que fornece uma boa indicação da informação compartilhada entre dois agrupamentos. É possível demonstrar que $I(U, V)$ é uma métrica ou distância. Como não há limite superior para $I(U, V)$, foi proposta uma versão normalizada que varia entre 0 e 1.

Com base nas equações 3.6 e 3.7, a Informação Mútua Normalizada (NMI) é definida conforme a Equação 3.9.

$$NMI(U, V) = \frac{I(U, V)}{\sqrt{H(U) \times H(V)}} \tag{3.9}$$

A NMI é a base da estimativa $\phi^{(NMI)}$ proposta por Strehl et al. (2002), definida conforme a Equação 3.10:

$$\phi^{(NMI)}(U,V) = \frac{\sum_{i=1}^{R}\sum_{j=1}^{C} N_{ij} \times \log(\frac{N_{ij}\times N}{a_i \times b_j})}{\sqrt{\sum_{i=1}^{R} a_i \times \log(\frac{a_i}{N}) \times \sum_{j=1}^{C} b_j \times \log(\frac{b_j}{N})}} \tag{3.10}$$

É possível calcular $\phi^{(NMI)}$ a partir das células da tabela de contingência descrita na Tabela 3.4. Deve ser notado que agrupamentos idênticos ($U = V$) têm $\phi^{(NMI)} = 1$, enquanto agrupamentos independentes, isto é, que não compartilham nenhuma informação entre si, têm $\phi^{(NMI)} = 0$.

Ghosh (2003, Cap. 10) aplicou a NMI em um estudo de caso de agrupamento de documentos. Segundo o autor, a estimativa $\phi^{(NMI)}$ é uma medida não-tendenciosa da utilidade do conhecimento obtido através do agrupamento em predizer rótulos de classe; isto é, a NMI mede a utilidade dos elementos dos grupos como preditores dos elementos correspondentes das classes.

Neste trabalho a estimativa $\phi^{(NMI)}$ é usada como índice externo de validação.

3.4 REDES NEURAIS ARTIFICIAIS

O neurônio é a célula cerebral cuja função principal é a coleta, processamento e disseminação de sinais elétricos. Pode-se supor que a capacidade que o cérebro possui de processar informação advém de redes de neurônios interligados. Segundo Russell e Norvig (2003), com base nessa suposição, alguns dos trabalhos pioneiros da Inteligência Artificial tinham por objetivo criar *redes neurais* artificiais.

McCulloch e Pitts (1943) propuseram um modelo matemático do neurônio, segundo o qual o neurônio "dispara" um sinal de saída quando uma combinação de seus sinais de entrada excede um determinado limiar. O modelo está ilustrado na Figura 3.4.

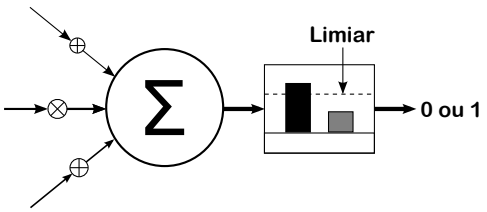


Figura 3.4: Modelo simples de um neurônio artificial

Segundo Fausett (1994), Redes Neurais Artificiais (RNAs) foram desenvolvidas como generalizações de modelos matemáticos de cognição humana, baseados nas seguintes premissas:

- O processamento de informação ocorre em muitas unidades simples chamadas *neurônios*;
- Sinais são passados entre neurônios através de conexões que os interligam;
- Cada sinapse tem um peso associado que, em uma rede neural típica, multiplica o sinal transmitido;
- Cada neurônio aplica uma função de ativação (geralmente não-linear) à sua entrada "líquida" (soma dos seus sinais de entrada multiplicados pelos respectivos pesos) para determinar seu sinal de saída.

As conexões entre os neurônios são chamadas de *sinapses* ou *pesos sinápticos*.

O autor também cita algumas características das RNAs que são sugeridas por neurônios biológicos:

- A memória é distribuída:
  1. A *memória de longa duração* reside nos pesos sinápticos;
  2. A *memória de curta duração* corresponde aos sinais transmitidos pelos neurônios.
- A intensidade de uma sinapse pode ser modificada pela experiência;
- Os neutransmissores para as sinapses podem ser excitatórios ou inibitórios.

Os neurônios são dispostos em *camadas*, conforme exemplificado na Figura 3.5. Através do uso de funções de ativação não-lineares, modificações nos pesos sinápticos e variações na quantidade de neurônios e número e arquitetura das camadas, a RNA torna-se capaz de resolver uma variedade de problemas.

Os neurônios da camada de entrada de uma RNA recebem estímulos do ambiente na forma de padrões de entrada. Os pesos sinápticos sofrem mudanças como resultado desses estímulos. Após essas mudanças, a rede neural pode responder de forma diferente a novos estímulos. O processo de aprendizado consiste no ajuste dos pesos sinápticos. Esse processo

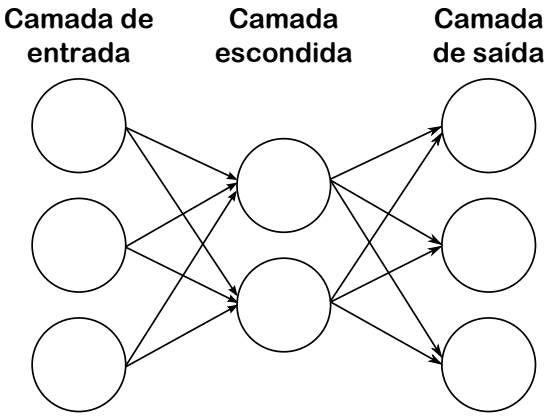


Figura 3.5: Exemplo de rede neural artificial simples

pode ser *supervisionado* (auxiliado por um professor) ou *não-supervisionado* (sem auxílio de um professor).

RNAs foram utilizadas com sucesso para resolver problemas em diversas áreas do conhecimento. Alguns desses problemas podem ser descritos de forma geral como problemas de *reconhecimento de padrões*, a exemplo de reconhecimento de letras e números escritos à mão. Um conjunto de padrões de treinamento, que codificam os caracteres, é apresentado à RNA para que esta possa aprendê-los; em seguida, são apresentados padrões novos à RNA, que deve ser capaz de classificá-los corretamente. É um exemplo de aprendizado supervisionado.

Agrupamento de dados, por sua vez, é um problema de aprendizado não-supervisionado, porque, conforme descrito na Seção 1.1.1, não se conta com nenhum tipo de auxílio externo para realizar a tarefa. Desta forma, as RNAs candidatas a resolver o problema de agrupamento de dados são aquelas que utilizam um processo de aprendizado não-supervisionado. Ao menos dois tipos de RNAs foram utilizadas para a tarefa de agrupamento de documentos, as RNAs SOM e ART1. As RNAs SOM são descritas na Seção 3.4.1, e as RNAs ART são abordadas na Seção 3.4.2.

3.4.1 MAPAS DE KOHONEN

As RNAs SOM (*Self-Organizing Map*) ou Mapa Auto-Organizável foram descritas por Kohonen (1981) *apud* Beebe (2007), e por isso há textos da literatura que as chamam de *Mapas de Kohonen*. Utilizam um processo de aprendizado não-supervisionado e produzem uma representação de baixa dimensionalidade (frequentemente bidimensional) dos padrões

de entrada – um *mapa*. Isto as torna úteis para produzir visualizações de baixa dimensionalidade a partir de conjuntos de dados de alta dimensionalidade. As dimensões do mapa de saída são um parâmetro do algoritmo, e devem ser fornecidas antes da execução. Na prática, as dimensões do mapa determinam o número de grupos no qual os padrões devem ser agrupados, e são comparáveis ao parâmetro k utilizado no algoritmo k-médias. Um esquema simplificado de uma RNA SOM é ilustrado na Figura 3.6.

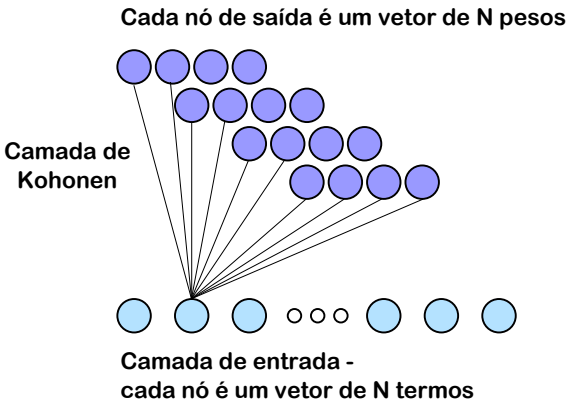


Figura 3.6: Esquema simplificado de RNA SOM, adaptado de Roussinov e Chen (1998)

As RNAs SOM foram aplicadas a diversos problemas, tais como reconhecimento da fala e compressão de imagens. Também foram utilizadas por Roussinov e Chen (1998) para agrupamento de documentos (páginas da WWW), que foram representados em um mapa bidimensional conforme ilustrado na Figura 3.7.

A próxima Seção discute a Teoria da Ressonância Adaptativa (*Adaptive Resonance Theory* (ART)).

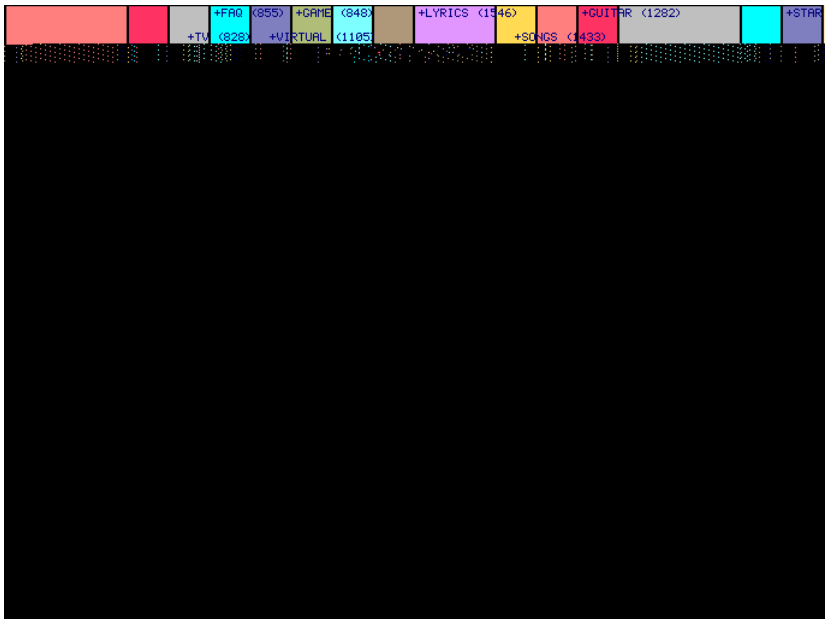


Figura 3.7: Mapa de 10.000 páginas de Internet (Roussinov e Chen, 1998)

3.4.2 TEORIA DA RESSONÂNCIA ADAPTATIVA

Os princípios básicos da Teoria da Ressonância Adaptativa (ART), inspirada no processo cognitivo humano, foram propostos por Grossberg (1976). A teoria descreve várias arquiteturas de RNAs auto-organizáveis que utilizam aprendizado não-supervisionado, e uma de suas principais metas de projeto foi superar o dilema da estabilidadade/plasticidade.

O dilema da estabilidadade/plasticidade pode ser descrito da seguinte forma: é desejável que um sistema seja capaz de aprender novos padrões sem esquecer o que já aprendeu.

Vários outros modelos de redes neurais não apresentam plasticidade porque não podem aprender novos padrões depois que a rede já foi treinada; e também não são estáveis porque, mesmo que possam ser retreinadas do zero para processar novos padrões, ao fazê-lo elas esquecem rapidamente o conhecimento prévio.

RNAs ART, por sua vez, são "plásticas" porque podem aprender dinamicamente novos padrões mesmo depois que a rede já se estabilizou, e também são estáveis porque preservam o conhecimento sobre padrões de entrada passados conforme novos padrões são apresentados. Cada grupo corresponde a um neurônio de saída, que o algoritmo cria e atualiza conforme os padrões de entrada são processados, o que caracteriza a rede como auto-organizável. Estas propriedades tornam as RNAs ART adequadas para agrupamento incremental de dados.

Esta dissertação estuda agrupamento de documentos com redes neurais ART1, projetadas para trabalhar com vetores de entrada binários, conforme descrito na Seção 3.5.

3.5 RNAs ART1

Dentre as várias arquiteturas de RNA descritas pela Teoria da Ressonância Adaptativa, as RNAs ART1 são aquelas projetadas para realizar tarefas de agrupamento de dados representados por vetores de entrada binários. Em termos formais, as redes ART1 são descritas por um sistema de equações diferenciais ordinárias (Carpenter e Grossberg, 1987b). Esse modelo, contudo, admite simplificações que permitem sua implementação computacional através de um algoritmo sequencial. Um algoritmo citado frequentemente na literatura é o de Moore (1988).

A organização básica de uma RNA ART1 é ilustrada na Figura 3.8, que apresenta dois componentes principais, os subsistemas *atencional* e de *orientação*. Os componentes principais do subsistema atencional são as duas camadas de neurônios F_1 e F_2. A *camada de comparação* F_1 possui N neurônios de entrada, e a *camada de reconhecimento* F_2 possui R neurônios de saída. N é o tamanho da entrada, isto é, a quantidade de padrões de entrada (neste trabalho, documentos) que são apresentados à rede; R é o número de grupos produzidos pelo algoritmo, que é calculado dinamicamente.

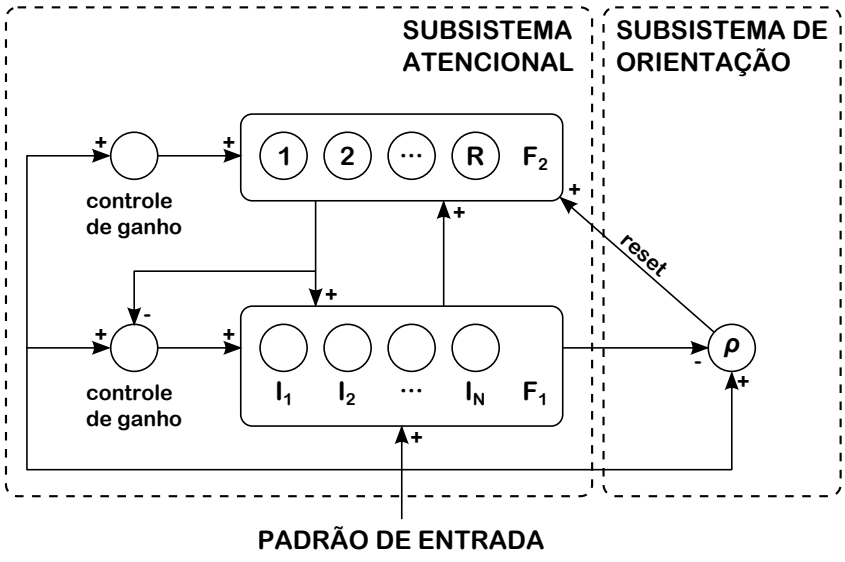


Figura 3.8: Organização básica de uma RNA ART1 (Carpenter e Grossberg, 1987b)

Os padrões de entrada $I_{1...N}$ são representados por vetores $I_j = \{0,1\}^M$ de dimensionalidade M, isto é, que possuem M componentes. Os vetores $I_{1...N}$ são apresentados à camada F_1. Os grupos correspondentes aos R neurônios da camada F_2 são representados por vetores $T_k = \{0,1\}^M$, chamados *protótipos*.

Os neurônios das duas camadas são completamente conectados entre si e possuem um fluxo bidirecional de informações. Há pesos sinápticos *bottom-up* (de F_1 para F_2), e *top-down* (de F_2 para F_1). Os pesos sinápticos *bottom-up* representam a memória de curta duração, isto é, as componentes do vetor que representa cada padrão de entrada, que mudam a cada novo padrão; os pesos sinápticos *top-down* representam a memória de longa duração, isto é, as componentes dos vetores que representam os protótipos dos grupos.

Após a apresentação do vetor de entrada I, os neurônios de saída (grupos) na camada F_2 competem para determinar qual deles será ativado, isto é, a qual deles o padrão de entrada poderá ser alocado. O neurônio com ativação máxima recebe o valor 1 e inibe os demais, que recebem o valor 0. Se o vetor I apresentado à camada F_1 e o protótipo T do neurônio vencedor da camada F_2 apresentam correspondência, isto é, possuem um número suficiente de dimensões em comum, diz-se que ocorreu *ressonância*; caso contrário, diz-se que ocorreu *reset*, o protótipo é descartado e reinicia-se a busca por um outro protótipo. Caso não seja encontrado um protótipo T que apresente correspondência com I, um novo grupo é criado, cujo protótipo tem seu vetor baseado no padrão de entrada.

Diz-se que os vetores I de um padrão de entrada e T do protótipo de um grupo apresentam correspondência quando o *teste de vigilância* é bem-sucedido. A vigilância (ρ) é um parâmetro adimensional que determina a fração das componentes de I valoradas como 1 que também devem existir em T.

Segundo Massey (2005a), as RNAs ART1 são sensíveis à ordem de apresentação dos padrões de entrada, e podem produzir soluções de agrupamento diferentes quando a ordem de apresentação dos padrões é modificada. Como os protótipos dos grupos são criados a partir dos padrões de entrada, uma mudança na ordem de apresentação dos padrões pode resultar na criação de protótipos diferentes.

A exposição acima não é uma descrição exaustiva dos mecanismos de funcionamento das RNAs ART1. Isto posto, foram apresentados todos os elementos necessários para a compreensão do algoritmo que permite implementar em *software* a arquitetura da rede, discutido na Seção 3.5.1.

3.5.1 ALGORITMO ART1

O algoritmo ART1 usado neste trabalho, descrito por Moore (1988), é apresentado no Algoritmo 1, e será chamado de "algoritmo ART1" deste ponto em diante. As definições e os parâmetros utilizados no algoritmo são descritos em seguida.

0. Start with zero cluster prototype vectors: the set P of prototype vectors is $\{\}$.
1. Let I = next input vector. Let P', the set of candidate prototype vectors, be equal to P.
2. Find the closest cluster prototype vector (if any). Call this cluster vector T. To find T is to find $i \in P'$ to maximize

$$\frac{T_i \cdot I}{\beta + ||T_i||_1}$$

for some $\beta \ll 1$.

3. If $P' = \{\}$, or if T_i is too far from I:

$$\frac{T_i \cdot I}{\beta + ||T_i||_1} < \frac{I \cdot I}{\beta + n},$$

then create a new cluster, j, and set $T_j = I$. Set $P = P \cup \{j\}$. Output j. Then go to step 1.

3'. If T_i does not match I, in other words,

$$\frac{T_i \cdot I}{I \cdot I} < \rho,$$

where $0 < \rho \leq 1$, then set $P' = P' - \{i\}$ and go to step 2.

4. Otherwise (T_i is close enough to I in both senses), update T_i: $T_i \leftarrow T_i \cap I$. Output i. Go to step 1.

Algoritmo 1: Algoritmo *Cluster-ART-I* descrito por Moore (1988)

O algoritmo ART1 agrupa vetores de entrada binários. Cada grupo tem um protótipo na forma de um vetor, e cada vetor de entrada é associado ao grupo cujo protótipo esteja mais próximo do vetor. O algoritmo cria tantos grupos quanto os dados exigirem, sendo que o número e o tamanho dos grupos (cardinalidade) dependem de dois parâmetros, β e ρ:

- β é o *parâmetro de escolha*, um pequeno número positivo que é somado ao denominador no passo 2 do algoritmo para evitar divisão por zero caso $||T_i||_1 = 0$. Carpenter et al. (1991) dizem que o limite $\beta \to 0$ é chamado de *limite conservador* (*conservative limit*) porque valores pequenos de β tendem a minimizar a recodificação, isto é, a atualização dos protótipos durante o aprendizado. β é usado na *função de escolha de categoria* descrita no passo 2 e também no teste de distância no passo 3 do algoritmo.

- ρ é o parâmetro de *vigilância*, $0 < \rho \leq 1$, que testa a similaridade entre o vetor de entrada e o protótipo no passo $3'$ do algoritmo e influencia diretamente o número de grupos criados, bem como a cardinalidade deles.

  Baixa vigilância leva a generalizações pouco refinadas e protótipos abstratos que representam muitos vetores de entrada, o que resulta em um número menor de grupos onde cada um contém um maior número de vetores de entrada. Alta vigilância leva a generalizações mais refinadas e protótipos que contêm um menor número de vetores de entrada, o que resulta em um número maior de grupos onde cada um contém um menor número de vetores de entrada.

Embora o algoritmo ART1 seja não-supervisionado, o usuário pode exercer uma pequena supervisão através do ajuste do parâmetro de vigilância. Esse ajuste é útil quando, finda a execução do algoritmo, a quantidade e a cardinalidade dos grupos são consideradas muito pequenas ou muito grandes.

Carpenter et al. (1991) dizem que tanto Fuzzy ART (outra arquitetura de RNA da família ART) quanto ART1 recebem três parâmetros: (*i*) um parâmetro de escolha α; (*ii*) uma taxa de aprendizado β, $0 < \beta \leq 1$; e (*iii*) a vigilância ρ. Moore (1988) cita apenas dois, o parâmetro de escolha β e a vigilância ρ. Embora os dois trabalhos descrevam conjuntos diferentes de parâmetros, trata-se do mesmo algoritmo. Isso porque o algoritmo ART1 descrito por Moore (1988) usa o símbolo β para o parâmetro de escolha no lugar do símbolo α citado por Carpenter et al. (1991) e implementa o chamado modo de *aprendizado rápido* (*fast learning*) descrito por Carpenter e Grossberg (1987b), onde se define a taxa de aprendizado $\beta = 1$.

Seguem as definições utilizadas pelo algoritmo:

- I – um vetor de entrada;

- P – o conjunto de protótipos de grupos, cujos elementos são vetores com a mesma dimensionalidade da coleção;

- P' – o conjunto de protótipos "candidatos" que concorrem para que um seja escolhido como o mais próximo de um vetor de entrada I;
- T – o protótipo vencedor, ou seja, o que está mais próximo de I. O protótipo também é chamado de *centróide* do grupo;
- $||u||$ = magnitude de um dado vetor u = número de componentes de u valoradas como 1;
- $u \cdot u$ é o produto escalar dos vetores u e v = número de componentes valoradas como 1 que u e v têm em comum;
- $u \cap v$ é vetor resultante da operação AND *bitwise* (*bit* a *bit*) dos vetores u e v;
- n – o número de dimensões de I.

Uma vez definidos todos os parâmetros do algoritmo ART1 definido por Moore (1988), o Algoritmo 2 apresenta a versão utilizada neste trabalho, a qual é adaptada do algoritmo ART1 original apresentado como Algoritmo 1.

Deve ser notado que o Algoritmo 2 é o mesmo algoritmo que o apresentado como Algoritmo 1, no entanto foi utilizada uma outra estrutura de dados que não um *array* para acelerar o processamento do agrupamento, uma vez que documentos representados sob a

forma de vetores tendem a ser esparsos. A motivação da escolha dessa estrutura de dados e sua lógica de funcionamento são descritas na Seção 4.3.

0. Comece com zero vetores de protótipos de grupos: $P = \{\}$.

1. Seja I = próximo vetor de entrada. Defina $P' = P$.

2. Encontre o protótipo T mais próximo de I (se houver). Defina $I = T$. Encontrar T equivale a encontrar o $i \in P'$ que maximiza

$$\frac{T_i \cdot I}{\beta + ||T_i||_1}$$

para algum $\beta \ll 1$.

$||T_i||_1$ conta os 1s em T_i e funciona como mecanismo de desempate (*tie-breaker*), favorecendo protótipos de menor magnitude em detrimento de outros de maior magnitude quando os primeiros são subconjuntos dos segundos (possuem 1s nas mesmas componentes) e I possui correspondência com ambos.

3. Se $P' = \{\}$, ou se T_i está muito distante de I:

$$\frac{T_i \cdot I}{\beta + ||T_i||_1} < \frac{I \cdot I}{\beta + n},$$

então crie um novo grupo, j, e defina $T_j = I$. Defina $P = P \cup \{j\}$. Imprima j. Vá para o passo 1.

3'. Se T_i não possui correspondência com I, ou seja,

$$\frac{T_i \cdot I}{I \cdot I} < \rho,$$

onde $0 < \rho \leq 1$, então defina $P' = P' - \{i\}$ e vá para o passo 2.

4. Caso contrário (T_i é próximo o suficiente de I em ambos os sentidos descritos nos passos 3 e 3'), atualize T_i: $T_i \leftarrow T_i \cap I$. Imprima i. Vá para o passo 1.

Algoritmo 2: Algoritmo ART1

Moore (1988) descreveu o algoritmo ART1 e relatou experimentações, porém não disponibilizou uma implementação do algoritmo na forma de um programa de computador. No decorrer da realização da pesquisa, foi localizada a implementação de Hudík (2011), escrita em linguagem C++. A análise do código-fonte disponibilizado por Hudík (2011)

evidenciou que, embora não haja citação explícita, trata-se de uma implementação do algoritmo ART1 descrito por Moore (1988). Dessa forma, a implementação de Hudík (2011) foi utilizada como base para o desenvolvimento do protótipo descrito no Capítulo 4.

Neste Capítulo foram apresentados os fundamentos teóricos que embasam esta pesquisa. O Capítulo 4 apresenta a proposta deste trabalho.

CAPÍTULO 4

PROPOSTA DO TRABALHO

Neste capítulo é apresentada em detalhes a proposta deste trabalho. A arquitetura definida para o protótipo da aplicação é apresentada na Seção 4.1. O módulo de indexação é descrito na Seção 4.1.1. Os módulos de busca e agrupamento são discutidos na Seção 4.1.2. A Seção 4.2 descreve os aspectos técnicos do protótipo. A Seção 4.3 apresenta a adaptação do algoritmo ART1 para reduzir seu tempo de execução no domínio de aplicação da Informática Forense, o que considera-se uma das contribuições desta pesquisa. A Seção 4.4 explica o que são e como são construídas as partições de referência, e como elas serão comparadas aos grupos obtidos pela aplicação. A Seção 4.5 discorre sobre as coleções que serão usadas nos experimentos do Capítulo 5. Por fim, na Seção 4.6, trabalhos correlatos são discutidos e comparados com a proposta desta dissertação.

4.1 ARQUITETURA DO PROTÓTIPO

O método de agrupamento de *hits* proposto foi materializado através de um par de programas desenvolvidos em linguagem Java, que juntos formam a aplicação denominada *Agrupador de Resultados de Buscas Textuais Forenses* – ARBTF. O escopo do ARBTF consiste unicamente em implementar o agrupamento de *hits* de conteúdo assemelhado e assim permitir a investigação do método proposto; seu propósito não é oferecer todas as funcionalidades das ferramentas forenses descritas na Seção 2.6, e sim fazer um complemento àquelas.

A arquitetura do protótipo é ilustrada na Figura 4.1, sendo que os módulos e suas respectivas interações serão detalhados nas próximas Seções. Há dois módulos: o de indexação (1) e o de agrupamento (2).

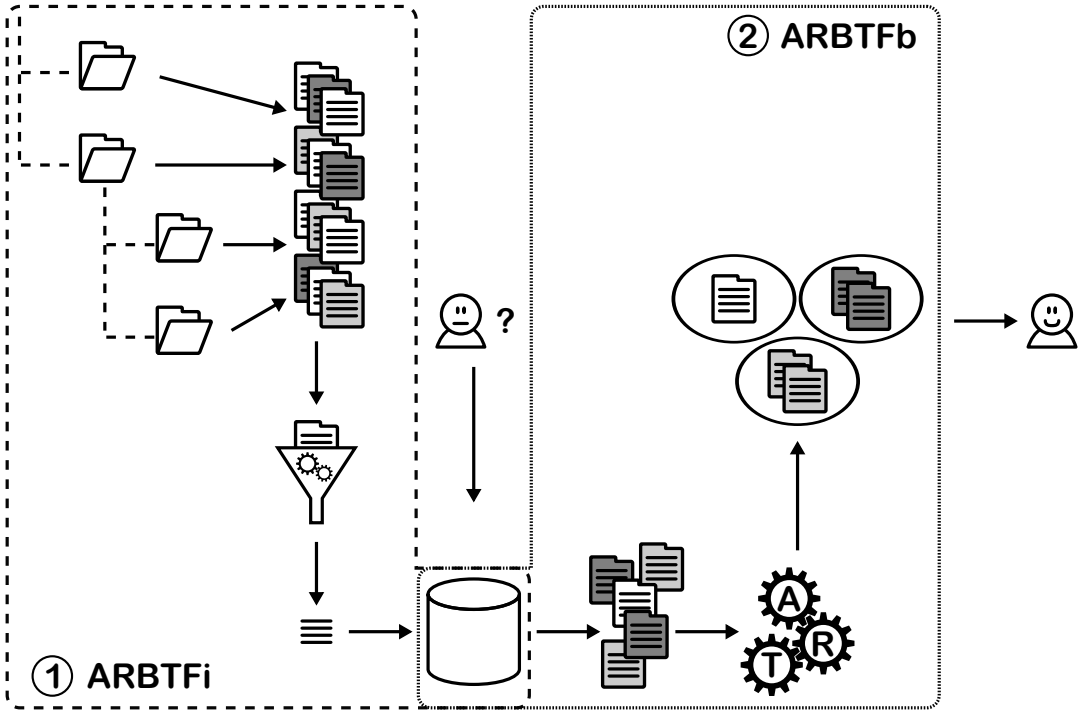


Figura 4.1: Arquitetura do ARBTF

4.1.1 MÓDULO DE INDEXAÇÃO

O módulo de indexação – ARBTFi foi definido a partir de duas bibliotecas da Apache Software Foundation (ASF). A biblioteca Tika (Mattmann e Zitting, 2011) extrai o texto dos diversos formatos de documentos, e a biblioteca Lucene (McCandless et al., 2010) é responsável pela indexação do texto.

O módulo ARBTFi recebe dois parâmetros textuais, sendo que o primeiro identifica o caso e o segundo indica um caminho do sistema de arquivos, que pode ser a raiz de uma unidade de disco ou uma pasta. O caminho é percorrido recursivamente e os seguintes tipos de arquivos são processados:

1. Arquivos de usuário do pacote Microsoft Office (.doc, .docx, .xls, .xlsx, ppt, .pptx, .pps e .ppsx);

2. Arquivos de texto plano (.txt);

3. Arquivos de texto estruturado (.pdf e .rtf);

4. Arquivos HTML e XML (.htm, .html e .xml).

Conforme citado na Seção 1.4, não são processados arquivos apagados, cifrados, esteganografados, com extensão incorreta, comprimidos, danificados ou incompletos.

O fluxograma do processamento de um arquivo é apresentado na Figura 4.2.

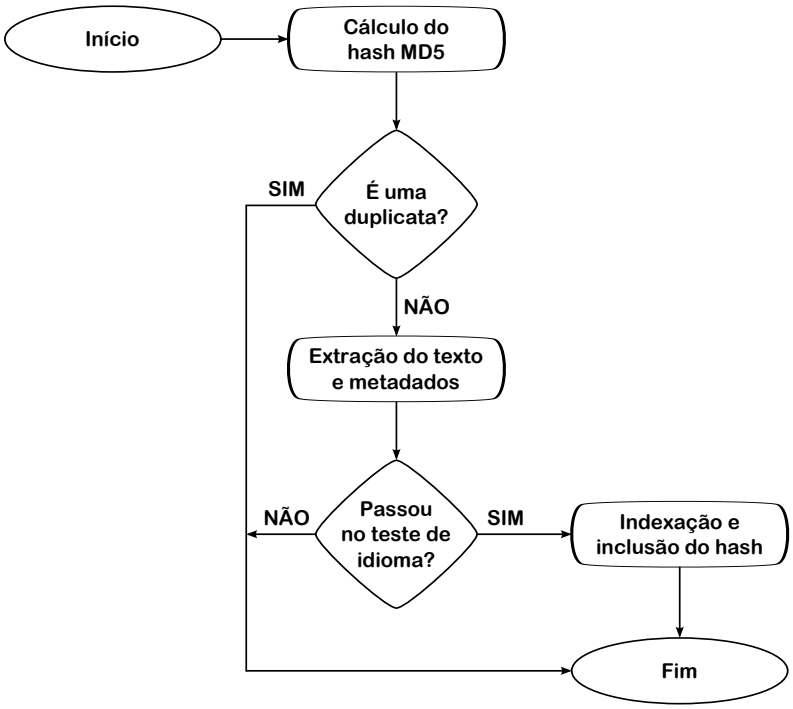


Figura 4.2: Fluxograma do processamento de um arquivo pelo ARBTFi.

O programa calcula o *hash* MD5[1], que é comparado aos *hashes* dos arquivos que já foram processados. Caso já tenha sido processado outro arquivo com o mesmo *hash*, é possível afirmar com altíssima probabilidade que se trata de uma duplicata, e o arquivo não é mais processado.

É importante ressaltar que arquivos duplicados não devem ser considerados inúteis nem descartáveis do ponto de vista forense. Um mesmo documento forjado utilizado para concorrer a dois processos licitatórios diferentes pode ser um indicador de reincidência e/ou má-fé do investigado; o perito deve relatar a existência desse tipo de duplicata, bem como o contexto em que foi encontrada, em seu laudo pericial. Optou-se por excluir

[1] `http://tools.ietf.org/html/rfc1321`, acesso em 29/10/2011.

os arquivos idênticos do processamento porque tenderiam a pertencer ao mesmo grupo e aumentariam o valor da estimativa $\phi^{(NMI)}$ sem necessariamente produzir soluções de agrupamento melhores. A utilidade do algoritmo é demonstrada quando este consegue agrupar adequadamente documentos cujos conteúdos sejam assemelhados no sentido de que possuem termos em comum, mas que não são idênticos.

Caso o arquivo não seja uma duplicata, seu texto e seus metadados[2] são extraídos. Os metadados e o texto extraídos são concatenados em uma cadeia textual, que será submetida ao teste de idioma. É importante ressaltar que os metadados selecionados para concatenação com o texto extraído são somente aqueles armazenados no próprio documento, que em geral são opcionais e podem ser modificados pelo usuário, variam conforme o tipo de arquivo, e podem ou não existir. Arquivos de texto plano (.txt) não armazenam metadados em seu interior; já os arquivos do Microsoft Office podem conter muitos metadados (autor, organização e assunto, entre outros). Os metadados armazenados no próprio documento não devem ser confundidos com aqueles armazenados no sistema de arquivos, tais como o nome do arquivo e suas datas de criação e modificação.

O módulo então submete a cadeia textual a um teste de idioma, cujo passo inicial é executar o algoritmo de detecção de idioma da biblioteca *language-detection*[3]. O algoritmo é probabilístico, e se baseia em perfis linguísticos gerados a partir de textos sabidamente escritos em uma lista de idiomas previamente elaborada. O algoritmo gera como saída uma lista de idiomas que foram detectados na cadeia textual e suas respectivas probabilidades de que foram identificados corretamente, após comparação com os dados armazenados nos perfis linguísticos. A taxa de acerto do algoritmo depende da quantidade de texto extraído; quanto mais texto, maior será a probabilidade de que o algoritmo conseguirá identificar corretamente o idioma. Passam no teste de idioma os arquivos em cuja cadeia textual o algoritmo detectar alguma das seguintes situações:

- O português é ao menos um dos idiomas detectados na cadeia textual;
- O algoritmo não consegue identificar nenhum idioma em nenhum trecho da cadeia textual. Esse resultado é possível porque o autor da biblioteca afirma que o modelo utilizado pelo algoritmo não funciona bem com textos curtos[4], e requer textos que contenham um mínimo de 10 a 20 palavras[5];

[2]Dados sobre os dados de um objeto.

[3]`http://code.google.com/p/language-detection/`, acesso em 12/10/2011.

[4]`http://code.google.com/p/language-detection/issues/detail?id=8`, acesso em 12/10/2011.

[5]`http://code.google.com/p/language-detection/wiki/FrequentlyAskedQuestion`, acesso em 12/10/2011.

- A cadeia textual contém texto em algum dos idiomas que não o português que o algoritmo é capaz de detectar, porém possui menos de 150 caracteres de extensão (número arbitrado após testes empíricos).

O propósito das duas últimas condições é não descartar arquivos com textos curtos cujo conteúdo consista basicamente de termos que não são palavras de dicionário, a exemplo de siglas, gírias, nomes próprios e senhas alfanuméricas. É frequente que o requisitante do exame pericial requisite a localização de arquivos que contenham determinados endereços de correio eletrônico, apelidos de pessoas investigadas e nomes de organizações que consistem de sequências de letras e números que não formam palavras de dicionário. Caso ocorrências dessas palavras-chave existam em arquivos que contenham pouco texto (por exemplo, um arquivo de texto plano que contém apenas um endereço de correio eletrônico e uma senha, ou um arquivo do Microsoft Word que contém apenas um apelido e uma imagem digitalizada), deixar de indexá-los porque o algoritmo não conseguiu identificar seu idioma terá o efeito de diminuir o *recall* das buscas, o que é indesejável, conforme exposto na Seção 1.1.

Se o arquivo passa no teste de idioma, a cadeia textual formada a partir da concatenação dos metadados e do texto extraídos é salva em um arquivo à parte, de extensão .txt, para permitir visualização rápida de seu conteúdo após a execução do módulo de busca. Através de recursos nativos da biblioteca Lucene, a cadeia textual é submetida às tarefas de pré-processamento descritas na Seção 3.2.1, quais sejam, análise léxica, eliminação de *stopwords*[6] e *stemming*. A cadeia textual pré-processada e o caminho completo do arquivo são armazenados em campos separados de um índice textual Lucene. Por fim, o *hash* do arquivo é adicionado à lista de *hashes* dos arquivos que já foram processados.

Concluído o processo de indexação de todos os arquivos, o índice pode ser consultado através da interface de programação disponibilizada pela biblioteca Lucene, conforme será descrito na Seção 4.1.2.

4.1.2 MÓDULOS DE BUSCA E AGRUPAMENTO

O módulo de busca e agrupamento – ARBTFb foi definido a partir das bibliotecas Lucene (McCandless et al., 2010), que é responsável por recuperar do índice textual os documentos

[6]Lista de *stopwords* em língua portuguesa gentilmente cedida pela Prof[a] Dr[a] Maria das Graças Volpe Nunes (`http://www.icmc.usp.br/~gracan/`) do Núcleo Interinstitucional de Linguística Computacional – NILC (`http://www.nilc.icmc.usp.br/nilc/`).

que contêm o termo de busca, e UJMP (*Universal Java Matrix Package*) (Arndt, 2011), que implementa estruturas de dados de matrizes e vetores esparsos.

Os parâmetros de tempo de execução do módulo ARBTFb são:

- Identificador do caso – obrigatório. Aponta qual índice Lucene deverá ser acessado pelo programa.
- Termo de busca – obrigatório. Define uma consulta em sintaxe Lucene[7].
- Vigilância – opcional. Valor da vigilância (ρ), $0 < \rho \leq 1$.
- Categoria – opcional. Caso informada, o ARBTFb somente buscará e agrupará os documentos que pertencerem a essa categoria. As categorias disponíveis são:

  1. *textoe* – arquivos de texto estruturado em formato Microsoft Word (.doc e .docx), PDF (.pdf) e RTF (.rtf);
  2. *textop* – arquivos de texto plano (.txt);
  3. *planilha* – arquivos de planilhas eletrônicas em formato Microsoft Excel (.xls e .xlsx);
  4. *ml* – arquivos em linguagens de marcação (*markup languages*) HTML e XML (.htm, .html e .xml).

  Estas categorias foram definidas a partir das categorias exibidas no FTK (AccessData, 2011), conforme apresentado na Seção 2.6.1.

- Pasta raiz – opcional. Caso informada, o ARBTFb somente buscará e agrupará os documentos que estiverem sob a árvore cuja raiz é esta pasta.
- Exclusão de pastas de sistema – opcional. Caso este parâmetro seja fornecido, o ARBTF não retornará documentos que estejam armazenados sob as árvores cujas raízes são pastas de sistema do sistema operacional Microsoft Windows, quais sejam, "Arquivos de Programas" e "Windows". Os documentos presentes em todas as outras pastas e que satisfazem o termo de busca são retornados.

O fluxograma de processamento do ARBTFb é apresentado na Figura 4.3.

Embora os módulos de busca e agrupamento do ARBTFb façam parte do mesmo programa, eles são conceitualmente independentes e serão descritos em separado.

[7] `http://lucene.apache.org/java/3_3_0/queryparsersyntax.html`, acesso em 12/10/2011.

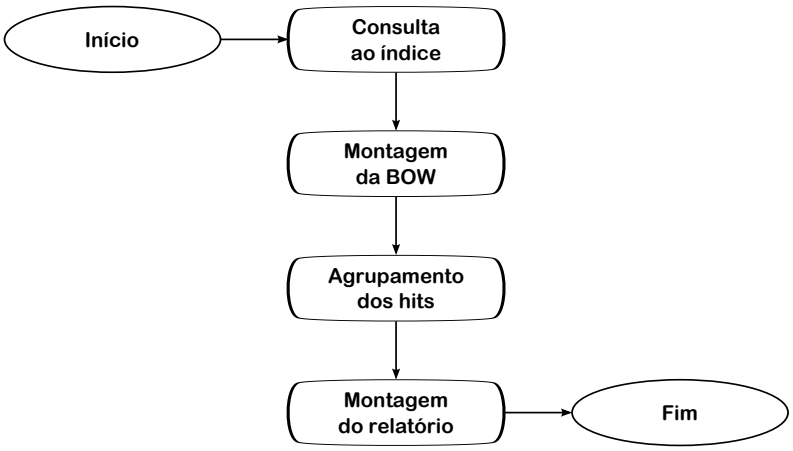


Figura 4.3: Fluxograma do processamento de uma busca pelo ARBTFb.

MÓDULO DE BUSCA

O módulo de busca é baseado na biblioteca Lucene e, portanto, recebe como parâmetro uma consulta em sintaxe Lucene, recuperando do índice todos os arquivos que satisfazem a consulta. O módulo então extrai do índice todos os termos contidos em cada arquivo recuperado e monta uma BOW (conforme exposto na Seção 3.2.2) onde estão representados todos os termos de todos os documentos que satisfizeram a consulta. Além disso, também é criada e alimentada uma estrutura de dados, denominadada *mapa de frequência*, que armazena a frequência absoluta de cada termo contido na BOW. A BOW e o mapa de frequência são então repassados para o módulo de agrupamento. Os documentos retornados pelo módulo de busca são incluídos na BOW pela ordem alfabética de seus nomes no sistema de arquivos. Esse é um aspecto importante porque, conforme citado na Seção 3.5, o algoritmo ART1 é sensível à ordem de apresentação dos vetores de entrada, e pode produzir soluções de agrupamento diferentes quando a ordem de apresentação dos vetores é modificada.

MÓDULO DE AGRUPAMENTO

Este módulo foi baseado no pacote ART para aprendizado de máquina não-supervisionado de autoria de Hudík (2011), escrito em linguagem C++, que foi utilizado nas experimentações relatadas por Hudík (2009). O algoritmo ART1 presente no pacote foi reescrito em linguagem Java e incluído no ARBTFb.

O algoritmo ART1 é executado em várias rodadas, com o fim de estabilizar os grupos. Considera-se que os grupos foram estabilizados quando alguma das seguintes condições se verifica:

- O programa atingiu a quantidade limite de rodadas, cujo valor padrão é de 100 rodadas;
- A *flutuação*[8] é inferior a um limiar, cujo valor padrão é de 2%;
- A quantidade de grupos não variou por 10 rodadas.

A lógica do módulo de agrupamento é apresentada no Algoritmo 3.

enquanto *(*rodada $<$ limite_rodadas*) E (*flutuacao $>$ limiar_flutuacao*) E (*rodadas_com_mesmo_num_grupos $<$ *9)* **faça**
    `// Agrupa os vetores de entrada e conta quantos vetores foram realocados de grupo.`
    Executar Algoritmo ART1;
    `// Cálculo de variáveis usadas na condição de parada do laço.`
    `// A flutuação é a porcentagem de vetores que foram incluídos em um grupo ou realocados de grupo.`
    flutuacao $\leftarrow$ (vetores_realocados$/N$) $\times$ 100
    `//` P `é a quantidade de protótipos da rodada atual,` $|P|_{ant}$ `é a quantidade de protótipos da rodada anterior.`
    se $|P| = |P|_{ant}$ **então**
        rodadas_com_mesmo_num_grupos $\leftarrow$ rodadas_com_mesmo_num_grupos $+1$
    senão
        rodadas_com_mesmo_num_grupos $\leftarrow 0$
    fim
    $|P|_{ant} \leftarrow |P|$
fim

Algoritmo 3: Módulo de agrupamento do ARBTFb

O módulo de agrupamento utiliza o algoritmo ART1 e recebe como parâmetro a BOW gerada pelo módulo de busca, a partir da qual cria uma estrutura de dados onde os documentos são representados na forma de vetores binários $d_i = [a_{i1}, \ldots, a_{iM}], d_i \in \{0,1\}^M$

[8]Porcentagem do número de vetores de entrada que ainda não estavam alocados a um grupo e foram alocados, ou que foram realocados para outro grupo.

onde a_{ik} é um dos atributos de d_i. A representação e a manipulação vetorial dos documentos são baseadas nas estruturas de dados e interfaces de programação da biblioteca UJMP (*Universal Java Matrix Package*) (Arndt, 2011). Os vetores são apresentados ao algoritmo ART1 e agrupados. Em seguida o módulo gera um relatório em formato HTML, exemplificado na Figura 4.4, onde são relacionados os grupos criados e seus respectivos documentos.

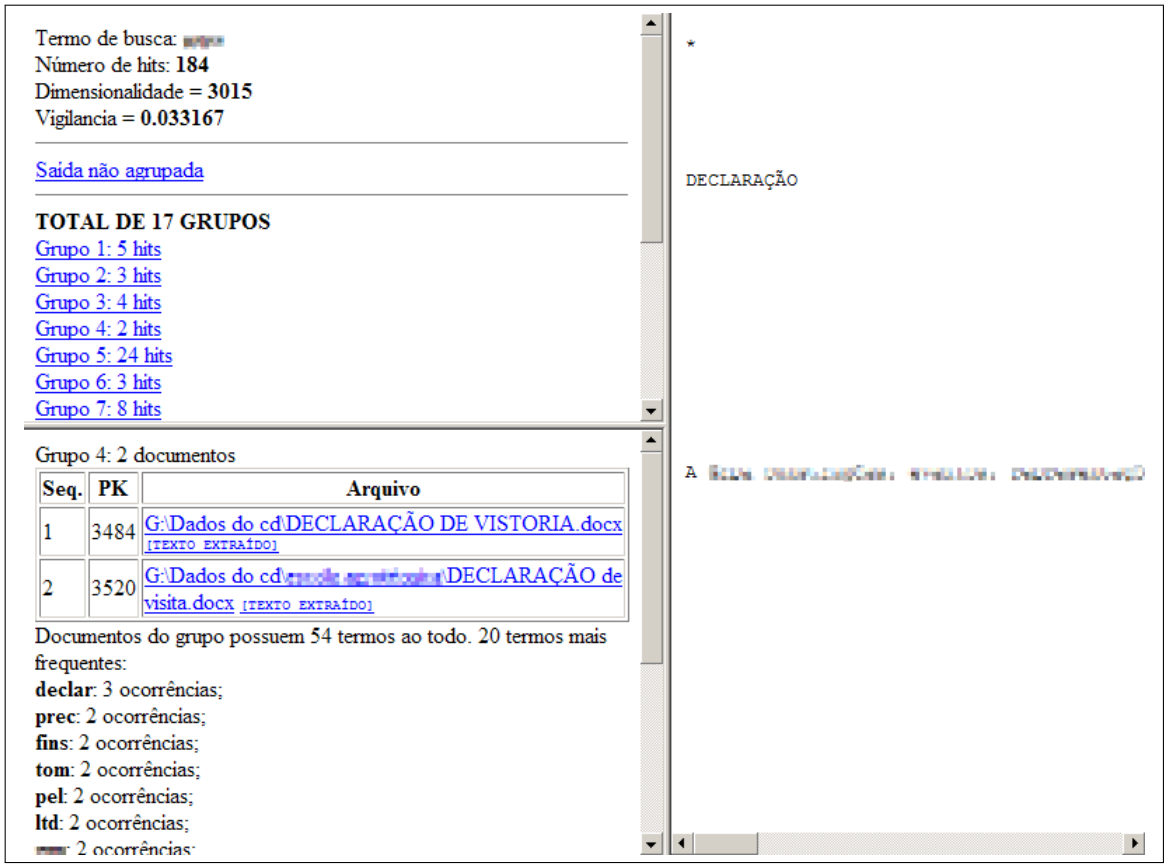


Figura 4.4: Relatório HTML gerado pelo módulo de agrupamento do ARBTFb.

O perito pode então navegar pelos grupos e visualizar o texto extraído dos documentos diretamente na tela do navegador, ou abrir o próprio documento; para que isso seja possível, o arquivo de imagem da mídia examinada deverá estar montado na raiz da mesma unidade de disco que foi utilizada durante a indexação. O relatório gerado possibilita o uso de uma abordagem de análise de dados exploratória, onde os dados são analisados para formular hipóteses, conforme descrito por Tukey (1980).

4.2 ASPECTOS TÉCNICOS

Os programas do ARBTF foram desenvolvidos em linguagem Java utilizando sistema operacional Windows XP de 32 *bits*. Todo o desenvolvimento e testes foram realizados em um computador equipado com um processador Intel Core 2 Duo T7600 de dois núcleos, fabricado em 2007, com frequência de operação de 2,33 GHz e 4 *megabytes* de memória *cache* L2, e 2 *gigabytes* de memória RAM. Durante os testes, a máquina virtual Java necessitou de parâmetros de linha de comando para aumentar os tamanhos inicial e final do *heap* para 512 e 1024 *megabytes* respectivamente. Esses podem ser considerados os requisitos de *hardware* para executar o ARBTF.

Como a linguagem Java é multiplataforma, teoricamente é possível executar os programas em qualquer sistema operacional que possua uma implementação da máquina virtual Java, a exemplo do Linux ou FreeBSD, porém esse cenário não foi testado. Todas as bibliotecas utilizadas, que foram citadas nas Seções 4.1.1 e 4.1.2, são gratuitas e de código aberto. Ambos os programas funcionam em interface de linha de comando. Os relatórios em formato HTML gerados pelo ARBTFb podem ser visualizados através de um navegador de Internet.

O módulo ARBTFi consiste de 1.087 linhas de código, sendo que a lógica de indexação consome 505 linhas e o resto do código consiste de classes e métodos utilitários. O módulo ARBTFb consiste de 1.503 linhas de código, sendo que a lógica de busca consome 481 linhas de código, a de agrupamento consome outras 368 e o resto do código consiste de classes e métodos utilitários.

O analisador léxico embutido da biblioteca Lucene para o idioma português do Brasil (*BrazilianAnalyzer*) teve de ser adaptado para o propósito deste trabalho, de forma que não fossem indexados valores numéricos nem termos com tamanho menor que 3 ou maior que 255 caracteres. O índice textual gerado pelo módulo pode ser lido por qualquer programa compatível com a biblioteca Lucene versão 3.3 ou superior, a exemplo do Luke[9], desde que as consultas sejam processadas pelo analisador léxico *BrazilianAnalyzer*.

[9] `http://code.google.com/p/luke/`, acesso em 13/10/2011.

4.3 REDUÇÃO DO TEMPO DE PROCESSAMENTO DO ALGORITMO ART1

O pacote de Hudík (2011) não foi projetado para processar padrões de entrada de alta dimensionalidade. Testes iniciais com documentos extraídos de coleções textuais forenses evidenciaram que o desempenho do programa era inadequado para o propósito desta pesquisa. Hudík (2009) alertou sobre essa inadequação (tradução livre):

> *Todos os algoritmos ART podem tratar apenas dados contínuos sem valores faltantes. Como muitos outros algoritmos de agrupamento, ele tem problemas se uma tarefa tem muitos atributos (maldição da dimensionalidade). Por essa razão ele não é o candidato ideal para tarefas como agrupamento de texto.*

Conforme aumentava a dimensionalidade do conjunto de documentos retornados, também aumentava proporcionalmente o tempo de processamento, que com uma BOW de 3 milhões de células chegou à casa dos minutos e assim diminuiu a aplicabilidade do uso interativo da técnica; pela experiência dos peritos, normalmente submetem-se dezenas de buscas textuais em cada exame pericial, e um tempo de resposta da ordem de minutos desencoraja o uso da técnica. Desta forma, com o objetivo de reduzir o tempo de processamento do algoritmo de agrupamento e assim viabilizar seu uso interativo, foi implementada a abordagem abaixo descrita.

Conforme descrito na Seção 3.2.2, a BOW é uma matriz de incidência $N \times M$ binária de N documentos e M termos. Se o documento d_i contém o termo t_j, diz-se que $a_{ij} = 1$; se não contém, diz-se que $a_{ij} = 0$. Mesmo uma coleção textual forense relativamente pequena, de 1.000 documentos contendo 1.000 termos diferentes, gera uma BOW com 1.000.000 (um milhão) de atributos. Na prática, frequentemente cada documento contém apenas uma pequena porcentagem dos termos da coleção, e a matriz de incidência apresenta característica *esparsa*. Stoer e Bulirsch (2002) trataram de problemas que apresentavam essa característica (tradução livre):

> *(...) nos quais a matriz A felizmente é* esparsa*, isto é, possui apenas relativamente poucos elementos não-zerados.*

É comum encontrar, em exames periciais, coleções textuais que contêm milhares de arquivos e dezenas de milhares de dimensões, e que ao serem representadas em uma BOW ocasionam o conhecido problema da maldição da dimensionalidade citado por Kriegel et al. (2009).

Cada termo presente em cada documento da coleção é representado por uma coluna na matriz de incidência binária. Os documentos frequentemente contêm apenas uma porcentagem relativamente pequena dos termos da coleção e isso se reflete em suas representações vetoriais, que contêm muitos atributos zerados. Esta característica é particularmente evidente em documentos cujo texto contém uma quantidade de termos pequena em relação ao número total de termos da coleção. A matriz é dita esparsa, porque a maior parte de seus atributos é zerada; os atributos não-zerados são minoria. Essa propriedade pode ser explorada para reduzir o tempo de execução do algoritmo de agrupamento.

A implementação original do algoritmo ART1 apresentada na Seção 3.5.1 percorre todas as dimensões de todos os vetores de entrada, estejam elas preenchidas ou não; em coleções textuais, cuja natureza comumente esparsa foi exposta acima, esta é uma abordagem subótima porque muitos ciclos de processamento e também espaço de armazenamento em memória principal são gastos com informações que não têm utilidade para o algoritmo. Um vetor de entrada com dimensionalidade 1.000 e que tenha apenas um atributo preenchido com o valor 1 demandará tantos ciclos de execução para ser processado quanto outro vetor que tenha todos os atributos preenchidos.

Desta forma, a contribuição específica desta parte do trabalho é substituir o vetor padrão por uma estrutura de dados que desconsidera dimensões cujo valor esteja zerado e, por esse motivo, permite iteração rápida de dimensões cujo valor seja diferente de zero, o que traz um ganho real de tempo ao percorrer uma matriz de incidência binária obtida a partir de uma coleção textual típica. Esta adaptação não reduz a complexidade computacional do algoritmo ART1 original porque no pior caso, onde todas as dimensões estão preenchidas, são gastos N passos para percorrer todas as dimensões; no melhor caso, entretanto, apenas 1 passo será gasto para percorrer um vetor que contenha apenas uma única dimensão preenchida, a quantidade mínima necessária de atributos preenchidos para que o vetor seja apresentado ao algoritmo de agrupamento no ARBTFb.

4.3.1 VETORES DE ENTRADA ESPARSOS

A análise do código-fonte em linguagem C++ do algoritmo ART1 disponibilizado por Hudík (2011) evidenciou que, embora não haja citação explícita, trata-se de uma implementação do algoritmo ART1 descrito por Moore (1988). A discussão que segue refere-se a essa implementação específica; até onde foi possível investigar, não foram localizadas outras implementações publicamente disponíveis.

A entrada do algoritmo é uma coleção de vetores binários $d_i = [a_{i1}, \ldots, a_{iM}], d_i \in \{0,1\}^M$ onde a_{ik} é um dos atributos de d_i. Um vetor pode ser implementado em C++ de forma simples e direta por meio do gabarito de classe *vector*, que é a abordagem utilizada por Hudík (2011) e se presta bem à tarefa de representar padrões de entrada quando estes não são esparsos; é inadequada, entretanto, para o propósito desta pesquisa porque os documentos que serão apresentados ao algoritmo ART1 têm característica esparsa.

Nos passos 2, 3, 3′ e 4 descritos no Algoritmo 1 da seção 3.5.1, o uso de *vector* faz o algoritmo iterar sobre todas as dimensões do vetor de entrada, do protótipo, ou de ambos. Não é necessário iterar sobre todas as dimensões, porque os cálculos somente levam em conta as dimensões cujo valor é 1 – que, conforme descrito na Seção 4.3, são relativamente poucas em coleções textuais típicas. Ao usar *vector* para representar os vetores envolvidos e iterar sobre todas as dimensões, o algoritmo gasta a maior parte de seu tempo de processamento de forma inútil e aloca memória desnecessariamente.

Para resolver esse problema, a proposta é substituir *vector* por uma estrutura de dados mais adequada à tarefa de agrupamento de documentos: um *vetor esparso*, que permita iteração rápida sobre as componentes cujo valor seja 1 e pule as componentes cujo valor seja 0. Ao utilizar essa estrutura de dados, o algoritmo será capaz de percorrer a matriz de incidência binária que representa os documentos muito mais rapidamente que com o gabarito de classe *vector*.

4.3.2 COMPLEXIDADE COMPUTACIONAL

Segundo Massey (2005a), o algoritmo ART1 original tem complexidade computacional de ordem $O(MN2^M)$ onde N é o número de documentos e M é o número de dimensões. Neste ponto também é necessário definir R como o número de grupos criados pelo algoritmo.

O fator MN representa o laço de estabilização, e N é $O(M)$. O fator 2^M vem do número máximo de protótipos de grupo que o algoritmo pode ter que percorrer até encontrar o protótipo que mais se assemelha ao vetor de entrada. Para vetores M-dimensionais, há um total de 2^M vetores de entrada possíveis, que no pior caso serão "agrupados" cada um em um grupo onde o protótipo e único membro é ele mesmo: $R = N = 2^M$.

O custo do algoritmo parece ser proibitivo porque M frequentemente alcança a casa dos milhares; no entanto, este tipo de cenário não é realista. Massey (2005a) afirma que R deve ser, no mínimo, algumas ordens de grandeza menor que N porque o propósito do agrupamento é comprimir o espaço de informação e facilitar sua exploração. Assim, em cenários realistas se verificaria que $R \ll N \ll M \ll 2^M$, número muito distante do pior

caso. Massey (2005a) menciona ainda que espera que R varie das dezenas às centenas, a depender da aplicação e do tamanho da coleção, e conclui afirmando que o algoritmo ART1 é $O(sRN)$, onde s é o número de iterações necessárias até a estabilização. Os experimentos conduzidos nesta pesquisa confirmam essa afirmação, conforme apresentado no Capítulo 5.

A complexidade da versão modificada do algoritmo ART1 (Algoritmo 2 apresentado na Seção 3.5.1) tem a mesma ordem $O(sRN)$ descrita, porém executa mais rapidamente que outras implementações porque pula as dimensões cujo valor é 0, conforme demonstrado na Seção 4.3.3.

4.3.3 IMPLEMENTAÇÃO

A implementação do algoritmo ART1 de autoria de Hudík (2011), escrita em linguagem C++, foi reescrita em linguagem Java, e o gabarito de classe *vector* foi substituído pela classe *DefaultSparseBooleanMatrix* da biblioteca UJMP (Arndt, 2011). Essa escolha foi motivada pelo fato que *DefaultSparseBooleanMatrix* é projetada para iterar rapidamente sobre as componentes cujo valor seja 1 e pular as componentes cujo valor seja 0. A nova estrutura de dados foi utilizada para representar os vetores de entrada e os protótipos dos grupos.

O programa *art_1.exe* foi compilado a partir do pacote ART de autoria de Hudík (2011) com o compilador *gcc* versão 4.4.1 em ambiente Windows XP de 32 *bits*. O programa *ART1.jar* foi construído a partir da reescrita, em linguagem Java, do pacote ART de autoria de Hudík (2011). O compilador utilizado foi o *javac* versão 1.6.0_26 de 32 *bits* distribuído no *Java Development Kit* (JDK) da Oracle Corporation. Os programas utilizam o Algoritmo 3 com os seguintes parâmetros:

- Vigilância (ρ) – $0,05$;
- Parâmetro de escolha (β) – $0,01$;
- Limite de rodadas – 100;
- Limiar de flutuação – 5%;
- Limite de rodadas com o mesmo número de grupos – 5.

Os testes foram executados com coleções de 200 a 2.000 documentos extraídos de coleções textuais forenses, com incrementos de 200 documentos. Os resultados experimentais, apre-

sentados na Tabela 4.1 e na Figura 4.5, evidenciaram uma redução substancial do tempo de processamento em relação ao programa escrito em C++.

A relação entre o aumento na quantidade de documentos e o tempo de execução do algoritmo não é linear. Uma possível explicação para esse fenômeno é que o tempo de execução é proporcional à quantidade de células da BOW que são percorridas por cada programa. O programa *art_1.exe* percorre todas as células da BOW. O programa *ART1.jar* percorre apenas as células que não estão zeradas. Esse é o cenário ilustrado pela Figura 4.6.

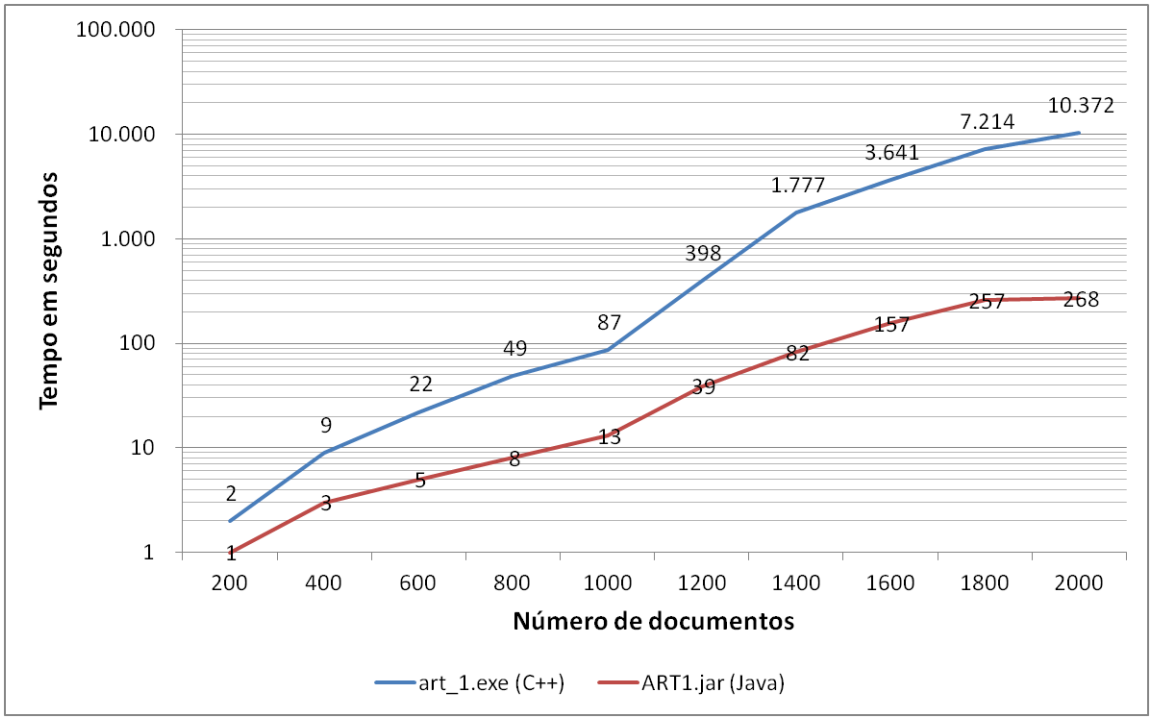


Figura 4.5: Tempos de execução em segundos (escala logarítmica) das implementações em C++ e em Java

As experimentações realizadas sugerem que a utilização interativa do protótipo pode ser viabilizada com o uso da classe *DefaultSparseBooleanMatrix* da biblioteca UJMP (Arndt, 2011).

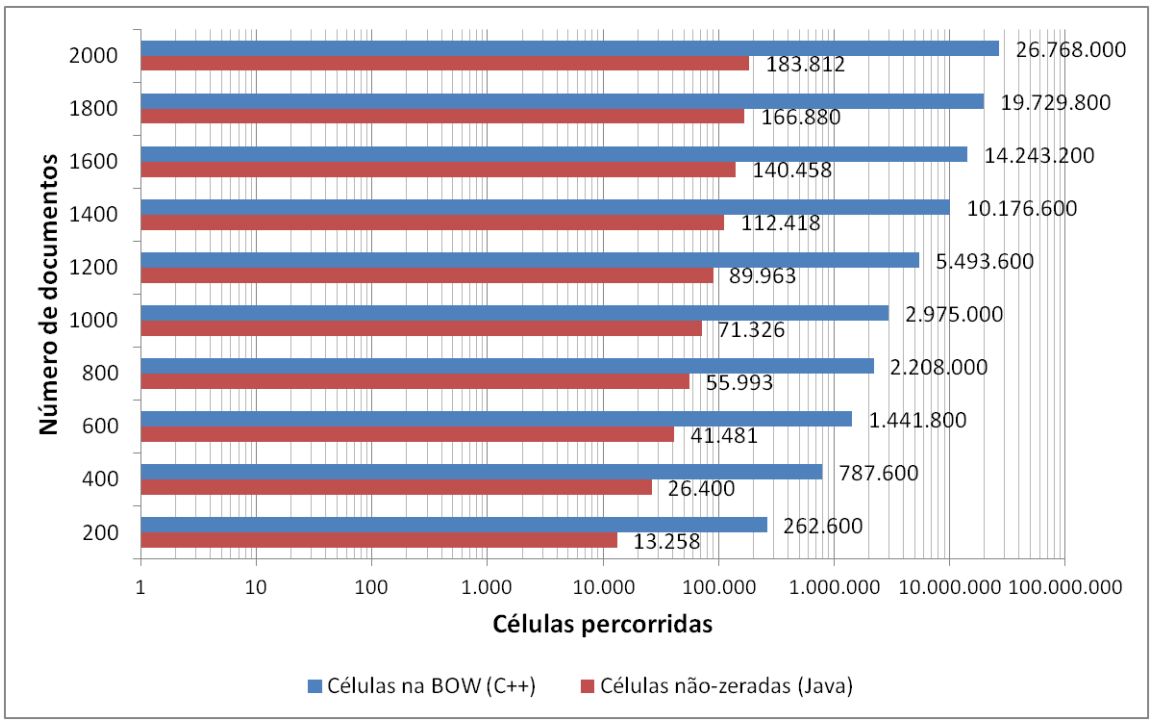


Figura 4.6: Quantidade de células percorridas (escala logarítmica) pelas implementações em C++ e em Java

Tabela 4.1: Comparação entre as implementações em C++ e em Java

N	200	400	600	800	1.000	1.200	1.400	1.600	1.800	2.000
art_1.exe (C++)	2 s	9 s	22 s	49 s	87 s	398 s	1.777 s	3.641 s	7.214 s	10.372 s
ART1.jar (Java)	1 s	3 s	5 s	8 s	13 s	39 s	82 s	157 s	257 s	268 s
Dimensões	1.313	1.969	2.403	2.760	2.975	4.578	7.269	8.902	10.961	13.384
Células na BOW	262.600	787.600	1.441.800	2.208.000	2.975.000	5.493.600	10.176.600	14.243.200	19.729.800	26.768.000
Células não-zeradas	13.258	26.400	41.481	55.993	71.326	89.963	112.418	140.458	166.880	183.812
Porcentagem não-zerada	5,05%	3,35%	2,88%	2,54%	2,40%	1,64%	1,10%	0,99%	0,85%	0,69%

4.4 PARTIÇÕES DE REFERÊNCIA

Para cada coleção foi elaborada uma lista de termos de busca. Foram realizadas buscas baseadas em cada termo, cujos resultados foram classificados manualmente de acordo com seu teor textual. Essas classes, que são consideradas corretas por terem sido obtidas a partir da classificação manual realizada por um especialista, serão as *partições de referência* com as quais serão comparados os agrupamentos obtidos pelo algoritmo. Por exemplo, o termo "ACME" pode ser usado para buscar documentos relacionados à empresa fictícia ACME S/A e retornar *hits* com diversos teores: contratos, notas fiscais, propostas comerciais e certidões, entre outros. O agrupamento obtido pelo algoritmo será então comparado às partições de referência (ou classes padrão) pré-estabelecidas por um especialista (*ground truth*), e a métrica de avaliação será a estimativa $\phi^{(NMI)}$, conforme citado na Seção 3.3.3.

As partições de referência podem ser consideradas o limite superior de qualidade dos agrupamentos obtidos através do algoritmo ART1. É razoável supor que a qualidade da solução obtida pelo agrupamento seja inferior àquela obtida pelo especialista, por não contar com o conhecimento do especialista; no entanto, o agrupamento é executado em menor tempo e com custo mais baixo, e pode ser útil em cenários onde existam restrições de tempo e recursos. Em um dos experimentos relatados no Capítulo 5, o protótipo agrupou 185 documentos em cerca de 4 segundos.

4.4.1 VALIDAÇÃO

Os grupos obtidos serão comparados às partições de referência (classes), e a estimativa $\phi^{(NMI)}$ será calculada.

Para alguns dos testes também será realizada uma análise qualitativa por Peritos Criminais Federais do Departamento de Polícia Federal, que avaliarão os grupos e dirão se são "bons"ou "ruins", e "úteis" ou "inúteis", bem como opinarão se preferem revisar os *hits* na forma de grupos ou de lista exaustiva. O que motiva essa avaliação é que um agrupamento cujo valor de NMI seja considerado baixo ainda pode ser um bom agrupamento, porque, conforme apresentado na Seção 1.1.1, podem existir várias formas corretas de dispor em grupos os arquivos de uma coleção textual, e nenhuma delas é necessariamente melhor que a outra. As citações que seguem são traduções livres. Segundo Han e Kamber (2006),

> *(...) o ruído de uma pessoa pode ser o sinal de outra pessoa.*

Jain (2010) afirma que

> *Na realidade, um grupo é uma entidade subjetiva que está no olho do observador e cuja importância e interpretação exigem conhecimento do domínio* [de aplicação]. (esclarecimento adicionado pelo tradutor)

Conforme o autor, o aspecto subjetivo também faz parte da avaliação de uma solução de agrupamento.

Por fim, é importante ressaltar o que dizem Aldenderfer e Blashfield (1984):

> *A estratégia de análise de agrupamento* busca *estrutura apesar de sua operação* impor *estrutura.* (grifo do tradutor)

Ou seja, embora o agrupamento seja uma técnica exploratória, os grupos obtidos seguem a estrutura determinada pelos algoritmos e parâmetros utilizados.

Sempre existe um critério subjetivo na avaliação das soluções de agrupamento produzidas por um algoritmo, critério esse relacionado ao objetivo do exame pericial e à interpretação do perito. Dessa forma, este trabalho utiliza, de forma complementar, os dois critérios discutidos:

- A estimativa $\phi^{(NMI)}$, que mede a correspondência entre a solução de agrupamento gerada pelo algoritmo ART1 e as classes padrão pré-estabelecidas por especialistas (*ground truth*), a qual será denominada *análise quantitativa*;
- A avaliação subjetiva realizada por peritos no exercício das atribuições periciais, a qual será denominada *análise qualitativa.*

Acredita-se que a combinação das duas análises seja adequada para validar a proposta.

4.5 COLEÇÕES

Serão utilizadas coleções textuais obtidas a partir de três mídias de armazenamento apreendidas durante o cumprimento de mandados de busca e apreensão pelo Departamento de Polícia Federal.

As duas primeiras mídias são relacionadas à investigação de supostas fraudes em licitações. Sedes de empresas e de entes da administração pública, a exemplo de prefeituras e órgãos públicos, são alvos comuns dos mandados de busca e apreensão dessa categoria de investigação. As mídias apreendidas frequentemente contêm muitos arquivos textuais de usuário porque são usados na rotina de trabalho da organização investigada. O requisitante

dos exames formulou uma série de quesitos e elencou uma lista de palavras-chaves composta de nomes de organizações e indivíduos. Os documentos que fizessem referência aos indivíduos e organizações investigados deveriam ser organizados e entregues juntamente com o laudo pericial, bem como deveriam ser apontados eventuais indícios de autoria, materialidade e dinâmica das condutas investigadas.

A terceira mídia é relacionada à investigação de suposta fraude em saques do Programa Bolsa-Família onde o investigado é um funcionário de um banco público. Foi apreendido o computador pessoal do investigado. O requisitante dos exames não formulou quesitos nem elencou palavras-chaves; apenas requisitou que fossem buscados indícios da prática do suposto crime e forneceu o nome completo do funcionário público.

Conforme citado na Seção 1.3, os dados reais dos casos, quais sejam, nomes de indivíduos e organizações, foram trocados por identificadores genéricos para não expor os investigados.

4.6 TRABALHOS CORRELATOS

Nesta Seção são apresentados dois trabalhos de agrupamento de documentos textuais forenses.

4.6.1 O TRABALHO DE BEEBE

Beebe (2007) tratou o problema de agrupamento de *hits* retornados por buscas em coleções textuais forenses. A técnica utilizada para o agrupamento foi a de RNAs SOM (Kohonen (1981) *apud* Beebe (2007)). A autora considerou o trabalho promissor, porém relatou uma série de problemas e limitações.

As buscas executadas por Beebe foram executadas no nível físico do arquivo de imagem da mídia examinada e abstraíram o sistema de arquivos. Há vários problemas com esta abordagem. O primeiro deles é que muitos formatos de documentos (Microsoft Office e PDF, dentre outros) não são armazenados em texto plano, mas em formatos binários proprietários, e necessitam ser submetidos a um processo de extração de texto (*parsing*) para que seu conteúdo possa ser processado. Embora não seja impossível reconstruir esses arquivos por um processo de *carving* para em seguida extrair seu conteúdo, fazê-lo a partir da estrutura oferecida pelo sistema de arquivos é mais simples e rápido.

O segundo problema é que os documentos contidos na mídia não necessariamente estão armazenados em regiões contíguas, e dessa forma a pesquisadora incorreu no risco de ter

processado apenas fragmentos de arquivos; em particular, palavras que correspondiam às buscas mas que se encontravam em fragmentos diferentes do arquivo, armazenados em regiões diferentes da mídia, não seriam localizadas. A justificativa para atuar no nível físico é que não é suficiente se ater ao sistema de arquivos em exames *postmortem* de Informática Forense porque muitos vestígios podem estar latentes no espaço livre e nas áreas não-alocadas da mídia de armazenamento; contudo, essa abordagem é questionável porque, embora seja intrinsecamente capaz de capturar vestígios que não se encontram no sistema de arquivos, é incapaz de capturar outros vestígios que se *encontram* no sistema de arquivos.

A proposta de processo de busca de Beebe (2007) não inclui uma etapa de indexação, e é elaborada uma lista de termos de busca que serão buscados no arquivo de imagem; os *hits* retornados por essas buscas são então examinados um a um, e ao fim desse processo o exame é considerado concluído. A decisão de não indexar o conteúdo do arquivo de imagem é discutível. Johansson (2003) demonstrou que a indexação torna muito mais rápidas as buscas textuais em perícias de Informática Forense. Mesmo que a indexação possa consumir tempo substancial, as buscas posteriores são executadas rapidamente, e os *hits* retornados podem servir de base para novas buscas, cenário compatível com o conceito de análise de dados exploratória, que propõe que os dados sejam analisados para então formular hipóteses (Tukey, 1980). Sem indexação, essas novas buscas seriam proibitivamente lentas em arquivos de imagem grandes, que são comuns em se tratando de discos rígidos; esse problema não ocorreria se fosse utilizada indexação, e seria possível aplicar princípios de análise de dados exploratória ao exame pericial.

Por fim, na proposta de Beebe (2007), o usuário necessitava informar, antes de submeter os termos de busca, o tamanho do mapa bidimensional que iria exibir os grupos obtidos. O trabalho não menciona técnicas que possam determinar ou mesmo sugerir esse valor, sendo esta definição feita de forma empírica e fixa, o que pode trazer problemas nos resultados alcançados pelas experimentações.

A proposta desta dissertação contrasta com o trabalho de Beebe (2007) nos pontos que seguem:

- Utiliza o sistema de arquivos, ao mesmo tempo que não atua no nível físico;
- Utiliza um indexador que processa documentos em formatos estruturados. Por outro lado, não processa o conteúdo da área livre nem da área não-alocada da mídia examinada, o que foi feito por Beebe (2007). A escolha de indexar documentos em formatos estruturados foi feita porque, em um grande número de casos, o maior entrave não

consiste em revelar eventuais documentos ocultos, e sim em organizar grandes quantidades de documentos encontrados nos sistemas de arquivos da mídia examinada. O índice textual construído torna mais rápidas as buscas;

- Oferece a opção de agrupar a coleção inteira, sem necessidade de especificar um termo de busca, o que pode ser útil para proporcionar uma visão geral do conteúdo da mídia examinada;
- Não requer que o usuário informe nenhum parâmetro além do termo de busca;
- Utiliza RNAs ART1, ao passo que Beebe (2007) utiliza RNAs SOM. Embora sejam RNAs muito diferentes, é possível identificar uma diferença básica e marcante entre elas, que é a das saídas produzidas. RNAs ART1 produzem como saída um número variável de grupos, cada um com sua respectiva lista de documentos; RNAs SOM produzem um mapa bidimensional de tamanho fixo.

4.6.2 O TRABALHO DE DECHERCHI ET AL.

Decherchi et al. (2009) realizaram um estudo onde foram agrupadas partes de uma coleção textual forense composta de mensagens de correio eletrônico. A coleção continha as caixas postais de 158 funcionários de nível gerencial da empresa norte-americana de energia Enron (Cohen, 2009). Foram escolhidos aleatoriamente 5 funcionários, que tiveram suas caixas postais submetidas a um processo de agrupamento. O algoritmo escolhido foi o k-médias, citado na Seção 3.3. Os autores do estudo escolheram $k = 10$, e justificaram que *"(...) esta escolha foi guiada pela necessidade prática de obter um número limitado de grupos informativos."*. Após formados os grupos, os termos considerados mais descritivos entre os 20 mais frequentes de cada grupo foram então selecionados, e a partir deles os autores buscaram interpretar os principais tópicos de cada grupo. Assim, é possível afirmar que os autores do estudo usaram o agrupamento como ferramenta para análise de dados exploratória (Tukey, 1980).

Os resultados foram considerados pelo autores como extremamente interessantes. Não são apresentados detalhes técnicos da implementação utilizada. É interessante notar que, enquanto Beebe (2007) agrupou *hits*, Decherchi et al. (2009) agruparam partes da coleção, partes essas que podiam ser consideradas coleções à parte. As duas abordagens podem ser vistas como técnicas exploratórias. O contraste entre elas é que o agrupamento de *hits*, que combina técnicas de aprendizado de máquina não-supervisionado e de Recuperação

da Informação, somente é capaz de produzir resultados tão bons quanto forem bons os termos de busca utilizados, de forma que o sucesso da investigação depende diretamente do conhecimento que o perito possui sobre o caso e de sua experiência anterior em casos semelhantes; o agrupamento de todos os documentos da coleção, por sua vez, oferece uma visão geral, mas em um nível de granularidade menos refinado que o da busca.

A proposta desta dissertação contrasta com o trabalho de Decherchi et al. (2009) nos pontos que seguem:

- Utiliza um algoritmo que determina por si só a quantidade de grupos, com base na estrutura interna dos dados apresentados. Isto é desejável para que não seja necessário arbitrar o número de grupos como fizeram Decherchi et al. (2009);

- Em seu modo básico de operação, agrupa os *hits* retornados por buscas por palavras-chave, ao passo que Decherchi et al. (2009) agruparam subcoleções da coleção principal.

Ambos os trabalhos de Beebe (2007) e Decherchi et al. (2009) exigem que o usuário informe, de uma forma ou de outra, o número de grupos nos quais os documentos devem ser alocados. Em contraste, a proposta deste trabalho é utilizar um algoritmo que determina por si só o número de grupos. As RNAs ART1 descritas pela Teoria da Ressonância Adaptativa (ART) apresentam as seguintes propriedades que as tornam particularmente aptas para o objetivo proposto neste trabalho, quais sejam:

- Não exigem que o número desejado de grupos seja informado de antemão, ao contrário dos algoritmos k-médias e SOM, e determinam dinamicamente o número de grupos de acordo com a estrutura dos padrões de entrada apresentados à RNA, isto é, são auto-organizáveis;

- São especialmente projetadas para resolver o dilema da estabilidade/plasticidade, descrito na Seção 3.4.2.

Neste Capítulo foi apresentada a proposta deste trabalho, bem como os trabalhos correlatos e as diferenças entre eles e a proposta. O Capítulo 5 apresenta os experimentos realizados e seus resultados.

CAPÍTULO 5

EXPERIMENTOS

Neste capítulo são apresentados os experimentos realizados com o protótipo do ARBTF descrito no Capítulo 4. Os resultados obtidos são discutidos.

5.1 MÉTODO

Os experimentos foram executados em arquivos de imagem extraídos de três mídias de armazenamento apreendidas pela Polícia Federal. As duas primeiras mídias, um disco rígido de computador e um dispositivo de memória *flash* removível (*pen drive*), são relacionadas à investigação de supostas fraudes em licitações. A terceira mídia, um disco rígido de computador, foi apreendida no interesse de uma investigação de suposta fraude em saques do Programa Bolsa-Família. Ambas as apreensões foram realizadas durante o cumprimento de mandados de busca e apreensão, conforme citado na Seção 4.5.

Os arquivos de imagem serão descritos pelo termo *evidência*, conforme definido na Seção 2.3 e detalhado na Seção 2.5.1, e estão relacionados na Tabela 5.1.

Tabela 5.1: Evidências examinadas

Evidência	Tipo	Tamanho	Suposto crime
Nº 1	Disco rígido	160 GB	Fraude em licitação
Nº 2	Memória *flash* removível	16 GB	Fraude em licitação
Nº 3	Disco rígido	160 GB	Fraude em saques do Bolsa-Família

As duas primeiras evidências são relacionadas à investigação de supostas fraudes em licitações. Sedes de empresas e de entes da administração pública, a exemplo de prefeituras e órgãos públicos, são alvos comuns dos mandados de busca e apreensão dessa categoria de

investigação. As mídias apreendidas frequentemente contêm muitos arquivos textuais de usuário porque são usados na rotina de trabalho da organização investigada. O requisitante dos exames formulou uma série de quesitos e elencou uma lista de palavras-chaves composta de nomes de organizações e indivíduos. Os documentos que fizessem referência aos indivíduos e organizações investigados deveriam ser organizados e entregues juntamente com o laudo pericial, bem como deveriam ser apontados eventuais indícios de autoria, materialidade e dinâmica das condutas investigadas.

A terceira evidência é relacionada à investigação de suposta fraude em saques do Programa Bolsa-Família. O investigado é um funcionário de um banco público, que já era investigado em um procedimento administrativo disciplinar do banco. Foi apreendido o computador pessoal do investigado. O requisitante dos exames não formulou quesitos nem elencou palavras-chaves; apenas requisitou que fossem buscados indícios da prática do suposto crime e forneceu o nome completo do funcionário público.

As evidências foram submetidas ao processamento do módulo ARBTFi, conforme descrito na Seção 4.1.1. Ao final do processamento, o índice a ser consultado pelo módulo ARBTFb, descrito na Seção 4.1.2, estava pronto.

O módulo ARBTFb foi executado com diversos termos de busca, e os *hits* foram agrupados com o algoritmo ART1, cujos parâmetros foram definidos conforme discutido na Seção 5.1.2. Em seguida, as soluções de agrupamento obtidas foram validadas utilizando-se o índice externo NMI descrito na Seção 3.3.3. As classes padrão utilizadas para validação foram elaboradas conforme descrito na Seção 5.1.1.

5.1.1 ELABORAÇÃO DAS CLASSES PADRÃO

As classes padrão para os documentos de todos os casos foram elaboradas por um especialista de acordo com o conteúdo dos documentos, conforme descrito na Seção 4.4. Esse critério foi citado por Manning et al. (2008) como forma de validar os resultados obtidos.

O especialista que elaborou as classes padrão também elaborou os laudos periciais relacionados aos casos investigados. Durante a elaboração das classes padrão, o especialista levou em conta o conteúdo semântico dos documentos, o nome do arquivo, bem como o título do documento (quando este o possuía), e a partir desses elementos definiu as classes e seus rótulos.

5.1.2 PARÂMETROS DA RNA ART1

Conforme descrito na Seção 3.5.1, o parâmetro de escolha (β) recebeu um valor pequeno, de $0,001$, definido empiricamente. Este limite conservador minimiza a recodificação, ou seja, a atualização dos protótipos durante o aprendizado, conforme descrito na Seção 3.5.1.

O valor padrão do parâmetro de vigilância foi $\rho = \frac{1}{\ln(N \times M)}$, onde N é a quantidade e M é a dimensionalidade total dos documentos a agrupar. O ponto de partida para definir esse valor, que é empírico, foi a sugestão de Massey (2005a) para o cálculo do valor da *vigilância útil mínima*, que o autor definiu como $\rho_{min} = \frac{1}{M}$. A coleção de dimensionalidade máxima que o autor utilizou em seus testes tinha $M = 2.600$, e as demais tinham dimensionalidades bem menores, onde M variava de 500 a 800. Esse valor não é adequado para o propósito desta pesquisa porque, exceto pelo uso de uma lista de *stopwords* e de um processo de *stemming*, não foi aplicada nenhuma técnica de redução de dimensionalidade; conjuntos de *hits* que continham muitos termos ocasionaram alta dimensionalidade (M na casa dos milhares).

A forma de cálculo do valor do parâmetro de vigilância descrita acima é apenas uma sugestão de natureza empírica. O ARBTFb permite que o usuário apresente o valor que desejar, com o fim de obter grupos de maior ou menor granularidade, conforme descrito na Seção 3.5.1. Entretanto, em nenhum dos experimentos que tiveram análise qualitativa, os usuários julgaram necessário especificar o valor do parâmetro de vigilância.

5.2 EXPERIMENTOS COM A EVIDÊNCIA Nº 1

As características detalhadas da evidência nº 1 estão detalhadas na Tabela 5.2, e suas quantidades de documentos estão detalhadas na Tabela 5.3.

Tabela 5.2: Evidência nº 1

Tipo	Tamanho	Quantidade de arquivos	Documentos textuais	Documentos indexados
Disco rígido	160 GB	49.480	5.970	3.455

Nem todos os documentos textuais presentes na mídia foram indexados. Isto se deve ao fato que alguns deles eram duplicatas ou não passaram no teste de idioma, conforme descrito na Seção 4.1.1.

Tabela 5.3: Quantidades de documentos textuais da evidência nº 1

Tipo de documento	Quantidade original	Quantidade após pré-processamento
Arquivos de usuário do pacote Microsoft Office (.doc, .docx, .xls, .xlsx, .ppt, .pptx, .pps, .ppsx)	767	578
Arquivos de texto plano (.txt)	737	118
Arquivos PDF e RTF (.pdf e .rtf)	450	196
Arquivos HTML e XML (.htm, .html e .xml)	4.016	2.563
Total	**5.970**	**3.455**

Para a evidência nº 1, foram aplicados termos de busca relacionados ao caso, que estão relacionados na Tabela 5.4. Os termos de busca foram aqueles elencados pelo requisitante dos exames, conforme descrito na Seção 4.5, como os principais indivíduos e organizações investigados. Os nomes reais foram trocados por identificadores genéricos para não expor os investigados.

Tabela 5.4: Termos de busca relacionados ao caso

Termo
Indivíduo nº 1
Indivíduo nº 2
Indivíduo nº 3
Organização nº 1
Organização nº 2

As Tabelas de contingência 5.5, 5.8, 5.9, 5.11, 5.13 e 5.15 relacionam os grupos obtidos às partições de referência para cada termo de busca.

As Tabelas 5.6, 5.10, 5.12, 5.14 e 5.16 descrevem as quantidades de termos dos grupos e de seus protótipos, bem como as quantidades de documentos total e por tipo. Segue a descrição das colunas:

- #Total Termos – quantidade total de termos únicos presentes nos documentos contidos no grupo;
- #Termos Prot. – quantidade total de termos únicos presentes no protótipo do grupo;
- #Docs – quantidade de documentos contidos no grupo;

- #DocsTE – quantidade de documentos no formato de texto estruturado: Microsoft Word (.doc e .docx), PDF (.pdf) e RTF (.rtf);
- #DocsTP – quantidade de documentos no formato de texto plano (.txt);
- #DocsPL – quantidade de documentos em formato de planilha: Microsoft Excel (.xls e .xlsx);
- #DocsML – quantidade de documentos em formato de linguagens de marcação (*markup languages*): HTML e XML (.htm, .html e .xml).

5.2.1 EVIDÊNCIA Nº 1 - TERMO DE BUSCA Nº 1

Para este teste foram realizadas as duas análises, quantitativa e qualitativa.

A classe "HISTÓRICO DE MSN[1]" é a mais numerosa da Tabela 5.5, e 17 de seus 19 documentos foram concentrados em 3 dos 19 grupos que foram gerados pelo algoritmo, e os últimos 2 documentos foram alocados em um único grupo. Uma análise detalhada revela que os protótipos dos grupos 8, 9, 15 e 17 são muito diferentes. As quantidades de termos dos protótipos dos grupos constam da Tabela 5.6. A interseção entre os termos dos protótipos dos grupos está na Tabela 5.7.

[1]Arquivos contendo registros históricos do programa de mensagens instantâneas *Windows Live Messenger*(Microsoft, 2011).

Tabela 5.5: Tabela de contingência do termo “Indivíduo nº 1” na evidência nº 1

Classes\Grupos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	Soma
1) ATESTADO							1													1
2) AULA																			1	1
3) CERTIDÃO						1														1
4) DECLARAÇÃO							1			3	2			2						8 (3º)
5) CAPA DE ENVELOPE										1										1
6) ESPELHO DE NOTA			4		1															5
7) LEI																		1		1
8) MANUAL																			1	1
9) HISTÓRICO DE MSN								9	4						4		2			19 (1º)
10) ORÇAMENTO					1															1
11) PROCURAÇÃO						2														2
12) PROPOSTA COMERCIAL			2							1		4	6	1		4				18 (2º)
13) DIGITALIZAÇÃO				8																8
14) SENHA		1																		1
15) SOLICITAÇÃO							1													1
16) PAPEL TIMBRADO	1																			1
Soma	1	1	6	8	2	3	3	9	4	5	2	4	6	3	4	4	2	1	2	70

Tabela 5.6: Dados dos grupos e protótipos para o termo “Indivíduo nº 1” na evidência nº 1

Grupo	#Total Termos	#Termos Prot.	#Docs	#DocsTE	#DocsTP	#DocsPL	#DocsML
1	4	4	1	1			
2	6	6	1	1			
3	380	19	6			6	
4	17	2	8	8			
5	139	8	2			2	
6	222	13	3	3			
7	75	3	3	3			
8	185	10	9				9
9	247	16	4				4
10	125	8	5	4		1	
11	60	51	2	2			
12	777	35	4			4	
13	641	22	6			6	
14	244	15	3	2		1	
15	486	18	4				4
16	671	46	4			4	
17	658	73	2				2
18	5016	5016	1	1			
19	4168	661	2	2			

Tabela 5.7: Interseção entre os termos dos protótipos dos grupos da classe "HISTÓRICO DE MSN"

Protótipo	P8	P9	P15	P17
P8	10	10	10	10
P9	10	16	10	11
P15	10	10	18	14
P17	10	11	14	73

Em seguida vem a classe "PROPOSTA COMERCIAL", cujos 18 documentos foram alocados em 6 grupos diferentes, sendo que 14 documentos foram alocados em 3 grupos. Novamente verificou-se que os protótipos dos grupos são muito diferentes. Dado o grande esforço manual necessário para a montagem da tabela que mostra a quantidade de termos compartilhados entre os protótipos, esta não foi calculada neste caso, e somente foi feita inspeção visual da lista de termos de cada protótipo para atestar a diferença entre eles. As propostas do grupo 12 tratavam principalmente de material escolar, de escritório e de limpeza; as do grupo 13 versavam sobre móveis e eletrodomésticos; e as do grupo 16 cotavam gêneros alimentícios.

A terceira classe mais numerosa, "DECLARAÇÃO", apresentou-se bastante fragmentada: seus 8 documentos estão dispersos em 4 grupos diferentes. A inspeção dos protótipos, mais uma vez, demonstrou que compartilhavam poucos termos em comum. Os documentos presentes em cada grupo compartilhavam todos os termos contidos no protótipo, e por isso foram considerados semelhantes pelo algoritmo.

A classe "DIGITALIZAÇÃO" contém 8 documentos. O grupo 4 produzido pelo algoritmo apresenta perfeita correspondência com a classe. Após análise, foi constatado que todos os documentos da classe não continham texto, apenas imagens digitalizadas de documentos exigidos para habilitação em certames licitatórios; os vetores que representavam os documentos continham basicamente os metadados, cuja semelhança foi suficiente para que pertencessem ao mesmo grupo. O protótipo do grupo continha apenas 2 termos, que estavam presentes em todos os documentos que foram associados ao grupo.

A classe "ESPELHO DE NOTA" contém 5 documentos, dos quais 4 estão no grupo 3 e 1 está no grupo 5. O protótipo do grupo 3 possui 19 termos únicos e o do grupo 5 possui 8, conforme pode ser visto na Tabela 5.6; os protótipos compartilham apenas 4 termos em comum. Durante inspeção detalhada dos documentos pertencentes à classe constatou-se que, apesar de pertencerem à mesma classe, são muito diferentes, tanto porque possuem poucos termos em comum quanto porque possuem quantidades muito diferentes de termos.

Documentos das classes “ATESTADO”, “SOLICITAÇÃO” e “DECLARAÇÃO” foram associados ao mesmo grupo, qual seja, o grupo 7. Os 3 documentos presentes no grupo possuíam 75 termos ao todo e tinham teores diferentes, porém compartilhavam os 3 termos do protótipo; como um dos testes que determinam a alocação de um documento a um grupo é sua correspondência com o respectivo protótipo, conforme discutido na Seção 3.5.1, um protótipo com poucos termos mas que oferece melhor correspondência com o vetor de entrada que outros protótipos com mais termos será o escolhido como o mais próximo. Dessa forma, foram associados ao mesmo grupo.

A classe “LEI” continha um único documento, cujo texto era longo e continha muitos termos que não eram encontrados em nenhum dos outros documentos das outras classes. O algoritmo, corretamente, criou um grupo cujo único documento era esse.

A análise quantitativa obteve o valor $\phi^{(NMI)} = 0,76$ para a solução de agrupamento. Dado que o valor máximo de $\phi^{(NMI)}$ é 1, o resultado obtido pode ser considerado positivo.

A análise qualitativa foi realizada por três Peritos Criminais Federais do Departamento de Polícia Federal. Seguem suas considerações:

- Todos afirmaram que era preferível revisar os *hits* na forma de grupos a fazê-lo na forma de lista exaustiva não-agrupada;

- Todos reclamaram que os rótulos dos grupos, que consistiam apenas de números, não transmitiam de forma imediata o teor dos documentos contidos no grupo, sendo necessário inspecionar o conteúdo dos documentos individualmente;

- Um deles afirmou que, mesmo nos casos em que fosse necessário visualizar todos os documentos de todos os grupos, a abordagem do agrupamento tinha o mérito de permitir que documentos com conteúdos assemelhados seriam examinados em sequência, ao passo que a lista exaustiva não-agrupada não dispunha de nenhum recurso para dividir os documentos em conjuntos temáticos;

- Um deles, ao verificar semelhanças entre dois grupos cujos protótipos compartilhavam termos em comum, questionou se o ARBTF oferecia a possibilidade de aglutinar grupos.

Os resultados das duas análises sugerem que a abordagem proposta nesta dissertação e materializada através do protótipo é útil.

5.2.2 EVIDÊNCIA Nº 1 - TERMO DE BUSCA Nº 2

Não foi possível obter voluntários para realizar a análise qualitativa deste teste. Dessa forma, somente foi realizada a análise quantitativa.

A tabela de contingência relativa a este termo de busca foi dividida em duas por conta da quantidade de grupos produzida pelo algoritmo.

Nas Tabelas 5.8 e 5.9, a classe mais numerosa é "PROPOSTA COMERCIAL". Seus 34 documentos estão dispersos em 15 grupos, sendo que 23 documentos estão concentrados em 6 grupos. Os documentos contêm quantidades de termos muito variadas, e os protótipos de seus respectivos grupos também têm tamanhos muito diferentes, conforme apresentado na Tabela 5.10. Em inspeção visual, os grupos cujos protótipos possuíam menor quantidade de termos continham documentos menos parecidos entre si, bem como continham documentos pertencentes a outras classes; grupos cujos protótipos possuíam maior quantidade de termos continham documentos mais semelhantes entre si.

A segunda classe mais numerosa é "DECLARAÇÃO", com 31 documentos alocados em 6 grupos diferentes; os 3 grupos mais numerosos contêm 24 documentos. Os grupos 2, 4, 5, 6 e 8 não contêm documentos de outras classes. Os termos do protótipo do grupo 29 são nomes próprios e de cidades, e talvez por isso o grupo contenha 1 documento da classe "CARTA" e todos os 5 documentos da classe "SOLICITAÇÃO".

A terceira classe mais numerosa é "HISTÓRICO DE MSN". O algoritmo conseguiu agrupar seus 25 documentos em 4 grupos; considerando que há 43 grupos ao todo, e que 20 dos documentos estão concentrados em apenas duas classes, o algoritmo conseguiu discriminar bem os documentos desta classe. É interessante notar que os 10 termos do protótipo do grupo 25 estão contidos no protótipo do grupo 23, que tem 23 termos; um possível caminho de melhoria no algoritmo seria que ele detectasse essa condição e permitisse a aglutinação dos grupos.

A quarta classe mais numerosa é "PLANILHA ORÇAMENTÁRIA", que contém planilhas orçamentárias de obras de engenharia civil. Seus 23 documentos estão alocados em 6 grupos, sendo que 18 estão alocados em 3 grupos. Os protótipos dos grupos 17 e 35 possuem respectivamente 24 e 27 termos, ao passo que o protótipo do grupo 11 possui apenas 6 termos. O documento da classe "PROPOSTA COMERCIAL" que foi alocado no grupo 35 elenca uma relação exaustiva de bens e serviços cujos termos constam dos documentos da classe "PLANILHA ORÇAMENTÁRIA" e também do protótipo do grupo.

A quinta classe mais numerosa é "ESPELHO DE NOTA". Dos 11 documentos desta classe, 8 foram alocados nos grupos 3 e 21, e os 3 demais documentos foram alocados cada

um em um grupo à parte. Os protótipos dos grupos 3 e 21, cada um com 4 documentos, continham principalmente termos presentes nos documentos da classe.

A sexta classe mais numerosa é "PROCURAÇÃO", com 10 documentos alocados em 3 grupos. Os protótipos dos grupos 31 e 36 contêm muitos termos em comum com os documentos da classe, e os grupos não contêm documentos de outras classes; o protótipo do grupo 18 possui menos termos em comum com os documentos da classe "PROCURAÇÃO" nele contidos, e talvez por isso também contenha um documento da classe "CARTA".

Embora os documentos de várias classes tenham sido alocados em vários grupos diferentes, a maioria dos grupos continha documentos de apenas uma ou duas classes. O valor final da estimativa $\phi^{(NMI)}$ da solução de agrupamento foi de $0,725$.

Tabela 5.8: Tabela de contingência do termo “Indivíduo nº 2” na evidência nº 1 - grupos de 1 a 20

Classes\Grupos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Soma
1) AULA																					1
2) DOC. BANCÁRIO	1																				1
3) FICHA CADASTRAL																					1
4) CARTA																		1			3
5) COMPROVANTE									1												1
6) CONTRATO																					1
7) CONTROLE DE CHEQUES	1																		1		2
8) CURRÍCULO																					2
9) DECLARAÇÃO		3		10	4	1		10													31 (2º)
10) EDITAL																					1
11) CAPA DE ENVELOPE																					1
12) ESPELHO DE NOTA			4												1	1					11 (5º)
13) FOLHA DE PAGAMENTO	1																				2
14) MANUAL																					2
15) MODELO DE PROPOSTA									1												1
16) HISTÓRICO DE MSN																					25 (3º)
17) NOTÍCIA-CRIME									1												1
18) COTAÇÃO												1			1						2
19) PLANILHA ORÇAMENTÁRIA											5			2			10				23 (4º)
20) PLANILHA DE PAGAMENTOS							2			1											3
21) PROCURAÇÃO																		3			10 (6º)
22) PROPOSTA COMERCIAL			2								1	1	2	1		5				4	34 (1º)
23) RELAÇÃO DE PAGAMENTOS	1																				1
24) DIGITALIZAÇÃO	1																				1
25) SENHA	3																				3
26) SOLICITAÇÃO																					5
27) PAPEL TIMBRADO																					1
28) TRABALHO ACADÊMICO																					1
Soma	8	3	6	10	4	1	2	10	3	1	6	2	2	3	2	6	10	4	1	4	171

Tabela 5.9: Tabela de contingência do termo “Indivíduo nº 2” na evidência nº 1 - grupos de 21 a 43

Classes\Grupos	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	Soma
1) AULA																							1	1
2) DOC. BANCÁRIO																								1
3) FICHA CADASTRAL						1																		1
4) CARTA									1	1														3
5) COMPROVANTE																								1
6) CONTRATO																			1					1
7) CONTROLE DE CHEQUES																								2
8) CURRÍCULO								1					1											2
9) DECLARAÇÃO									3															31 (2º)
10) EDITAL						1																		1
11) CAPA DE ENVELOPE								1																1
12) ESPELHO DE NOTA	4													1										11 (5º)
13) FOLHA DE PAGAMENTO													1											2
14) MANUAL																				1		1		2
15) MODELO DE PROPOSTA																								1
16) HISTÓRICO DE MSN			4		16		3													2				25 (3º)
17) NOTÍCIA-CRIME																								1
18) COTAÇÃO																								2
19) PLANILHA ORÇAMENTÁRIA				1										1	3						1			23 (4º)
20) PLANILHA DE PAGAMENTOS																								3
21) PROCURAÇÃO											5					2								10 (6º)
22) PROPOSTA COMERCIAL				1						4		3			1		4	3	1		1			34 (1º)
23) RELAÇÃO DE PAGAMENTOS																								1
24) DIGITALIZAÇÃO																								1
25) SENHA																								3
26) SOLICITAÇÃO									5															5
27) PAPEL TIMBRADO		1																						1
28) TRABALHO ACADÊMICO																						1		1
Soma	4	1	4	2	16	2	3	2	9	5	5	3	2	2	4	2	4	3	2	3	2	2	1	171

Tabela 5.10: Dados dos grupos e protótipos para o termo “Indivíduo nº 2” na evidência nº 1

Grupo	#Total Termos	#Termos Prot.	#Docs	#DocsTE	#DocsTP	#DocsPL	#DocsML
1	76	2	8	7		1	
2	82	5	3	3			
3	277	7	6			6	
4	143	8	10	10			
5	106	18	4	4			
6	9	9	1	1			
7	12	9	2			2	
8	163	6	10	10			
9	145	6	3	3			
10	18	18	1			1	
11	227	6	6	1		5	
12	244	23	2			3	
13	350	38	2			2	
14	197	9	3			3	
15	161	7	2			2	
16	492	28	6	1		5	
17	582	24	10			10	
18	145	7	4	4			
19	7	7	1	1			
20	328	19	4			4	
21	258	16	4			4	
22	5	5	1	1			
23	476	23	4				4
24	363	21	2			2	
25	516	10	16				16
26	611	45	2	1		1	
27	731	50	3				3
28	143	10	2	1		1	
29	190	6	9	9			
30	483	18	5	1		4	
31	142	71	5	5			
32	589	63	3	1		2	
33	195	10	2	1		1	
34	352	16	2			2	
35	768	27	4			4	
36	109	71	2	2			
37	570	27	4	3		1	
38	464	41	3			3	
39	446	35	2	1		1	
40	1720	100	3	1			2
41	705	79	2			2	
42	2174	235	2	2			
43	3125	3125	1	1			

5.2.3 EVIDÊNCIA Nº 1 - TERMO DE BUSCA Nº 3

Não foi possível obter voluntários para realizar a análise qualitativa deste teste. Dessa forma, somente foi realizada a análise quantitativa.

Tabela 5.11: Tabela de contingência do termo "Indivíduo nº 3" na evidência nº 1

Classes\Grupos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Soma
1) CERTIDÃO														7		7 (3º)
2) PLANILHA DE CUSTO							2									2
3) DECLARAÇÃO	1		5	2	2	3				3						16 (2º)
4) ESPELHO DE NOTA		3														3
5) FORMULÁRIO	2															2
6) HISTÓRICO DE MSN								2								2
7) PROCURAÇÃO												1				1
8) PROPOSTA COMERCIAL		1					1		5		4	1	8		2	22 (1º)
9) SOLICITAÇÃO				1												1
Soma	3	4	5	3	2	3	3	2	5	3	4	2	8	7	2	56

Tabela 5.12: Dados dos grupos e protótipos para o termo "Indivíduo nº 3" na evidência nº 1

Grupo	#Total Termos	#Termos Prot.	#Docs	#DocsTE	#DocsTP	#DocsPL	#DocsML
1	39	5	3	3			
2	357	19	4			4	
3	89	3	5	5			
4	84	9	3	3			
5	102	95	2	2			
6	86	7	3	3			
7	431	31	3			3	
8	109	16	2				2
9	511	15	5			5	
10	85	21	3	3			
11	558	28	4			4	
12	266	23	2	1		1	
13	757	40	8			8	
14	89	87	7	7			
15	392	387	2			2	

Na Tabela 5.11, a classe mais numerosa é "PROPOSTA COMERCIAL", cujos 22 documentos foram alocados em 7 grupos diferentes pelo algoritmo, sendo que 17 documentos foram alocados em 3 grupos. A análise detalhada dos documentos de cada grupo mostrou que, a despeito de serem todos classificados como declarações, são baseados em modelos diferentes, cuja estrutura foi capturada pelo algoritmo na forma de grupos separados.

Os 16 documentos da segunda classe mais numerosa, "DECLARAÇÃO", foram alocados em 6 grupos diferentes. Os protótipos dos grupos possuíam muitos termos em comum com seus respectivos documentos; uma análise detalhada revelou que os documentos, embora tenham sido classificados na mesma classe, eram baseados em modelos diferentes, de forma semelhante ao que ocorreu com os documentos da classe "PROPOSTA COMERCIAL".

Todos os 7 documentos da classe mais numerosa, "CERTIDÃO" foram alocados ao grupo 14. Isso se deveu ao fato de que os documentos eram quase idênticos; a única diferença entre eles era um número de controle, que não foi indexado pelo módulo ARBTFi pelos motivos expostos na Seção 4.2. Dessa forma, do ponto de vista do indexador e do buscador, os documentos eram iguais. Essa correspondência perfeita entre a classe e o grupo contribuiu para elevar o valor da estimativa $\phi^{(NMI)}$.

Comparada às quantidades constantes das duas primeiras tabelas de contingência, a quantidade de *hits* obtida com o termo "Indivíduo nº 3" parece pequena. Ainda comparado aos primeiros resultados, a quantidade de classes e grupos tambem é menor. Foi obtida a estimativa $\phi^{(NMI)} = 0,713$.

5.2.4 EVIDÊNCIA Nº 1 - TERMO DE BUSCA Nº 4

Não foi possível obter voluntários para realizar a análise qualitativa deste teste. Dessa forma, somente foi realizada a análise quantitativa.

Na Tabela 5.13, há um total de 23 classes e 27 grupos. As 5 classes mais numerosas concentram 146 dos 185 documentos; os 5 grupos mais numerosos concentram 112 documentos.

A classe "DECLARAÇÃO" é a mais numerosa e mais fragmentada. Seus 52 documentos estão alocados em 10 grupos diferentes. O grupo 3 contém 13 documentos, um quarto do total, e 2 dos 5 termos de seu protótipo são específicos de declarações: os radicais *decl* e *declar*. Os grupos 6 e 15 também possuem ao menos um desses termos em seus protótipos. Os demais grupos não possuem nenhum dos termos, e talvez por isso também contenham documentos de outras classes.

Todos os 34 documentos da segunda classe mais numerosa, "CERTIDÃO" foram associados ao grupo 13. Isso se deveu ao fato de que os documentos eram quase idênticos; a única diferença entre eles era um número de controle, que não foi indexado pelo módulo ARBTFi. Dessa forma, do ponto de vista do indexador e do buscador, os documentos eram iguais. Essa correspondência perfeita entre a classe e o grupo contribuiu para elevar o valor da estimativa $\phi^{(NMI)}$.

A terceira classe mais numerosa, "PROPOSTA COMERCIAL", teve seus 28 documentos espalhados em 7 grupos, sendo que os 3 grupos mais numerosos receberam 21 documentos. Os grupos 14, 15, 23 e 24 continham o termo *propost* em seus protótipos. Os protótipos dos grupos 14 e 15 continham respectivamente 10 e 13 termos; os protótipos dos grupos 23 e 24 continham respectivamente 69 e 47 termos, conforme consta na Tabela 5.14.

Isso sugeriu que os documentos dos grupos 14 e 15 eram mais genéricos, e os documentos dos grupos 23 e 24 eram mais específicos; em inspeção mais detalhada, isso se verificou verdadeiro.

A classe "DIGITALIZAÇÃO" é a quarta mais numerosa. O grupo 11 produzido pelo algoritmo contém todos os documentos dessa classe. Após análise, foi constatado que todos os documentos da classe "DIGITALIZAÇÃO" não continham texto, apenas imagens digitalizadas de documentos exigidos para habilitação em certames licitatórios; os vetores que representavam os documentos continham basicamente os termos extraídos dos metadados *Título* e *Autor*, cuja semelhança foi suficiente para que pertencessem ao mesmo grupo.

A quinta classe mais numerosa, "PLANILHA ORÇAMENTÁRIA", continha 10 documentos que foram alocados em 5 grupos diferentes, sendo que 4 dos documentos foram alocados no grupo 9. Nesse mesmo grupo estão contidos 3 documentos da classe "CRONOGRAMA DE OBRA". O protótipo do grupo continha 13 termos, sendo que 4 deles podiam ser encontrados em documentos das duas classes.

A classe "PROCURAÇÃO" é a sexta mais numerosa, e seus 8 documentos foram todos alocados no grupo 20. Dentre os 10 termos do protótipo do grupo, havia 4 nomes próprios e de organizações; esse talvez seja o motivo de que o grupo também contém 11 documentos da classe "DECLARAÇÃO".

As demais classes apresentaram quantidades relativamente pequenas de documentos, concentrados em 1 ou 2 grupos para cada classe, exceto a classe "PESQUISA DE PREÇOS", cujos documentos foram associados a 3 grupos.

A solução de agrupamento obteve estimativa $\phi^{(NMI)} = 0,719$.

Tabela 5.13: Tabela de contingência do termo “Organização nº 1” na evidência nº 1

Classes\Grupos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	Soma
1) ATESTADO					1																							1
2) CARTA										2																		2
3) CERTIDÃO													34															34 (2º)
4) CERTIDÃO NEGATIVA																1												1
5) CONTRATO																										1		1
6) CRONOGRAMA DE OBRA		1							3																			4
7) CURRÍCULO																						1						1
8) DECLARAÇÃO			13	4		2				9			5		5				1	11		1				1		52 (1º)
9) CAPA DE ENVELOPE				1																								1
10) FOLHA DE PONTO							1																					1
11) FOLHA DE PAGAMENTO							2																					2
12) FORMULÁRIO				1																								1
13) LISTA DE PAGAMENTOS								1																				1
14) HISTÓRICO DE MSN																		2							4			6
15) OFÍCIO					1																							1
16) PESQUISA DE PREÇOS												1							2		1							4
17) PLANILHA ORÇAMENTÁRIA		1							4			2			1												2	10 (5º)
18) PROCURAÇÃO																				8								8 (6º)
19) PROPOSTA COMERCIAL		1								6				7	8							1	2	3				28 (3º)
20) DIGITALIZAÇÃO											22																	22 (4º)
21) SENHA	2																											2
22) SOLICITAÇÃO										1																		1
23) XML DE SISTEMA																	1											1
Soma	2	3	13	6	2	2	3	1	7	18	22	3	39	7	14	1	1	2	3	19	1	3	2	3	4	2	2	185

Tabela 5.14: Dados dos grupos e protótipos para o termo “Organização nº 1” na evidência nº 1

Grupo	#Total Termos	#Termos Prot.	#Docs	#DocsTE	#DocsTP	#DocsPL	#DocsML
1	8	2	2	1	1		
2	190	11	3			3	
3	159	6	13	13			
4	108	4	6	6			
5	50	12	2	2			
6	87	6	2	2			
7	80	34	3	2		1	
8	18	18	1			1	
9	269	13	7			7	
10	327	9	18	17		1	
11	31	2	22	22			
12	522	28	3			3	
13	225	10	39	39			
14	367	10	7	1		6	
15	466	13	14	6		8	
16	63	63	1	1			
17	116	116	1				1
18	36	10	2				2
19	418	20	3	1		2	
20	373	10	19	19			
21	162	162	1			1	
22	219	9	3	2		1	
23	280	69	2			2	
24	673	47	3			3	
25	668	37	4				4
26	271	33	2	2			
27	574	154	2			2	

5.2.5 EVIDÊNCIA Nº 1 - TERMO DE BUSCA Nº 5

Não foi possível obter voluntários para realizar a análise qualitativa deste teste. Dessa forma, somente foi realizada a análise quantitativa.

Tabela 5.15: Tabela de contingência do termo "Organização nº 2" na evidência nº 1

Classes\Grupos	1	2	3	4	5	6	7	8	9	10	11	12	13	Soma
1) AULA											1			1
2) DECLARAÇÃO								1						1
3) MANUAL													2	2
4) MODELO DE RECIBO	1													1
5) HISTÓRICO DE MSN		1	17	7	4	8	6			5		5		53 (1º)
6) PROCURAÇÃO								1						1
7) PROPOSTA COMERCIAL								1						1
8) RECURSO ADMINISTRATIVO									1					1
9) SOLICITAÇÃO	1													1
10) PAPEL TIMBRADO	1													1
Soma	3	1	17	7	4	8	6	3	1	5	1	5	2	63

Tabela 5.16: Dados dos grupos e protótipos para o termo "Organização nº 2" na evidência nº 1

Grupo	#Total Termos	#Termos Prot.	#Docs	#DocsTE	#DocsTP	#DocsPL	#DocsML
1	54	4	3	3			
2	15	15	1				1
3	532	8	17				17
4	406	11	7				7
5	495	22	4				4
6	558	12	8				8
7	893	31	6				6
8	275	12	3	2		1	
9	160	160	1	1			
10	991	30	5				5
11	3125	3125	1	1			
12	1552	71	5				5
13	2322	506	2	2			

Na Tabela 5.15 a classe "HISTÓRICO DE MSN" concentra 53 dos 63 documentos. O nome da organização investigada é um substantivo comumente usado em relações interpessoais, e foi constatado que muitos dos *hits* eram falsos-positivos, cujo conteúdo não tinha a ver com a organização. Os documentos da classe encontravam-se distribuídos em 8 dos 13 grupos.

As demais classes apresentavam correspondência com os demais grupos. No entanto, como os documentos da classe "HISTÓRICO DE MSN" apresentavam-se dispersos em vários grupos e constituíam a maioria dos documentos obtidos pela busca, obteve-se $\phi^{(NMI)} = 0,5$, o valor mais baixo entre todos os calculados para os termos de busca da evidência nº 1.

5.2.6 EVIDÊNCIA Nº 1 - RESUMO E DISCUSSÃO

A Tabela 5.17 apresenta um resumo das buscas submetidas à evidência nº 1 e seus resultados.

Tabela 5.17: Tabela-resumo das buscas na evidência nº 1

Termo de busca	Resultados	Dimensões	Classes	Grupos	$\phi^{(NMI)}$	ρ
Indivíduo nº 1	70	8.682	16	19	0,76	0,075089
Indivíduo nº 2	171	8.064	28	43	0,725	0,070737
Indivíduo nº 3	56	1.997	9	15	0,713	0,086023
Organização nº 1	185	3.061	23	27	0,719	0,07549
Organização nº 2	63	6.804	10	13	0,5	0,077111

Foi constatado que os grupos tenderam a conter arquivos do mesmo tipo, como pode ser visto nas Tabelas 5.6, 5.10, 5.12, 5.14 e 5.16. Isso sugere que talvez seja útil classificar os *hits* por tipo antes de agrupá-los, porque a classificação por tipo de arquivo é trivial, não sendo necessário usar um algoritmo de agrupamento para fazê-lo.

O valor calculado automaticamente para o parâmetro de vigilância produziu grupos que fragmentaram as classes, mas sem misturá-las umas com as outras na maioria dos casos, como pode ser visto nas Tabelas de contingência 5.5, 5.8, 5.9, 5.11, 5.13 e 5.15.

Segundo os Peritos Criminais Federais que realizaram a análise qualitativa descrita na Seção 5.2.1, o agrupamento facilita a criação dos *bookmarks*, descritos na Seção 2.5.

Todas as buscas apresentaram dimensionalidade na casa dos milhares, mas o tempo total de execução do ARBTFb foi de no máximo 5 segundos. Sem a melhoria na forma de armazenamento dos vetores de entrada descrita na Seção 4.3.3, o tempo de execução provavelmente teria sido bem mais longo.

5.3 EXPERIMENTOS COM A EVIDÊNCIA Nº 2

As características da evidência nº 2 estão detalhadas na Tabela 5.18, e suas quantidades de documentos estão detalhadas na Tabela 5.19.

Para a evidência nº 2, dado o seu tamanho reduzido e também a pequena quantidade de documentos nela armazenada, não foram feitas buscas, e o agrupamento foi feito sobre todos os documentos indexados na fase de pré-processamento.

Não foi possível obter voluntários para realizar a análise qualitativa deste teste. Dessa forma, somente foi realizada a análise quantitativa.

Tabela 5.18: Evidência nº 2

Tipo	Tamanho	Quantidade de arquivos	Documentos textuais	Documentos indexados
Memória *flash* removível (*pen drive*)	16 GB	3.756	160	156

Tabela 5.19: Quantidades de documentos textuais da evidência nº 2

Tipo de documento	Quantidade original	Quantidade após pré-processamento
Arquivos de usuário do pacote Microsoft Office (.doc, .docx, .xls, .xlsx, .ppt, .pptx, .pps, .ppsx)	39	39
Arquivos de texto plano (.txt)	2	2
Arquivos PDF e RTF (.pdf e .rtf)	119	115
Arquivos HTML e XML (.htm, .html e .xml)	0	0
Total	**160**	**156**

Na Tabela 5.20, há 31 classes ao todo e 114 dos 156 documentos estão concentrados nas 3 classes mais numerosas; os 3 grupos mais numerosos contêm 117 documentos, em correspondência quase perfeita com as classes.

A classe "DESENHO (CAD)" é a mais numerosa, e o grupo 25 produzido pelo algoritmo possui correspondência idêntica com ela. Após análise, foi constatado que todos os documentos da classe não continham texto, somente desenhos técnicos de CAD[2].

A classe "CERTIDÃO" teve todos os seus 33 documentos associados a um único grupo, que por sua vez continha apenas 3 documentos de outra classe. Isso se deveu ao fato de que os documentos que continham certidões eram quase idênticos; a única diferença entre eles era um número de controle, que não foi indexado pelo módulo ARBTFi. Dessa forma, do ponto de vista do indexador e do buscador, os documentos eram iguais. Essa correspondência quase perfeita entre a classe e o grupo contribuiu para elevar o valor da estimativa $\phi^{(NMI)}$.

A classe "DIGITALIZAÇÃO" também tem 33 documentos. O grupo 2 produzido pelo algoritmo contém todos os documentos dessa classe, em correspondência perfeita. Após análise, foi constatado que todos os documentos da classe "DIGITALIZAÇÃO" não continham texto, apenas imagens digitalizadas de documentos exigidos para habilitação em

[2]Do inglês *Computer Aided Design*, ou desenho auxiliado por computador.

certames licitatórios; os vetores que representavam os documentos continham basicamente os termos extraídos dos metadados "Título" e "Autor", cuja semelhança foi suficiente para que pertencessem ao mesmo grupo.

As outras classes continham, em sua maioria, apenas 1 documento cada uma; o algoritmo aglutinou algumas delas em grupos com 2 documentos cada um.

Os bons resultados de agrupamento relativos às classes "DESENHO (CAD)", "DIGITALIZAÇÃO" e "CERTIDÃO" permitiram a obtenção de uma estimativa $\phi^{(NMI)} = 0,95$. O valor é bastante alto, considerando o valor máximo $\phi^{(NMI)} = 1$.

Tabela 5.20: Tabela de contingência de todos os documentos da evidência nº 2

Classes\Grupos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	Soma
1) ARTIGO DE LEI																	1									1
2) AUTORIZAÇÃO				1																						1
3) CAPA DE RELATÓRIO			1																							1
4) CERTIDÃO										33																33 (2º)
5) CURRÍCULO																		2								2
6) DECLARAÇÃO				1		1																				2
7) DESENHO (CAD)																									48	48 (1º)
8) DESPESA DE CAMPANHA																			1							1
9) ESTIMATIVA DE VOTAÇÃO									1																	1
10) ORÇAMENTO DE CAMPANHA					1																					1
11) FICHA CADASTRAL																		1								1
12) LEI																					1					1
13) MANUAL															1											1
14) MEMORIAL DESCRITIVO																							1			1
15) OFÍCIO										3										3						6
16) PLANEJAMENTO DE CAMPANHA														1												1
17) ORÇAMENTO PESSOAL			1																							1
18) ORÇAMENTO DE OBRA													1													1
19) POWERPOINT											1															1
20) PROPOSTA COMERCIAL														1												1
21) REL. DE LICITAÇÕES ABERTAS	2																						1			3
22) REL. DE LICITAÇÕES EM ANDAMENTO									1																	1
23) RELATÓRIO DE RECURSOS																						1				1
24) LISTA DE MATERIAL DE CAMPANHA					5																					5
25) RELAÇÃO DE OBRAS																						1				1
26) RELAÇÃO DE PAGAMENTOS												1														1
27) DIGITALIZAÇÃO		33																								33 (3º)
28) SENHA								1																		1
29) SUMÁRIO DE TRABALHO ACADÊMICO																1										1
30) PAPEL TIMBRADO							1																			1
31) TRABALHO ACADÊMICO																	1							1		2
Soma	2	33	2	2	6	1	1	1	2	36	1	1	1	2	1	1	2	3	1	3	1	2	2	1	48	156

Tabela 5.21: Dados dos grupos e protótipos para o termo “Organização nº 1” na evidência nº 1

Grupo	#Total Termos	#Termos Prot.	#Docs	#DocsTE	#DocsTP	#DocsPL	#DocsML
1	12	10	2	2			
2	22	2	33	33			
3	55	5	2	1		1	
4	85	4	2	2			
5	88	3	6			6	
6	34	34	1	1			
7	18	18	1	1			
8	5	5	1		1		
9	90	9	2	1		1	
10	256	15	36	36			
11	82	82	1	1			
12	68	68	1	1			
13	83	83	1	1			
14	273	14	2	1		1	
15	47	47	1		1		
16	14	14	1	1			
17	355	22	2	2			
18	615	25	3	3			
19	85	85	1			1	
20	119	98	3	3			
21	1778	1778	1	1			
22	429	90	2	1		1	
23	764	58	2	2			
24	160	160	1	1			
25	0	0	48	48			

Tabela 5.22: Tabela-resumo do agrupamento na evidência nº 2

Documentos	Dimensões	Classes	Grupos	$\phi^{(NMI)}$	ρ
156	3.379	31	25	0,95	0,078079

5.3.1 EVIDÊNCIA Nº 2 - RESUMO E DISCUSSÃO

Alguns dos protótipos apresentavam termos como números por extenso e siglas de unidades de medida. Dado que, conforme discutido na Seção 3.2.1, tais termos têm pouco poder discriminante quando isolados de seu contexto original, eles podem ser considerados para inclusão na lista de *stopwords*.

Uma possível explicação para que alguns protótipos tivessem quantidades de termos bem menores que os demais é que o passo 4 do algoritmo ART1, apresentado na Seção 3.5.1, ao incorporar um vetor de entrada, descarta componentes do protótipo valoradas como 1 que no vetor de entrada estejam valoradas como 0. Assim, quando um protótipo incorpora um vetor de entrada com o qual possui poucos termos em comum e o valor da vigilância é baixo, o efeito prático é que os termos do protótipo que não estão presentes no vetor de entrada são descartados e protótipo torna-se "diluído".

5.4 EXPERIMENTOS COM A EVIDÊNCIA Nº 3

As características detalhadas da evidência nº 3 estão detalhadas na Tabela 5.23, e suas quantidades de documentos estão detalhadas na Tabela 5.24.

Tabela 5.23: Evidência nº 3

Tipo	Tamanho	Quantidade de arquivos	Documentos textuais	Documentos indexados
Disco rígido	160 GB	217.872	14.070	3.922

Tabela 5.24: Quantidades de documentos textuais da evidência nº 3

Tipo de documento	Quantidade original	Quantidade após pré-processamento
Arquivos de usuário do pacote Microsoft Office (.doc, .docx, .xls, .xlsx, .ppt, .pptx, .pps, .ppsx)	55	39
Arquivos de texto plano (.txt)	3.459	1.408
Arquivos PDF e RTF (.pdf e .rtf)	50	24
Arquivos HTML e XML (.htm, .html e .xml)	10.506	2.451
Total	**14.070**	**3.922**

Conforme citado na Seção 5.1, para a evidência nº 3 o requisitante dos exames não formulou quesitos nem elencou palavras-chaves; apenas requisitou que fossem buscados indícios da prática do suposto crime e forneceu o nome completo do funcionário público investigado, cujo computador foi apreendido. Dessa forma, a abordagem utilizada para esta evidência foi a de análise exploratória de dados (Tukey, 1980).

Dada a grande quantidade de arquivos presentes na evidência, foram submetidas duas buscas como ponto de partida para a análise exploratória:

1. Todos os documentos de texto estruturado em formato Microsoft Word (.doc e .docx), PDF (.pdf) e RTF (.rtf) presentes na evidência, excluindo-se aqueles presentes em pastas de sistema;

2. Todos os documentos presentes na evidência, sem distinção de tipo e excluindo-se aqueles presentes em pastas de sistema, que contivessem o sobrenome do funcionário público.

Para este teste foram realizadas as duas análises, quantitativa e qualitativa.

5.4.1 EVIDÊNCIA Nº 3 - DOCUMENTOS DE TEXTO ESTRUTURADO

A classe mais numerosa, "CONVERSA DE CORREIO ELETRÔNICO", teve seus 8 documentos alocados em 2 grupos, sendo que um deles contém 6 dos documentos. Ambos os grupos também contêm documentos de outras classes.

A segunda e a terceira classes mais numerosas, respectivamente "FORMULÁRIO DE ANTECIPAÇÃO DE BENEFÍCIO" e "AUTO DE DEFESA", contêm 3 documentos cada uma, sendo que os documentos da primeira estão divididos em 2 grupos (6 e 13), e os da segunda estão divididos em 3 grupos (4, 14 e 15). Uma possível explicação é que os grupos, bem como seus protótipos, têm quantidades de termos muito diferentes, conforme mostrado na Tabela 5.26.

As outras 18 classes possuíam 1 ou 2 documentos cada uma, e todas, exceto a classe "CONTRATO DE LICENÇA DE SOFTWARE", tiveram esses documentos associados a um único grupo.

Obteve-se a estimativa $\phi^{(NMI)} = 0,853$. O valor pode ser considerado alto, porque o valor máximo de $\phi^{(NMI)}$ é 1.

A análise qualitativa foi realizada por um Perito Criminal Federal do Departamento de Polícia Federal. Seguem suas considerações:

- Os nomes de vários arquivos listados na saída não-agrupada davam indicações de seu conteúdo, diminuindo o apelo do agrupamento;
- O grupo 9 conseguiu reunir documentos que continham mensagens de correio eletrônico cujos nomes de arquivo não indicavam essa natureza, sendo que eram todos relevantes para o exame;
- Os grupos 11 e 13 misturaram documentos que, embora possuíssem termos em comum, não possuíam semelhança semântica;
- O grupo 14 mostra que o investigado usou um documento (texto de direito) como base para redigir o outro (auto de defesa);
- Os grupos 14 e 15 têm documentos que inicialmente pareciam bastante semelhantes, mas que, após análise mais detida, apresentavam tamanhos bem diferentes – os do grupo 14 são mais curtos, e os do grupos 15 mais extensos.

Tabela 5.25: Tabela de contingência dos documentos de texto estruturado da evidência nº 3

Classes\Grupos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Somas
1) AUTO DE DEFESA				1										1	1		3 (3º)
2) AUTO DE RECURSO															1		1
3) CERTIDÃO DE ANTECEDENTES							1										1
4) CONTRATO DE EMPRÉSTIMO											1						1
5) CONTRATO DE LICENÇA DE SOFTWARE													1		1		2
6) CONVERSA DE CORREIO ELETRÔNICO				2					6								8 (1º)
7) DECLARAÇÃO DE HISTÓRICO PROFISSIONAL					1												1
8) DECLARAÇÃO DE POBREZA	1																1
9) FORM. DE ANTECIPAÇÃO DE BENEFÍCIO						1							2				3 (2º)
10) FORM. DE DECLARAÇÃO DE DEPENDENTES								1									1
11) FORM. DE REQUERIMENTO DE APOSENTADORIA								1									1
12) HISTÓRICO DE MSN										1							1
13) INTERPOSIÇÃO DE AÇÃO									1								1
14) MANUAL												2					2
15) MODELO DE DOCUMENTO																1	1
16) POEMA										1							1
17) PROCURAÇÃO						2											2
18) REQUERIMENTO											1						1
19) SENHA			2														2
20) TEXTO DE DIREITO														1			1
21) TRABALHO ESCOLAR		1															1
Somas	1	1	2	3	1	3	1	2	7	2	2	2	3	2	3	1	36

Tabela 5.26: Dados dos grupos e protótipos para os documentos de texto estruturado da evidência nº 3

Grupo	#Total Termos	#Termos Prot.	#Docs	#DocsTE	#DocsTP	#DocsPL	#DocsML
1	60	60	1	1			
2	223	223	1	1			
3	12	2	2	2			
4	163	15	3	3			
5	67	67	1	1			
6	281	15	3	3			
7	62	62	1	1			
8	51	5	2	2			
9	376	21	7	7			
10	266	22	2	2			
11	541	50	2	2			
12	513	84	2	2			
13	672	50	3	3			
14	679	91	2	2			
15	1685	277	3	3			
16	0	0	1	1			

5.4.2 EVIDÊNCIA Nº 3 - TERMO DE BUSCA - SOBRENOME

A classe mais numerosa, "HISTÓRICO DE MSN", teve seus 25 documentos alocados em 7 grupos diferentes, 3 deles com 3 documentos (grupos 15, 17 e 21) e os outros 4 com 4 documentos cada um (grupos 16, 18, 19 e 20). Conforme apresentado na Tabela 5.28, os grupos e seus respectivos protótipos possuem quantidades muito diferentes de termos, o que possivelmente contribui para a distribuição observada.

A segunda classe mais numerosa, "ARQ. TEMPORÁRIO DE INTERNET", possui 15 documentos que foram associados a 6 grupos diferentes, sendo que 2 deles (grupos 4 e 9) concentram 10 dos documentos.

A terceira classe mais numerosa, "CONVERSA DE CORREIO ELETRÔNICO", teve seus 8 documentos divididos em 4 grupos com 2 documentos cada um. Os grupos e seus respectivos protótipos tinham quantidades de termos muito diferentes.

As demais classes possuíam de 1 a 3 documentos cada uma, e todas, exceto a classe "AUTO DE DEFESA", tiveram esses documentos alocados em um único grupo.

Obteve-se a estimativa $\phi^{(NMI)} = 0,68$, em virtude da divisão dos documentos das 3 classes mais numerosas em muitos grupos.

A análise qualitativa foi realizada por um Perito Criminal Federal do Departamento de Polícia Federal. Seguem suas considerações:

- Os nomes de vários arquivos listados na saída não-agrupada não davam indicações de seu conteúdo. O agrupamento ajudou a dividi-los tematicamente;
- O grupo 1 misturou boletos de cobrança com conversas de correio eletrônico, porque todos continham o nome completo do acusado e a palavra "advogado". Ou seja, embora houvesse palavras em comum, não havia semelhança semântica;
- Os grupos onde foram alocados os documentos das classes "ARQ. TEMPORÁRIO DE INTERNET" apresentavam uniformidade temática, mas os grupos que continham os documentos da classe "HISTÓRICO DE MSN" não apresentavam;
- Os documentos relevantes para a apuração estavam concentrados nos grupos 10, 11 e 12.

Tabela 5.27: Tabela de contingência do termo “Sobrenome” na evidência nº 3

Classes\Grupos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	Somas
1) ARQUIVOS TEMPORÁRIOS DE INTERNET				3		1	1	1	7				2									15 (2º)
2) AUTO DE DEFESA										1	1											2
3) AUTO DE RECURSO										1												1
4) BOLETO BANCÁRIO	3																					3
5) CERTIDÃO DE ANTECEDENTES					1																	1
6) CERTIDÃO NEGATIVA					1																	1
7) CONVERSA DE CORREIO ELETRÔNICO	2		2	2						2												8 (3º)
8) DECLARAÇÃO DE HISTÓRICO PROFISSIONAL		1																				1
9) DECLARAÇÃO DE POBREZA												1										1
10) FORM. DE ANTECIPAÇÃO DE BENEFÍCIO														1								1
11) FORM. DE REQUERIMENTO DE APOSENTADORIA		1																				1
12) HISTÓRICO DE MSN															3	4	3	4	4	4	3	25 (1º)
13) PROCURAÇÃO			2																			2
14) REQUERIMENTO											1											1
15) SENTENÇA JUDICIAL													1									1
Soma	5	2	4	5	2	1	1	1	7	4	2	1	3	1	3	4	3	4	4	4	3	64

Tabela 5.28: Dados dos grupos e protótipos para o termo “Sobrenome” na evidência nº 3

Grupo	#Total Termos	#Termos Prot.	#Docs	#DocsTE	#DocsTP	#DocsPL	#DocsML
1	147	9	5	2			3
2	85	5	2	2			
3	276	14	4	4			
4	353	21	5	2	3		
5	122	31	2	1			1
6	241	241	1		1		
7	308	308	1		1		
8	131	131	1		1		
9	218	85	7		7		
10	1276	92	4	4			
11	516	51	2	2			
12	60	60	1	1			
13	1022	45	3				3
14	182	182	1	1			
15	1141	45	3				3
16	4009	120	4				4
17	3206	155	3				3
18	7632	177	4				4
19	9038	266	4				4
20	13611	371	4				4
21	13085	582	3				3

5.4.3 EVIDÊNCIA Nº 3 - RESUMO E DISCUSSÃO

A Tabela 5.29 apresenta um resumo das buscas submetidas à evidência nº 3 e seus resultados.

Tabela 5.29: Tabela-resumo das buscas na evidência nº 3

Critério de busca	Resultados	Dimensões	Classes	Grupos	$\phi^{(NMI)}$	ρ
Docs. de texto estruturado	36	2.930	21	16	0,853	0,0867
Termo sobrenome	64	41.693	15	21	0,68	0,0678

Os resultados obtidos com a evidência nº 3 sugerem que o agrupamento poderia se beneficiar do processamento semântico dos resultados, em virtude de que documentos que possuíam palavras em comum mas não tinham semelhança semântica foram associados aos mesmos grupos.

A combinação dos resultados das análises quantitativas e qualitativas sugere que a abordagem proposta nesta dissertação e materializada através do protótipo é promissora.

5.5 DISCUSSÃO GERAL

Verificou-se em todos os testes experimentais realizados que o algoritmo conseguiu agrupar documentos cujos textos possuíam termos em comum. A granularidade dos grupos foi norteada pelo parâmetro de vigilância. Obteve-se $0,5 \leq \phi^{(NMI)} \leq 0,95$.

Para que os grupos se aproximem das classes, é necessário que seus protótipos contenham os termos dos documentos pertencentes às classes. Mas os protótipos não são especificados pelo usuário. Conforme apresentado na Seção 3.5, segundo Massey (2005a), as RNAs ART1 são sensíveis à ordem de apresentação dos padrões de entrada, e podem produzir soluções de agrupamento diferentes quando a ordem de apresentação dos padrões é modificada. Como os protótipos dos grupos são criados a partir dos padrões de entrada, uma mudança na ordem de apresentação dos padrões pode resultar na criação de protótipos diferentes. Não foi possível determinar a ordem de apresentação que induz o valor mais alto possível para a estimativa $\phi^{(NMI)}$.

Outro fator que possivelmente aumentaria o valor da estimativa $\phi^{(NMI)}$ seria a adaptação da lista de *stopwords* para que o índice somente armazenasse os termos de maior valor discriminante para os documentos, e dessa forma os protótipos não contivessem termos de menor valor discriminante. Essa adaptação, no entanto, não está no escopo desta dissertação, que se propõe unicamente a investigar a utilidade das RNAs ART1 para agrupar

documentos com conteúdos assemelhados retornados por buscas por palavras-chave em coleções textuais de Informática Forense.

Em todos os testes o algoritmo apresentou tempo de execução reduzido; a execução mais demorada consumiu cerca de 5 segundos. Esse aspecto é importante porque o perito pode, após submeter uma busca e seu respectivo agrupamento, fazer uma breve análise e considerar que os grupos gerados são demasiado genéricos ou refinados; o apelo da utilização do algoritmo para organizar os *hits* torna-se reduzido se uma nova execução, com um novo valor para o parâmetro de vigilância, não puder ser realizada de forma interativa com tempo de resposta aceitável. O critério de aceitabilidade do tempo de resposta depende do usuário e, portanto, é subjetivo.

Foi constatado que houve casos em que os documentos que o especialista associou à mesma classe tinham poucos termos em comum além do nome da classe; isso se verificou, por exemplo, na classe "PROPOSTA COMERCIAL" descrita na Tabela 5.5. Em muitos dos documentos pertencentes a essa classe nas várias tabelas de contingência, o termo "PROPOSTA" encontrava-se no início do arquivo, em posição de destaque, com fonte tipográfica formatada em negrito e/ou com tamanho maior que o restante do texto do documento. Buscadores de Internet atribuem pesos diferentes às várias regiões de uma página, conforme descrito na seção 3.1; uma tal abordagem pode ser utilizada em uma versão modificada do algoritmo que, ao compor os vetores de entrada, associe maior poder discriminante a esses termos mais destacados. Para fazê-lo, no entanto, seria necessário mudar a representação utilizada, que é uma matriz de incidência binária, conforme exposto na subseção 3.2.2, e utilizar outra técnica de agrupamento que não a de RNAs ART1, que somente tratam vetores de entrada binários.

O agrupamento descrito na seção 5.3, onde foram agrupados todos os documentos processados na mídia, sem a filtragem proporcionada por termos de busca, produziu o melhor valor para a estimativa $\phi^{(NMI)}$. Nesse caso em particular, muitos dos documentos eram bastante parecidos entre si e também muito diferentes dos demais, a quantidade de documentos processada era relativamente pequena (156 documentos ao todo) e a dimensionalidade total também era relativamente baixa (3.379 dimensões).

Um teste de agrupamento de todos os documentos foi realizado com a evidência nº 1, que continha 3.455 documentos com 39.905 dimensões ao todo; o agrupamento resultante foi processado em 53 minutos e 55 segundos, e os 373 grupos resultantes, obtidos com o valor de $0,053645$ calculado automaticamente para o parâmetro de vigilância, tinham protótipos com poucos termos (muitos com apenas um termo) e misturavam documentos

de usuário com arquivos instalados por aplicativos, com os quais pode-se supor que não teriam nenhuma semelhança. Uma possível interpretação é que a filtragem proporcionada pelos termos de busca, que não foram utilizados nesse caso, diminui a quantidade de documentos retornados, bem como evita que o algoritmo de agrupamento necessite comparar documentos bastante diferentes entre si.

Muitos dos documentos recuperados eram de formatos estruturados, e não de texto plano. O conteúdo textual desses arquivos não teria sido extraído durante o processo de indexação se o indexador tivesse ignorado o sistema de arquivos e não tivesse realizado o processo de extração de texto (*parsing*) exigido pelos formatos binários proprietários.

Uma forma, que não foi tentada, de filtrar os documentos que chegam ao algoritmo é, durante o pré-processamento, utilizar uma base de dados de *hashes* para descartar arquivos sabidamente ignoráveis que são copiados nas mídias pelos programas de instalação do sistema operacional e de aplicativos. Tal medida poderia reduzir a quantidade de arquivos a processar, o número de dimensões da coleção, e o tempo necessário à execução do algoritmo.

Para situações como as descritas na Seção 4.3.3, onde a quantidade de células não-zeradas na BOW ultrapassa 100.000 e o tempo de execução do algoritmo ultrapassa 1 minuto, uma possível abordagem seria a redução de dimensionalidade por corte de frequência; os termos mais e menos frequentes da coleção seriam cortados, conforme proposto por Luhn (1958).

A reclamação dos especialistas de que os grupos não possuíam rótulos descritivos, conforme descrito na subseção 5.2.6, sugere que rótulos mais descritivos que apenas números e listas de documentos podem ajudar no processo de revisão dos *hits*.

CAPÍTULO 6

CONCLUSÕES

Este capítulo elenca as contribuições e apresenta as conclusões desta dissertação. Sugestões de trabalhos futuros são apresentadas na Seção 6.1.

As contribuições deste trabalho de pesquisa consistem do atingimento de todos os objetivos elencados na Seção 1.2, quais sejam:

1. As limitações dos atuais métodos para revisão de *hits* em coleções textuais forenses foram mostradas na Seção 2.6;

2. Foi desenvolvido um método de pré-processamento para transformar os arquivos das coleções textuais em representações tratáveis pelas RNAs ART1, conforme relatado na Seção 4.1.1;

3. Foi desenvolvido um protótipo de aplicação de busca e agrupamento de *hits* com conteúdo assemelhado em coleções textuais forenses que utiliza o método proposto, conforme descrito na Seção 4.1.2;

4. Foram realizadas experimentações com o uso do protótipo e dados provenientes de casos reais para validar o método de agrupamento de conteúdos assemelhados em coleções textuais forenses, conforme apresentado no Capítulo 5;

5. Foram validados os resultados das experimentações através de parâmetros objetivos, além das intrínsecas características subjetivas de conhecimento dos especialistas, conforme relatado no Capítulo 5.

Outra contribuição foi a adaptação do algoritmo ART1 para reduzir seu tempo de execução ao agrupar padrões de entrada esparsos e de alta dimensionalidade, conforme descrito na Seção 4.3.3.

A aplicabilidade das RNAs ART1 para agrupar *hits* com conteúdo assemelhado retornados por buscas em coleções textuais forenses foi demonstrada. *Hits* com conteúdos assemelhados retornados por buscas em coleções textuais forenses foram agrupados com sucesso, com resultados promissores sob critérios de avaliação quantitativos e qualitativos. A técnica e o método materializados através do protótipo foram considerados úteis.

Os experimentos demonstraram que é possível gerar grupos de qualidade sem necessidade de especificar o valor do parâmetro de vigilância ρ da rede ART1. O algoritmo ART1 determina automaticamente o número de grupos com base no valor do parâmetro de vigilância e na estrutura dos dados. A especificação do valor do parâmetro de vigilância é opcional e sua semântica é simples. Essas características contrastam com os algoritmos k-médias e SOM, que exigem que o número desejado de grupos seja informado de antemão e não sugerem um valor. Beebe (2007) e Decherchi et al. (2009) tiveram que arbitrar esses valores em seus trabalhos, que foram descritos respectivamente nas Seções 4.6.1 e 4.6.2.

Também foi demonstrado que não é suficiente agrupar os resultados obtidos por buscas realizadas no nível físico do arquivo de imagem. Ignorar o sistema de arquivos e a estrutura presente em diversos formatos populares de documentos de usuário pode acarretar redução substancial do *recall*, algo inaceitável no domínio da Informática Forense. Essa é uma lacuna do trabalho de Beebe (2007), que foi descrito na Seção 4.6.1.

Foi constatado em vários dos resultados dos experimentos que os grupos tenderam a conter arquivos do mesmo tipo. Isso sugere que pode ser útil classificar os *hits* por tipo de arquivo antes de agrupá-los, em vez de agrupar todos os documentos sem fazer distinção do tipo de arquivo. A classificação por tipo de arquivo é trivial, não sendo necessário usar um algoritmo de agrupamento para fazê-lo.

Agrupar com base apenas nas palavras presentes e/ou ausentes nos documentos pode induzir o algoritmo a misturar documentos que não possuem nenhuma semelhança semântica. Isso sugere que o agrupamento poderia se beneficiar do processamento semântico dos resultados.

A adaptação do algoritmo ART1 para utilizar uma estrutura de dados que leva em conta a natureza esparsa dos vetores obtidos a partir de documentos de coleções textuais mostrou-se eficaz para reduzir seu tempo de execução. Os testes descritos na Seção 4.3.3 foram até 38 vezes mais rápidos com o algoritmo adaptado que com o algoritmo original.

A definição do valor padrão do parâmetro de vigilância (ρ) foi empírica. Como esse valor é fundamental para a formação dos protótipos dos grupos, o ideal seria que houvesse

um método científico comprovadamente eficaz para estimar um valor. Ainda assim, os resultados obtidos foram adequados.

Dado o grande esforço necessário para elaborar as partições de referência e tabular todos os dados produzidos nos testes, foram feitos testes com três imagens de mídias apreendidas de pequeno volume de dados. Faz-se necessário aprofundar o estudo com experimentações de maior escopo, conforme sugerido na Seção 6.1.

A tarefa de agrupar documentos de coleções textuais forenses é intrinsecamente difícil. As coleções apresentam inúmeros tipos diferentes de documentos estruturados e não-estruturados, com enorme variedade de conteúdo e volume textual.

Nada pode superar a revisão exaustiva, individual e rigorosa de cada resultado de busca, que se torna inviável devido à alta dimensionalidade dos dados e limitações de recursos. Contudo, foi demonstrado que é possível ao perito obter através do agrupamento uma visão geral do teor dos documentos retornados. Isso pode ser útil, por exemplo, quando o objetivo é obter rapidamente informações de inteligência que possam demonstrar a materialidade e/ou a autoria de uma conduta criminosa, bem como vínculos até então desconhecidos entre pessoas investigadas. Tais informações podem ser cruciais para embasar novas diligências, ou até mesmo pedidos de prisão, no interesse da Justiça.

6.1 TRABALHOS FUTUROS

Há diversas possibilidades para trabalhos futuros. Pode-se citar:

- A utilização de RNAs ART não-binárias, como Fuzzy ART (Carpenter et al., 1991) e ART 2 (Carpenter e Grossberg, 1987a), para o agrupamento de documentos representados por vetores não-binários, possivelmente com o uso da medida *Term Frequency–Inverse Document Frequency (tf-idf)* (Jones, 1972; Robertson, 2004) para representar os documentos;

- Agrupamento hierárquico de documentos através do encadeamento ou paralelização de redes ART1 independentes;

- Realização de experimentos com grandes volumes de dados, possivelmente através da junção de várias imagens de mídias apreendidas para formar uma única coleção;

- Realização de experimentos com coleções compostas exclusivamente de mensagens de correio eletrônico, a exemplo daqueles descritos por Decherchi et al. (2009);

- Comparação com outros algoritmos de agrupamento, tais como k-médias (Lloyd, 1982) e EM (*Expectation-Maximization*) (Dempster et al., 1977), aplicados às mesmas coleções e com o cálculo de índices externos e relativos;
- Aplicação de algoritmos da família ART com processamento distribuído, em ambientes de *grid* computacional ou redes *Peer-to-Peer*;
- Pesquisa sobre como gerar automaticamente rótulos descritivos para os grupos, ou mesmo pesquisar o uso de rótulos semânticos;
- Pesquisa sobre como melhorar a apresentação e a navegação dos grupos e resultados.

Enfim, há vários trabalhos futuros possíveis a partir da realização deste primeiro ensaio de uso de RNAs ART1 no domínio da Informática Forense, que foi apresentado nesta dissertação. Essas oportunidades indicam a viabilidade do tema de pesquisa com a abertura de desafios novos e interessantes no campo jovem e promissor da Informática Forense.

REFERÊNCIAS BIBLIOGRÁFICAS

AccessData (2011). *Forensic Toolkit® 3.4.1*. Disponível em: `http://accessdata.com/products/computer-forensics/ftk`, acesso em 29/10/2011.

Aldenderfer, M. S. e Blashfield, R. K. (1984). *Quantitative Applications in the Social Sciences 44: Cluster Analysis*, volume 31. Sage Publications, Beverly Hills.

Andrews, N. O. e Fox, E. A. (2007). Recent developments in document clustering. Technical Report 1000, Department of Computer Science, Virginia Tech. Disponível em: `http://eprints.cs.vt.edu/archive/00001000/`, acesso em 29/10/2011.

Arndt, H. (2011). *Universal Java Matrix Package (UJMP)*. Disponível em: `http://sourceforge.net/projects/ujmp/`, acesso em 29/10/2011.

Baeza-Yates, R. A. e Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Beebe, N. L. (2007). *Improving information retrieval effectiveness in digital forensic text string searches: clustering search results using self-organizing neural networks*. PhD thesis, Department of Information Systems and Technology Management, College of Business, The University of Texas at San Antonio. AAI3289195.

Beebe, N. L. e Clark, J. G. (2005). A hierarchical, objectives-based framework for the digital investigations process. *Digital Investigation*, 2(2):147–167.

Beebe, N. L. e Clark, J. G. (2007). Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results. *Digital Investigation*, 4(Supplement 1):49–54.

Beyer, K. S.; Goldstein, J.; Ramakrishnan, R. e Shaft, U. (1999). When is "nearest neighbor" meaningful? In *Proceedings of the 7th International Conference on Database Theory*, ICDT '99, pages 217–235, London, UK. Springer-Verlag.

Carpenter, G. A. e Grossberg, S. (1987a). ART 2: self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, 26(23):4919–4930.

Carpenter, G. A. e Grossberg, S. (1987b). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37:54–115.

Carpenter, G. A.; Grossberg, S. e Rosen, D. B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771.

Carrier, B. (2011). *The Sleuth Kit 3.2.2*. Disponível em: `http://www.sleuthkit.org/`, acesso em 29/10/2011.

Carvey, H. (2009). *Windows Forensic Analysis DVD Toolkit*. Syngress Media. Syngress Pub.

Casey, E. (2007). What does "forensically sound" really mean? *Digital Investigation*, 4(2):49–50.

Cohen, W. W. (2009). *Enron Email Dataset*. Disponível em: `http://www-2.cs.cmu.edu/~enron/`, acesso em 29/10/2011.

Cover, T. M. e Thomas, J. A. (2006). *Elements of Information Theory*. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience.

de Araújo, G. R. F. e Ralha, C. G. (2011). Computer forensic document clustering with ART1 neural networks. In *Proceedings of The Sixth International Conference on Forensic Computer Science (ICoFCS) 2011, Florianópolis, Brasil*, pages 106–114. DOI: `http://dx.doi.org/10.5769/C2011011`.

Decherchi, S.; Tacconi, S.; Redi, J.; Leoncini, A.; Sangiacomo, F. e Zunino, R. (2009). Text clustering for digital forensics analysis. In Herrero, l.; Gastaldo, P.; Zunino, R. e Corchado, E., editors, *Computational Intelligence in Security for Information Systems*, volume 63 of *Advances in Intelligent and Soft Computing*, pages 29–36. Springer Berlin / Heidelberg.

Dempster, A. P.; Laird, N. M. e Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Farmer, D. e Venema, W. (2005). *Forensic discovery*. Addison-Wesley.

Fausett, L. (1994). *Fundamentals of neural networks: architectures, algorithms, and applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Forte, D. (2004). The importance of text searches in digital forensics. *Network Security*, 2004(4):13–15.

Ghosh, J. (2003). Scalable clustering. In Ye, N., editor, *Handbook of Data Mining*, chapter 10, pages 247–277. Lawrence Erlbaum Associates.

Grossberg, S. (1976). Adaptive pattern classification and universal recoding: I. parallel development and coding of neural feature detectors. *Biological Cybernetics*, 23:121–134. 10.1007/BF00344744.

Guidance (2011). *EnCase® Forensic 7.01.02*. Disponível em: `http://www.guidancesoftware.com/forensic.htm`, acesso em 29/10/2011.

Han, J. e Kamber, M. (2006). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Hoelz, B. W. P. (2009). MADIK Uma abordagem multiagente para o exame pericial de sistemas computacionais. Master's thesis, Universidade de Brasília, Instituto de Ciências Exatas, Departamento de Ciência da Computação.

Hotho, A.; Nürnberger, A. e Paaß, G. (2005). A brief survey of text mining. *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, 20(1):19–62.

Hudík, T. (2009). *Learning algorithms in processing of various difficult medical and environmental data*. PhD thesis, Faculty of Informatics, Masaryk University.

Hudík, T. (2011). *ART software package*. Disponível em: `http://users.visualserver.org/xhudik/art/`, acesso em 29/10/2011.

Huebner, E.; Bem, D. e Bem, O. (2008). Computer forensics - past, present and future. *Journal of Information Science and Technology*, 5(3):43–59.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666.

Jain, A. K. e Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice Hall, Upper Saddle River, NJ, USA.

Johansson, C. (2003). Computer forensic text analysis with open source software. Master's thesis, Dept. of Software Engineering and Computer Science, Blekinge Tekniska Högskola.

Johnson, R. E. e Foote, B. (1988). Designing reusable classes. *Journal of Object-Oriented Programming*, 1(2):22–35.

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

Kohonen, T. (1981). Automatic formation of topological maps of patterns in a self-organizing system. In Oja, E. e Simula, O., editors, *Proc. 2SCIA, Scand. Conf. on Image Analysis*, pages 214–220, Helsinki, Finland. Suomen Hahmontunnistustutkimuksen Seura r. y.

Kriegel, H.-P.; Kröger, P. e Zimek, A. (2009). Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*, 3:1:1–1:58.

Kuncik, N. A. (2010). Introducing data mining to digital forensic investigation process. Master's thesis, UCD School of Computer Science and Informatics, College of Engineering Mathematical and Physical Sciences, University College Dublin, Ireland.

Lessing, M. e von Solms, B. (2008). Live forensic acquisition as alternative to traditional forensic process. In *IT-Incidents Management & IT-Forensics - IMF 2008, Conference Proceedings*, pages 107–124.

Leuski, A. (2001). Evaluating document clustering for interactive information retrieval. In *Proceedings of the tenth international conference on Information and knowledge management*, CIKM '01, pages 33–40, New York, NY, USA. ACM.

Leuski, A. e Allan, J. (2000). Improving interactive retrieval by combining ranked lists and clustering. In *IN PROCEEDINGS OF RIAO'2000*, pages 665–681.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2:159–165.

Manning, C. D.; Raghavan, P. e Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Massey, L. (2002). Determination of clustering tendency with ART neural networks. In *Proceedings of the 4th International Conference on Recent Advances in Soft Computing*.

Massey, L. (2003). On the quality of ART1 text clustering. *Neural Networks*, 16:771–778.

Massey, L. (2005a). *Le groupage de texte avec les réseaux de neurones ART1*. PhD thesis, Faculté d'ingénierie du Collège militaire royal du Canada.

Massey, L. (2005b). Real-world text clustering with Adaptive Resonance Theory neural networks. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN) 2005*, volume 5, pages 2748–2753.

Mattmann, C. A. e Zitting, J. L. (2011). *Tika in Action*. Manning Publications Co., Greenwich, CT, USA.

McCandless, M.; Hatcher, E. e Gospodnetić, O. (2010). *Lucene in Action, Second Edition*. Manning Publications Co., Greenwich, CT, USA.

McCulloch, W. S. e Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5(4):115–133.

McKemmish, R. (1999). What is forensic computing? *Trends and Issues in Crime and Criminal Justice*, 118:1–6.

Microsoft (2011). *Windows Live Messenger*. Disponível em: `http://explore.live.com/windows-live-messenger`, acesso em 23/10/2011.

Moore, B. (1988). ART1 and pattern clustering. In Touretzky, D.; Hinton, G. e Sejnowski, T., editors, *Proceedings of the 1988 Connectionist Models Summer School*, pages 174–185, San Mateo, Pittsburgh. Morgan Kaufmann.

Page, L.; Brin, S.; Motwani, R. e Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Disponível em: `http://ilpubs.stanford.edu:8090/422/`, acesso em 29/10/2011.

Palmer, G. (2001). A road map for digital forensic research. Technical Report DTR - T001-01 FINAL, Digital Forensic Research Workshop. Disponível em: `http://www.dfrws.org/2001/dfrws-rm-final.pdf`, acesso em 29/10/2011.

Prosise, C. e Mandia, K. (2003). *Incident response and computer forensics*. Security Series. McGraw-Hill/Osborne.

Robertson, S. (2004). Understanding inverse document frequency: On theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520.

Roussinov, D. G. e Chen, H. (1998). A scalable self-organizing map algorithm for textual classification: A neural network approach to thesaurus generation. *Communication Cognition and Artificial Intelligence, Spring*, 15:81–112.

Russell, S. e Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition.

Soares, M. V.; Prati, R. C. e Monard, M. C. (2008). PreTexT II: Descrição da Reestruturação da Ferramenta de Pré-Processamento de Textos. Relatório Técnico 333, ICMC-USP, São Carlos, Brasil. Disponível em: `http://www.icmc.usp.br/~biblio/BIBLIOTECA/rel_tec/RT_333.pdf`, acesso em 29/10/2011.

Stoer, J. e Bulirsch, R. (2002). *Introduction to Numerical Analysis*. Texts in applied mathematics. Springer.

Strehl, A.; Ghosh, J. e Cardie, C. (2002). Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617.

Tukey, J. W. (1980). We need both exploratory and confirmatory. *The American Statistician*, 34(1):23–25.

van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworths, London, 2nd edition.

Vinh, N. X.; Epps, J. e Bailey, J. (2009). Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1073–1080. ACM.

Whitcomb, C. M. (2002). An historical perspective of digital evidence: A forensic scientist's view. *International Journal of Digital Evidence*, 1(1):1–9.

Zeng, H.-J.; He, Q.-C.; Chen, Z.; Ma, W.-Y. e Ma, J. (2004). Learning to cluster web search results. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 210–217, New York, NY, USA. ACM.