UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

CÁSSIO OLIVEIRA CAMILO

Uma Metodologia para Mineração de Regras de Associação Usando Ontologias para Integração de Dados Estruturados e Não-Estruturados

Goiânia
2010

CÁSSIO OLIVEIRA CAMILO

Uma Metodologia para Mineração de Regras de Associação Usando Ontologias para Integração de Dados Estruturados e Não-Estruturados

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

Área de concentração: Sistemas de Informação.

Orientador: Prof. Dr. João Carlos da Silva

Goiânia
2010

CÁSSIO OLIVEIRA CAMILO

Uma Metodologia para Mineração de Regras de Associação Usando Ontologias para Integração de Dados Estruturados e Não-Estruturados

Dissertação defendida no Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Mestre em Ciência da Computação, aprovada em 23 de Agosto de 2010, pela Banca Examinadora constituída pelos professores:

Prof. Dr. João Carlos da Silva
Instituto de Informática – UFG
Presidente da Banca

Prof. Dr. Cedric Luiz de Carvalho
Instituto de Informática – UFG

Profa. Dra. Maria Luiza Machado Campos
Departamento de Ciência da Computação – UFRJ

Cássio Oliveira Camilo

Graduou–se em Ciência da Computação pela PUC-Goiás (Pontifícia Universidade Católica de Goiás). Gestor de Tecnologia da Informação, lotado na Secretaria de Segurança Pública do Estado de Goiás desde 2007 atuando junto a equipe da Assessoria de Informática e Telecomunicação.

À Deus pela vida e oportunidades.
A minha esposa, aos meus familiares e amigos.

Agradecimentos

Agradeço a Deus pelas graças alcançadas.

Ao Prof. Dr. João Carlos da Silva, pela orientação, dedicação e comprometimento durante todo o período do mestrado.

A todos os servidores do Instituto de Informática da Universidade Federal de Goiás, em especial ao Prof. Dr. Cedric Luiz de Carvalho, pelas contribuições e sugestões.

À Kenia, minha esposa, pela compreensão, paciência e, acima de tudo, amor durante os muitos momentos de ausência. Seu apoio foi fundamental.

À equipe da Secretaria de Segurança Pública do Estado de Goiás pela presteza ao disponibilizar as informações necessárias à realização deste trabalho.

A toda minha família e amigos pelos incentivos.

“Veni, vidi, vici”.

Júlio César,
General Romano.

Resumo

Camilo, Cássio Oliveira. **Uma Metodologia para Mineração de Regras de Associação Usando Ontologias para Integração de Dados Estruturados e Não-Estruturados**. Goiânia, 2010. 148p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

Métodos de mineração de dados e mineração de textos têm sido aplicados em diversas áreas do conhecimento para recuperação de informações úteis a partir de grandes volumes de dados. Dentre os diversos métodos de mineração de dados propostos na literatura, a mineração de regras de associação tem sido de grande utilidade. Entretanto, um dos grandes problemas gerados pela aplicação deste método sobre um grande volume de dados é, em geral, a produção de uma quantidade significativa de regras, dificultando a escolha daquelas mais relevantes para responder a uma consulta. O presente trabalho propõe uma metodologia para minerar dados de fontes estruturadas e não estruturadas, visando gerar regras de associação entre termos extraídos dessas fontes. O processo de mineração de dados de fontes não-estruturadas é auxiliado por uma Ontologia para mapear conhecimentos de um domínio específico. O resultado desta etapa é convertido para uma representação estruturada, e é então combinado com os dados obtidos de outras fontes estruturadas. Além do modelo de suporte e confiança, utiliza-se uma combinação das medidas de interesse objetivas e subjetivas para filtrar o conjunto de regras obtido. Para analisar sua viabilidade em situações reais, a metodologia proposta neste trabalho foi submetida à aplicação de ocorrências policiais de uma instituição governamental, sob conjuntos de dados armazenados em fontes estruturadas e não estruturadas.

Palavras–chave

Mineração de Dados, Mineração de Texto, Recuperação de Informação, Extração de Informação, Conceitos, Ontologia, Regras de Associação

Abstract

Camilo, Cássio Oliveira. **A Methodology for Mining Association Rules Using Ontologies for Integrating Structured and Non-Structured Data**. Goiânia, 2010. 148p. MSc. Dissertation. Instituto de Informática, Universidade Federal de Goiás.

Data and text mining methods have been applied in several areas of knowledge with the purpose of extracting useful information from large data volumes. Among the various data mining methods reported by specialized literature, association rule mining has proved useful in producing understandable rules. However, one of its major problems is the significant amount of rules produced, which hampers the selection of the more relevant rules needed to reply to a query. This study proposes a method for mining data from structured and unstructured sources in order to generate association rules between the terms extracted. The process of mining data from unstructured sources is assisted by an ontology that maps knowledge from a specific domain. The result of such process is converted into structured data and combined with data from other structured sources. A combination of objective and subjective interest measures is used to filter the set of rules obtained, in addition to support and confidence model. To verify the feasibility of this method in real-life situations, it was applied to a database of police occurrence reports of a government institution, which included data stored in structured and unstructured sources.

Keywords

Data mining, Text mining, Information Retrievel, Information Extraction, Concept, Ontology, Association Rules

Sumário

Lista de Figuras **12**

Lista de Tabelas **14**

1 Introdução **15**
- 1.1 Motivação 16
- 1.2 Objetivos 18
- 1.3 Organização da Dissertação 19

2 Fundamentação Teórica **20**
- 2.1 Descoberta de Conhecimento 20
  - 2.1.1 Descoberta do Conhecimento em Dados Estruturados 21
  - 2.1.2 Descoberta do Conhecimento em Dados Não-Estruturados 22
- 2.2 Mineração de Dados 23
  - 2.2.1 Tarefas 24
  - 2.2.2 Dados 25
- 2.3 Mineração de Textos 29
- 2.4 Recuperação de Informação 30
  - 2.4.1 Processo de Indexação 30
  - 2.4.2 Modelos de Recuperação de Informação 31
    - 2.4.2.1 Modelo Booleano 32
    - 2.4.2.2 Modelo Vetorial 32
    - 2.4.2.3 Modelo Probabilístico 33
  - 2.4.3 Medidas de Eficácia 34
- 2.5 Extração de Informação - EI 34
- 2.6 Ontologia 35
  - 2.6.1 Tipos de Ontologias 36
  - 2.6.2 Linguagem 36
  - 2.6.3 Metodologias 37
- 2.7 Análise Criminal 38

3 Ontologias, Mineração de Dados e Textos **41**
- 3.1 Mineração de Dados apoiada por Ontologia 41
- 3.2 Padrão OWL 42
  - 3.2.1 Elementos Básicos 42
    - 3.2.1.1 Classes 43
    - 3.2.1.2 Instâncias 43
    - 3.2.1.3 Propriedades 43
- 3.3 Definição de Conceitos 44

3.4 Cálculo de Similaridade entre Ontologia e Texto 45
3.4.1 Medidas de distância 46
3.4.2 Coeficiente de Associação 47

4 Regras de Associação **49**
4.1 Mineração de Itens Frequentes 49
4.2 Definição Formal 50
4.3 Algoritmo Apriori 51
4.3.1 Etapas do algoritmo *Apriori* 51
4.3.2 Algoritmo *Apriori* 54
4.4 Medidas de Interesse 54
4.4.1 Medidas de Interesse Objetivas 54
4.4.1.1 Modelo Suporte e Confiança 54
4.4.1.2 *Lift* 55
4.4.1.3 Novidade 55
4.4.1.4 Convicção 55
4.4.2 Medidas de Interesse Subjetivas 56
4.4.2.1 Impressão Geral 57
4.4.2.2 Conhecimento Impreciso 58

5 Metodologia Proposta **59**
5.1 Trabalhos Relacionados 59
5.1.1 *Interestingness Analysis System - IAS* 59
5.1.2 Extração de regras de associação de dados textuais 60
5.1.3 Utilização de uma Ontologia na melhora do valor de suporte 60
5.1.4 RulEE-SEAR 61
5.1.5 Mineração em dados estruturados e não-estruturados 61
5.1.6 Classificação de jurisprudências 62
5.1.7 Análise de diagnósticos médicos 62
5.2 Metodologia Proposta 62
5.2.1 Preparação do ambiente 65
5.2.2 Processamento 66
5.2.2.1 Extração de conceitos 66
5.2.2.2 Filtragem das regras 69

6 Desenvolvimento do Sistema **71**
6.1 Estudo de Caso 71
6.2 Ferramentas Utilizadas 77
6.2.1 WEKA 77
6.2.2 Protégé 77
6.2.3 *Framework* Jena 78
6.2.4 Ferramenta de ETL *Kettle* 79
6.3 Funcionamento do Sistema 79
6.3.1 Etapa: Preparação do Ambiente 80
6.3.2 Etapa: Processamento 81
6.3.2.1 Extração de conceitos 86
6.3.2.2 Filtragem das regras 88

7 Resultados **92**
7.1 Contextualização 92
7.2 Etapa: Preparação do Ambiente 93
7.3 Etapa: Processamento 93
7.3.1 Extração de conceitos nos textos 93
7.3.2 Filtragem das regras 98

8 Conclusões **103**
8.1 Contribuições 105
8.2 Produções Bibliográficas 105
8.2.1 Artigos Publicados 105
8.2.1.1 Recuperação Contextualizada de Documentos em Bibliotecas Digitais Integradas [74] 105
8.2.2 Relatórios Técnicos 106
8.2.2.1 Mineração de Dados: Conceitos, Tarefas, Métodos e Ferramentas 106
8.2.2.2 Um estudo sobre a interação entre Mineração de Dados e Ontologias 106
8.3 Trabalhos Futuros 106

Referências Bibliográficas **108**

A Arquivo OWL da Ontologia de domínio criada **117**

B Códigos Java utilizado **135**
B.1 Método para realizar o *stemming* 135
B.2 Método para recuperar os conceitos da Ontologia 136
B.3 Método utilizado para cálculo da similaridade 140
B.4 Método utilizado para geração e filtragem das regras 142

Lista de Figuras

2.1 Abordagens do processo de Descoberta de Conhecimento 21
2.2 Visão geral do processo de KDD 22
2.3 Visão geral do processo de KDT 23
2.4 Evolução das técnicas de visualização 26
2.5 Atividades do pré-processamento 27
2.6 Etapas da mineração de textos 29
2.7 Estrutura do Índice Invertido 31
2.8 Cosseno θ entre o documento d_1 e a consulta q 33
2.9 Processo de Extração de Informação 34

3.1 *Framework* geral para a integração entre Mineração e Ontologia 42
3.2 Exemplo de um Conceito com seus termos relacionados 45
3.3 Estrutura de um Conceito 45
3.4 Processo de cálculo de similaridade entre texto e Ontologia 46
3.5 Espaço euclidiano com os pontos 47

4.1 Regra de associação 50

5.1 Fluxo geral da metodologia proposta 64
5.2 Visão geral da metodologia proposta 65
5.3 Etapa: Preparação do ambiente 66
5.4 Etapa: Processamento 67

6.1 Tela inicial do sistema SPP 72
6.2 Tela de cadastro de um boletim de ocorrência no sistema SPP 73
6.3 Exemplo do Boletim de Ocorrência contendo a parte estruturada (dados do fato, vítima) e não-estruturada (histórico) 74
6.4 Visão geral do processo de análise 75
6.5 Ferramenta WEKA 77
6.6 Ferramenta Protégé mostrando a hierarquia de uma Ontologia 78
6.7 Ferramenta Kettle para Integração de Dados 79
6.8 Fluxo criado utilizando a ferramenta *Kettle* 80
6.9 Tela inicial do sistema proposto 81
6.10 Visão Geral da Ontologia “Homicídio Doloso” criada na ferramenta Protégé 82
6.11 Conceito “ArmadeFogo” e seus termos 83
6.12 Exemplo do conceito “Faca” e seus termos 83
6.13 Exemplo do relacionamento temUm do conceito “Homicídio” 84
6.14 Diagrama de Classes do sistema proposto 85
6.15 Diagrama de sequência para funcionalidade “Gerar Regras” 86

6.16 Tela utilizada para definição das medidas objetivas 89
6.17 Tela utilizada para definição das medidas subjetivas 90
6.18 Regras produzidas 91

7.1 Resultado geral da extração de informações 96
7.2 Resultado da segunda validação 98
7.3 Comparativo entre as regras geradas e as realmente úteis utilizando apenas as medidas de interesse objetivas 100
7.4 Comparativo entre as regras geradas e as realmente úteis utilizando a combinação de medidas de interesse objetivas e subjetivas 101

Lista de Tabelas

4.1 Exemplo de transações 52
4.2 1-itemsets 52
4.3 2-itemsets 52
4.4 3-itemsets 53

5.1 Vetor de termos do documento 69
5.2 Vetor de termos do conceito “ArmaDeFogo” 69
5.3 Vetor de termos do conceito “Homicídio” 69
5.4 Vetor de termos do conceito “Vingança” 69

6.1 Atributos utilizados no processo de geração das regras 76
6.2 Estrutura do arquivo ARFF 78
6.3 Conceitos e Similaridades obtidas com o processamento do histórico da ocorrência 88
6.4 Valores das medidas de interesse objetivas 89
6.5 Valores das medidas de interesse subjetivas 90

7.1 Grupo 2: conceitos identificados pela metodologia 95
7.2 Grupo 2: conceitos identificados pela análise do especialista 95
7.3 Grupo 3: conceitos identificados pela metodologia 96
7.4 Grupo 3: conceitos identificados pela análise do especialista 96
7.5 Quantidade de ocorrências 97
7.6 Configuração das medidas de interesse subjetivas 99
7.7 Valores das medidas de interesse objetivas 100
7.8 Conjunto de Regras Geradas 102

CAPÍTULO 1

Introdução

Com o crescimento exponencial da capacidade de gerar, coletar e armazenar dados, impulsionado por questões como o barateamento dos componentes computacionais, as exigências científicas (projetos em bioinformática e nanotecnologias) e sociais (necessidades de monitoramento e rastreamento de informações), as instituições enfrentam dificuldades em processar e analisar estes dados [117].

As organizações perceberam que não bastaria somente armazenar os dados. Era preciso tratá-los, de forma que fosse possível extrair deles informações úteis. Entretanto, devido ao grande volume de dados, o processo de análise, geralmente manual, tornou-se impraticável, demandando novas tecnologias capazes de automatizar este processo.

As técnicas tradicionais de pesquisa, recuperação e tratamento dos dados em sistemas convencionais, como a linguagem SQL (*Structured Query Language*) [78], as ferramentas OLAP (*Online Analytical Processing*) [42] ou os mecanismos de visualização de dados [67], encontram dificuldades em atender as novas necessidades dos usuários. Neste contexto, surge a Mineração de Dados, com o intuito de sanar essas necessidades.

O objetivo da mineração de dados é encontrar conhecimento útil a partir de um conjunto de dados, de forma que este conhecimento possa ser utilizado na tomada de decisão. O conhecimento descoberto deve ser compreensível, interessante e interpretável pelos usuários. Entretanto, uma das dificuldades na utilização de técnicas de mineração de dados é justamente compreender os modelos gerados pelas técnicas [43].

Neste sentido, a utilização de modelos baseados em regras tem auxiliado os usuários a interpretarem os resultados produzidos. A técnica de Mineração de Regras de Associação é aplicada para identificar relacionamentos e padrões implícitos contidos nos repositórios de dados [1]. Porém, o número de regras geradas em geral é extremamente alto, e muitas vezes elas não revelam padrões interessantes. Faz-se necessário, assim, identificar meios de melhorar a geração e filtragem das regras obtidas [103].

Com o intuito de melhorar a qualidade final das regras geradas durante o processo de mineração utilizam-se as medidas de interesses, que visam incorporar o conhecimento do usuário na interpretação das regras, eliminando as menos interessantes. As medidas são divididas em dois grupos: objetivas e subjetivas [99]. As medidas de interesse objetivas

baseiam-se em valores estatísticos e as medidas de interesse subjetivas utilizam-se do conhecimento prévio dos especialistas.

Outro aspecto que dificulta a extração de informações de grandes volumes de dados deve-se ao fato de que em média 80% dos dados de uma instituição encontram-se armazenados de maneira não-estruturada (textual) [102]. Como o processo de mineração de dados necessita que os dados estejam armazenados de uma forma estruturada (tabular), faz-se necessário processar os dados não-estruturados, transformando-os em estruturados. Realiza-se este processo através da Mineração de Texto.

Diversas linhas de pesquisa concentram-se no processamento dos dados não-estruturados, dentre as quais podem ser destacadas: Processamento de Linguagem Natural (PLN), Recuperação de Informação (RI) e Extração de Informação (EI). De maneira geral, as propostas concentram-se na interpretação dos termos dos textos, de forma que as palavras relevantes sejam identificadas, obtidas e possam representar o documento de origem.

É possível perceber ainda que, independente da técnica utilizada, é necessário representar e utilizar o conhecimento prévio de especialistas sobre determinado domínio para que o processo de interpretação dos textos possa ser otimizado. As Ontologias têm sido utilizadas neste sentido, possibilitando o mapeamento do conhecimento prévio em uma linguagem que permita aos programas de computador interpretar esse conhecimento, viabilizando a interação entre homem-máquina e máquina-máquina [68].

Assim, o presente trabalho propõe uma metodologia para automatizar o processo de mineração de regras de associação em dados estruturados e não-estruturados, utilizando-se de uma Ontologia para interpretação de conceitos contidos nos dados não-estruturados, possibilitando assim a integração entre os diferentes tipos de dados. A proposta utiliza-se da combinação de um conjunto de medidas de interesse objetivas e subjetivas para filtragem das regras menos interessantes.

1.1 Motivação

Diante dos grandes desafios impostos para a automatização do processo de análise dos dados, aliado ao fato do crescente uso de estruturas textuais para armazenar informações, é latente a necessidade de métodos que possam contribuir com o usuário para um tratamento efetivo do grande volume de dados armazenados.

As instituições perceberam que não basta apenas armazenar os dados, é preciso tratá-los e utilizá-los como subsídio para a tomada de decisão. Em um cenário que geralmente envolve diversos bancos de dados, arquivos de textos, *sites* pessoais, dentre tantas outras formas de armazenamento, a análise desses dados torna-se crítica, complexa e onerosa para ser realizada de forma manual.

É cada vez mais necessário o uso de ferramentas que integrem de forma automática essas diversas fontes, consolidando-as em uma estrutura padronizada, a partir da qual seja possível realizar consultas e inferências. Este processo, apesar de parecer simples, envolve diversas etapas que precisam ser consideradas, tais como: limpeza dos dados, evitando que valores errados sejam considerados durante a análise; padronização dos valores, uma vez que o mesmo dado pode ter um valor em uma fonte e outro valor em outra; transformação de alguns dados simples em dados derivados, evitando um número excessivo de variáveis; entre outras.

Devido ao grande volume de informações armazenadas em formato não-estruturado, o processo de análise e extração de informações a partir de dados textuais é cada vez mais necessário. Porém, devido ao grande número de variáveis que devem ser consideradas durante o processo de análise, tais como forma da escrita, linguagem utilizada, questões regionais, formato etc., esta atividade torna-se não trivial.

São exigidos mecanismos automatizados para que se possa extrair dos textos as informações relevantes para uma determinada situação. Surge a necessidade de mapear o conhecimento de um especialista sobre um determinado domínio para que então seja possível uma comunicação automatizada homem-máquina. É necessário que a máquina consiga "entender" os conceitos utilizados pelo usuário e então recuperá-los nos textos, como em um processo de interpretação humana.

Percebe-se, assim, a necessidade de combinar, durante o processo de análise, os dados obtidos das diversas fontes, quer estejam em formatos estruturados ou não-estruturados. Por exemplo, as informações estatísticas das instituições devem ser combinadas com os relatórios feitos por especialistas. A ideia é buscar os dados, não importando o formato, combiná-los, processá-los e extrair deles informações.

Além do processamento das informações contidas em dados textuais, outra situação que tem demandado esforços consideráveis é o descobrimento de padrões implícitos nos mesmos, porque, devido ao grande volume de dados disponíveis, torna-se impossível analisá-los de maneira manual ou utilizando técnicas tradicionais de exploração.

A descoberta de conhecimento é a chave para este processo. Dentre as diversas técnicas existentes, destaca-se a mineração de dados. Com esta técnica é possível, de forma automática, processar um grande volume de dados em busca de padrões implícitos que sejam relevantes para o usuário. Com isso, surge outro desafio: tornar o resultado obtido por estas técnicas utilizável e interpretável para os usuários. Há necessidade de criar estruturas que sejam facilmente reconhecidas e que permitam a extração de conhecimentos que auxiliem na tomada de decisão.

É preciso permitir que o usuário seja capaz que interagir com as técnicas de descoberta de conhecimento, possibilitando-o eliminar resultados irrelevantes e óbvios. Devido ao grande número de padrões descobertos, uma análise geral também se torna

inviável. É necessário permitir, então, que o usuário transmita seu conhecimento sobre o domínio de forma a filtrar os resultados.

De forma complementar a todas estas necessidades, existe também a exigência por ferramentas que consigam automatizar o máximo possível, de maneira confiável e segura, estes processos.

Assim, estas necessidades por métodos e ferramentas que auxiliem na resolução dos problemas descritos são os fatores motivadores deste trabalho, destacando-se principalmente a extração automática de informações de dados não-estruturados e a busca da melhoria dos resultados obtidos com as técnicas de descoberta de conhecimento.

1.2 Objetivos

Propor uma metodologia e uma ferramenta que consiga automatizar o processo de geração de padrões úteis para dados obtidos de diversas fontes (base de dados relacionais, arquivos de textos, *data warehouse*[1]) e em diversos formatos (estruturados e não-estruturados) é o foco deste trabalho.

Pretende-se sistematizar a busca por dados em diversos formatos (estruturados e não-estruturados), consolidando-os em um repositório central, para, em seguida, com o uso do conhecimento de especialistas, extrair conceitos relevantes presentes nos dados não-estruturados, combiná-los com os demais dados estruturados e extrair padrões úteis implícitos nos mesmos.

A ferramenta criada será utilizada em uma instituição governamental para auxiliar seus servidores no processo de análise dos dados.

O desenvolvimento da metodologia e da ferramenta tem como objetivos:

- Sistematizar um processo que permita a geração de padrões úteis a partir de dados estruturados e não-estruturados;
- Possibilitar a extração de informações de dados textuais através da utilização do conhecimento fornecido pelos especialistas;
- Permitir a interação do usuário com os padrões obtidos possibilitando eliminar dados inúteis;
- Contribuir com estudos posteriores nas áreas de Descoberta de Conhecimento, Ontologias e Mineração de Regras de Associação;
- Auxiliar os servidores da instituição governamental utilizada como estudo de caso no processo de análise dos dados.

[1] Um data warehouse é utilizado para armazenar informações relativas às atividades de uma organização em bancos de dados, de forma consolidada.

1.3 Organização da Dissertação

O presente trabalho, além deste Capítulo 1, está organizado como descrito a seguir.

No Capítulo 2 é feita a descrição das principais tecnologias utilizadas neste trabalho. São abordados conceitos tais como o processo de descoberta de conhecimento os fundamentos da mineração de dados e textos, a teoria básica sobre recuperação de informação e extração da informação, os conceitos essenciais de ontologias e, por fim,há uma visão geral do processo de análise criminal, utilizado como motivação para o nosso estudo de caso.

No Capítulo 3, é abordado de forma mais detalhada o processo de integração entre a mineração e o uso de Ontologias, e são descritos a linguagem OWL e seus principais conceitos, a representação do conhecimento através de "Conceitos", e, por fim, os cálculos de similaridades utilizados.

Já no Capítulo 4, são abordadas as regras de associação, contemplando a definição formal e o algoritmo Apriori. Por fim, são abordadas as medidas de interesse objetivas e subjetivas, incluindo as fórmulas utilizadas para o cálculo.

O Capítulo 5 aborda a metodologia proposta de maneira detalhada. Além disto, os trabalhos relacionados são descritos.

No Capítulo 6, é mostrado em detalhes o desenvolvimento do sistema utilizado para validação da metodologia proposta.

No Capítulo 7, são apresentados os resultados da aplicação do sistema desenvolvido nos dados da Secretaria de Segurança Pública do Estado de Goiás.

No Capítulo 8, são apresentadas as conclusões e perspectivas futuras.

Ao final, no Apêndice A é apresentada a Ontologia no formato do arquivo OWL criado neste trabalho, e no Apêndice B são apresentados os principais trechos dos códigos utilizados no desenvolvimento da ferramenta proposta.

CAPÍTULO 2

Fundamentação Teórica

Este capítulo contemplará os conceitos básicos necessários para compreensão do trabalho proposto. Na Seção 2.1, será discutido o processo da descoberta de conhecimento e suas formas. A Seção 2.2 apresentará conceitos envolvidos na mineração de dados, destacando-se as tarefas existentes e a correta análise dos dados. A Seção 2.3 abordará a disciplina de mineração de textos. Na Seção 2.4, serão apresentados os fundamentos básicos do processo de recuperação de informação. A Seção 2.5 apresentará o conceito da extração de informação. As ontologias serão abordadas na Seção 2.6. Por fim, a Seção 2.7 apresentará conceitos concernentes ao estudo de caso utilizado.

2.1 Descoberta de Conhecimento

O processo da descoberta de conhecimento consiste em tratar os dados brutos provenientes de diferentes fontes, de forma que se obtenha a informação e, a partir desta, seja possível gerar o conhecimento. Nos dias atuais, devido ao grande volume de informações disponíveis, o tratamento manual dos dados tornou-se uma atividade inviável [113].

Surge, então, o processamento automatizado dos dados através do computador, chamado Descoberta de Conhecimento, do inglês *Knowlegde Discovery - KD*. Como mostra a Figura 2.1, existem duas principais abordagens para o processo de descoberta de conhecimento: uma baseada em dados estruturados, chamado *Knowlegde Discovery in Databases - KDD* e outra baseada em dados não-estruturados, chamada *Knowlegde Discovery from Text - KDT*.

A seguir, serão analisadas abordagens de descoberta de conhecimento em dados estruturados e não-estruturados.

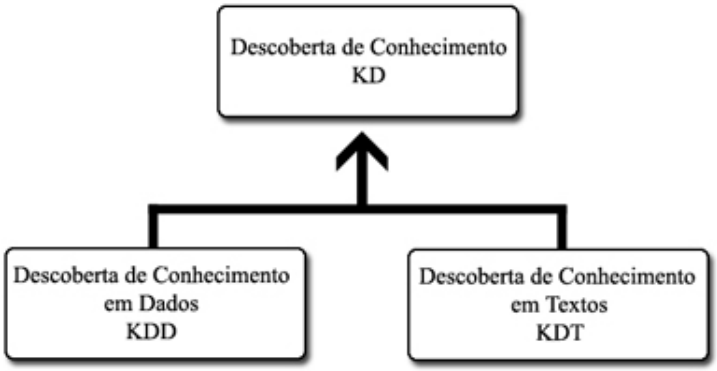


Figura 2.1: *Abordagens do processo de Descoberta de Conhecimento*

2.1.1 Descoberta do Conhecimento em Dados Estruturados

As primeiras aplicações da descoberta de conhecimento em dados estruturados se utilizaram da tecnologia de Bancos de Dados, fato que levou ao surgimento da terminologia KDD (*Knowlegde Discovery in Databases*), ou Descoberta de Conhecimento em Base de Dados.

Segundo Fayyad [27], o modelo tradicional de transformação dos dados em informação (conhecimento) consiste em um processamento manual de todas as informações obtidas, por especialistas que, então, produzem relatórios que deverão ser analisados. Devido ao grande volume de dados existentes na maioria das situações, o processo manual supracitado torna-se impraticável. Ainda segundo Fayyad, o KDD é uma tentativa de solucionar o problema advindo da chamada "era da informação", a sobrecarga de dados.

Para Wives [113], os métodos e ferramentas utilizados para realizar o KDD foram desenvolvidos com base em três áreas: Estatística, Inteligência Artificial e Recuperação de Informações (*Information Retrieval*).

Fayyad [27] define o KDD como: "um processo não trivial de identificação de novos padrões válidos, úteis e compreensíveis". O principal objetivo desse processo está ligado à descoberta de relacionamentos e dados implícitos em registros de bancos de dados, através do estudo e desenvolvimento de processos de extração de conhecimento. A Figura 2.2 apresenta as principais fases do KDD, descritas a seguir.

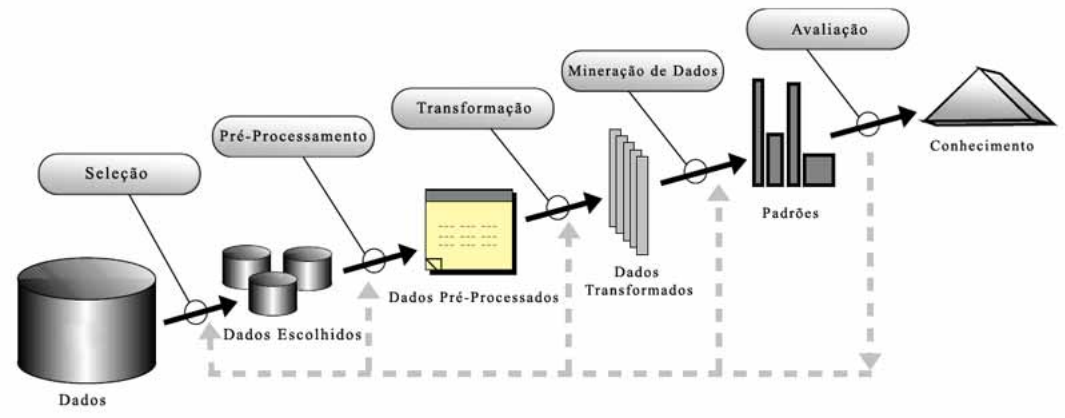


Figura 2.2: *Visão geral do processo de KDD*

- **Seleção:** Nesta etapa deverão ser definidos os objetivos pretendidos com o processo de KDD. Com base nestes objetivos realiza-se a seleção dos dados disponíveis identificando aqueles que possam, de fato, contribuir com o processo. Esta etapa é fundamental para o andamento do mesmo, pois é a partir dela que as demais decisões serão tomadas.
- **Pré-Processamento e Transformação:** São realizadas, nestas etapas, operações que visam padronizar e uniformizar os dados. Compreendem operações de: extração e integração, transformação, limpeza, seleção e redução. O principal objetivo destas etapas é preparar os dados para a etapa de mineração.
- **Mineração de Dados:** É nesta etapa que serão aplicadas diversas técnicas (inteligência artificial, aprendizado de máquina, redes neurais, estatística, reconhecimento de padrões, entre outras) sobre os dados, para que sejam extraídas deles informações. Este assunto será tratado com maior detalhamento na Seção 2.2.
- **Avaliação:** Nesta etapa, os resultados obtidos são analisados, validados e distribuídos. Os ajustes necessários são feitos e o processo recomeça.

2.1.2 Descoberta do Conhecimento em Dados Não-Estruturados

Entende-se pelo termo "dado não-estruturado" aquele dado que não se encontra padronizado, nem seguindo uma organização prévia de distribuição e armazenamento. Em geral, quando se fala em dados não-estruturados, refere-se aos dados que estão armazenados no formato textual, em sua forma livre.

Com o advento da Internet e a popularização da mesma e de seus serviços (*e-mails*, *intranets*, *blogs*, *sites*) teve início a geração de um grande contingente de informações não estruturadas e semiestruturadas. Isto possibilitou o surgimento de uma nova área de descoberta de conhecimento, intitulada KDT (*Knowledge Discovery from Texts* ou Descoberta de Conhecimento em Textos) [114].

Estima-se que, de todas as informações contidas nas organizações, 80% delas estejam contidas em um formato textual, o que traz um grande desafio para esta área [102].

As etapas do processo de KDT são similares às do KDD. A Figura 2.3 mostra o processo do KDT. Inicialmente, os documentos relevantes a uma consulta do usuário são coletados. Realiza-se o pré-processamento destes documentos, visando, dentre outras coisas: eliminar termos desnecessários, corrigir ortografia e padronizar os textos. Em seguida, são aplicadas técnicas de mineração, visando extrair as informações contidas nos dados. Por fim, são interpretados os resultados e produz-se conhecimento.

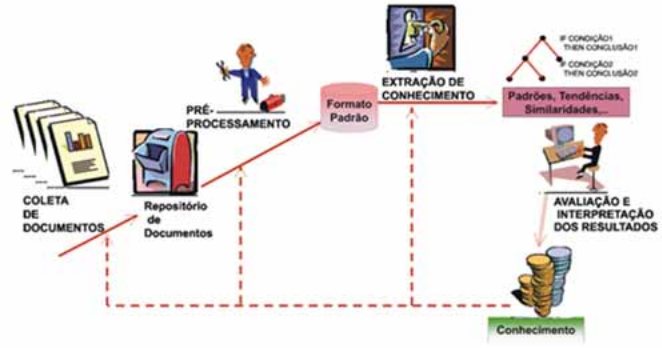


Figura 2.3: *Visão geral do processo de KDT*

2.2 Mineração de Dados

A Mineração de Dados, do inglês *Data Mining*, considerada uma área interdisciplinar inicialmente teve grande influência de três outras áreas: Estatística, Banco de Dados e Aprendizado de Máquina, sendo que destas áreas vieram suas principais definições:

- Em Hand et al. [44], a definição é dada sob uma perspectiva estatística: “Mineração de Dados é a análise de grandes conjuntos de dados a fim de encontrar relacionamentos inesperados e de resumir os dados de uma forma que eles sejam tanto úteis quanto compreensíveis ao dono dos dados”.
- Em Cabena et al. [13], a definição é dada a partir de uma perspectiva de banco de dados: “Mineração de Dados é um campo interdisciplinar que junta técnicas de máquinas de conhecimentos, reconhecimento de padrões, estatísticas, banco de dados e visualização, para conseguir extrair informações de grandes bases de dados”.
- Em Fayyad et al. [27], a definição é dada da perspectiva do aprendizado de máquina: “Mineração de Dados é um passo no processo de Descoberta de Conhecimento que consiste na realização da análise dos dados e na aplicação de algoritmos de

descoberta que, sob certas limitações computacionais, produzem um conjunto de padrões de certos dados".

Analisa-se, a seguir, a forma de classificação da Mineração de Dados e é feito um estudo sobre a unidade básica da mineração: os dados.

2.2.1 Tarefas

Larose [53] classifica a Mineração de Dados de acordo com as tarefas que podem ser realizadas. As tarefas mais comuns são:

Descrição (*Description*) Utilizada para descrever os padrões e tendências reveladas pelos dados. A descrição geralmente oferece uma possível interpretação para os resultados obtidos. É muito utilizada em conjunto com as técnicas de análise exploratória de dados, para comprovar a influência de certas variáveis no resultado obtido.

Classificação (*Classification*) Uma das tarefas mais comuns, a Classificação visa identificar a qual classe um determinado registro pertence. Nesta tarefa, o modelo analisa o conjunto de registros fornecidos, e cada registro indica a qual classe ele pertence, a fim de "aprender" como classificar um novo registro (aprendizado supervisionado). Por exemplo, é categorizado cada registro de um conjunto de dados contendo as informações sobre os colaboradores de uma empresa: Perfil Técnico, Perfil Negocial e Perfil Gerencial. O modelo analisa os registros e então é capaz de dizer em qual categoria um novo colaborador se encaixa. A tarefa de classificação pode ser usada, por exemplo, para:

- Determinar quando uma transação de cartão de crédito pode ser uma fraude;
- Identificar, em uma escola, qual a turma mais indicada para um determinado aluno;
- Diagnosticar onde uma determinada doença pode estar presente;
- Identificar quando uma pessoa pode ser uma ameaça em termos de segurança.

Estimação (*Estimation*) ou Regressão (*Regression*) A estimação é similar à classificação, porém é usada quando o registro é identificado por um valor numérico e não um categórico. Assim, pode-se estimar o valor de uma determinada variável analisando os valores das demais. Por exemplo, um conjunto de registros contendo os valores mensais gastos por diversos tipos de consumidores e de acordo com os hábitos de cada um, em que, após ter analisado os dados, o modelo é capaz de dizer, por analogia, qual será o valor gasto por um novo consumidor. A tarefa de estimação pode ser usada, por exemplo, para:

- Estimar a quantia a ser gasta por uma família de quatro pessoas durante a volta às aulas;

- Estimar a pressão ideal de um paciente tendo como base a idade, o sexo e a massa corporal.

Predição (*Prediction*) A tarefa de predição é similar às tarefas de classificação e estimação, porém ela visa descobrir o valor futuro de um determinado atributo. Alguns exemplos estão enumerados abaixo:

- Predizer o valor de uma ação com três meses de adiantamento;
- Predizer o percentual de tráfego que será aumentado na rede se a velocidade aumentar;
- Predizer o vencedor do campeonato baseando-se na comparação das estatísticas dos times.

Agrupamento (*Clustering*) A tarefa de agrupamento visa identificar e aproximar os registros similares. Um agrupamento (ou *cluster*) é uma coleção de registros similares entre si, porém, diferentes dos outros registros nos demais agrupamentos. Esta tarefa difere da classificação, pois não necessita que os registros sejam previamente categorizados (aprendizado não-supervisionado). Além disso, ela não tem a pretensão de classificar, estimar ou predizer o valor de uma variável. Ela apenas identifica os grupos de dados similares. O agrupamento pode ser utilizado, por exemplo, para:

- Segmentação de mercado para um nicho de produtos;
- Auditoria, separando comportamentos suspeitos;
- Reduzir para apenas um conjunto de atributos registros similares com centenas de atributos.

Associação (*Association*) A tarefa de associação consiste em identificar o relacionamento entre atributos. Eles se apresentam na forma: *SE* atributo X *ENTÃO* atributo Y ($X \Rightarrow Y$). É uma das tarefas mais conhecidas devido aos bons resultados obtidos, principalmente nas análises da "Cestas de Compras" (*Market Basket*), em que identificamos quais produtos são levados junto pelos consumidores. Alguns exemplos da sua aplicação são:

- Determinar os casos em que um novo medicamento pode apresentar efeitos colaterais;
- Identificar os usuários de planos que respondem bem à oferta de novos serviços.

2.2.2 Dados

Conhecer o tipo dos dados com os quais se trabalhará também é fundamental para a escolha da técnica de mineração mais adequada. Os dados podem ser categorizados em dois tipos: quantitativos e qualitativos. Os dados quantitativos são representados por

valores numéricos. Eles ainda podem ser discretos e contínuos. Já os dados qualitativos contêm os valores nominais e ordinais (categóricos). Em geral, antes de se aplicar os algoritmos de mineração é necessário explorar, conhecer e preparar os dados.

Nesse sentido, uma das primeiras atividades é obter uma visualização dos dados, de forma que se possa ter uma visão geral, para depois decidir qual linha seguir. Diversas são as técnicas utilizadas para a visualização dos dados. Simoff [98], Rezende [88], Myatt [66], Myatt et al. [67], NIST [69] e Canada [14] apresentam diversas abordagens para as visualizações. Keim [50] apresenta um estudo sobre as diversas técnicas de visualização. A Figura 2.4 mostra a evolução destas técnicas.

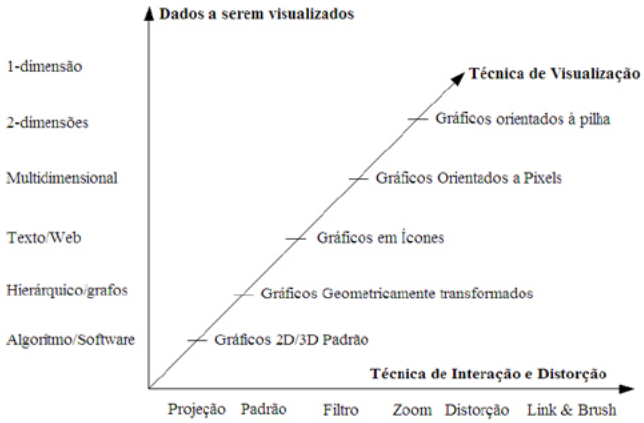


Figura 2.4: *Evolução das técnicas de visualização*

Com uma visão inicial dos dados definida, é necessário explorá-los, buscando, além de mais conhecimento sobre os mesmos, encontrar valores que possam comprometer sua qualidade, tais como: valores em branco ou nulos, valores viciados, variáveis duplicadas, dentre outros. À medida em que problemas vão sendo encontrados e o entendimento vai sendo obtido, ocorre a preparação dos dados para que os algoritmos de mineração possam ser aplicados. Segundo Olson et al. [75], o processo de preparação dos dados, na maioria dos projetos de mineração, compreende até 50% de todo o processo. Para McCue [62], esta etapa pode compreender até 80%.

Han e Kamber [43] descrevem várias técnicas estatísticas de análise de dispersão (*Quartiles*, Variância) e de medida central (média, mediana, moda e faixa de valores) combinadas com gráficos (Histogramas, Frequência, Barra, *BoxPlot*, Dispersão) que podem ser utilizados na exploração dos dados. Myatt [66] utiliza a técnica de Análise Exploratória dos Dados (*EDA - Exploratory Data Analisis*) para auxiliar nessa atividade.

O processo de preparação dos dados para a mineração, também chamado de pré-processamento (Figura 2.5), segundo Han et al. [43], consiste principalmente em:

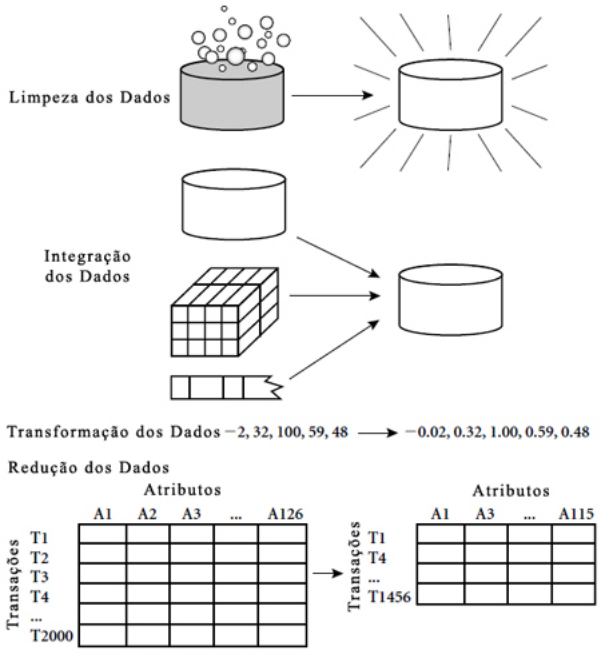


Figura 2.5: *Atividades do pré-processamento*

Limpeza dos dados: Frequentemente, os dados são encontrados com diversas inconsistências, tais como registros incompletos, valores errados e dados inconsistentes. A etapa de limpeza dos dados visa eliminar estes problemas de modo que eles não influenciem no resultado dos algoritmos usados. As técnicas usadas nesta etapa vão desde a remoção do registro com problemas, passando pela atribuição de valores padrões, até a aplicação de técnicas de agrupamento para auxiliar na descoberta dos melhores valores. Devido ao grande esforço exigido nesta etapa, Han et al. [43] propõem o uso de um processo específico para a limpeza dos dados.

Integração dos dados: É comum obter os dados a serem minerados de diversas fontes: banco de dados, arquivos de textos, planilhas, *data warehouses*, vídeos, imagens, entre outras. Surge, então, a necessidade da integração destes dados de forma a obter um repositório único e consistente. Para isto, é necessária uma análise aprofundada dos dados observando redundâncias e dependências entre as variáveis e valores conflitantes (categorias diferentes para os mesmos valores, chaves divergentes, regras diferentes para os mesmos dados etc.).

Transformação dos dados: A etapa de transformação dos dados merece destaque. Alguns algoritmos trabalham apenas com valores numéricos e outros apenas com valores categóricos. Nestes casos, é necessário transformar os valores numéricos em categóricos ou os categóricos em valores numéricos. Não existe um critério único para transformação dos dados e diversas técnicas podem ser usadas de acordo com os objetivos pretendidos. Algumas das técnicas empregadas nesta etapa são: suavização (remove valores errados dos dados), agrupamento (agrupa valores em

faixas sumarizadas), generalização (converte valores muito específicos para valores mais genéricos), normalização (coloca as variáveis em uma mesma escala) e criação de novos atributos (gerados a partir de outros já existentes).

Redução dos dados: O volume de dados usado na mineração costuma ser alto. Em alguns casos, este volume é tão grande que torna o processo de análise dos dados e da própria mineração impraticável. Nestes casos, as técnicas de redução de dados podem ser aplicadas para que a massa de dados original seja convertida em uma massa de dados menor, sem perder, porém, a representatividade original. Isto permite que os algoritmos de mineração sejam executados com mais eficiência, mantendo a qualidade do resultado. As estratégias adotadas nesta etapa são: criação de estruturas otimizadas para os dados (cubos de dados), seleção de um subconjunto dos atributos, redução da dimensionalidade e discretização. Dentre as diversas técnicas, a PCA - *Principal Components Analysis*, desempenha um papel muito importante na redução da dimensionalidade [94]. Outra técnica muito utilizada é a Discretização Baseada na Entropia [43].

Geralmente, os repositórios usados possuem milhares de registros. Neste contexto, o uso de todos os registros do repositório, para a construção do modelo de Mineração de Dados, é inviável. Assim, utiliza-se uma amostra (mais representativa possível), a qual é dividida em três conjuntos:

1. Conjunto de Treinamento (*Training Set*): conjunto de registros usados no qual o modelo é desenvolvido;
2. Conjunto de Testes (*Test Set*): conjunto de registros usados para testar o modelo construído;
3. Conjunto de Validação (*Validation Set*): conjunto de registros usados para validar o modelo construído;

Essa divisão em grupos é necessária para que o modelo não fique dependente de um conjunto de dados específico e, ao ser submetido a outros conjuntos (com valores diferentes dos usados na construção e validação do modelo), apresente resultados insatisfatórios. Este efeito é chamado de efeito *Bias*. À medida que a precisão do modelo para um conjunto de dados específico é aumentada, perde-se a precisão para outros conjuntos.

Apesar da grande maioria dos repositórios conter um volume alto de registros, em alguns casos o que ocorre é o inverso. Neste caso, algumas estratégias foram desenvolvidas para gerar o conjunto de dados a partir dos registros existentes [11] [110] [112].

É importante destacar que, apesar de existir um volume muito grande de dados nas empresas, eles raramente são disponibilizados para fins de pesquisa. Assim, muitas vezes, novos algoritmos são criados de forma teórica em ambientes acadêmicos e, pela

falta de dados, não se consegue uma avaliação em um ambiente mais próximo do real. Para auxiliar nas pesquisas, repositórios comuns e públicos com diversas bases de dados foram criados por diversas instituições. Um dos repositórios mais conhecidos, com bases em diferentes negócios, tamanhos e tipos, pode ser encontrado em [83].

2.3 Mineração de Textos

A Mineração de Textos surgiu da intersecção de várias áreas de pesquisa, tais como Recuperação de Informação, Processamento de Linguagem Natural, Extração de Informação e Mineração de Dados, visando auxiliar na extração de informação útil em textos [20]. A Recuperação da Informação visa trabalhar o armazenamento e localização das informações. O Processamento de Linguagem Natural preocupa-se com a análise morfossintática do texto. A Extração de Informação visa extrair conceitos pré-definidos do texto. A Mineração de Dados, por sua vez, visa descobrir as informações implícitas em bancos de dados.

Segundo Lopes [58], o termo refere-se ao processo de extração de padrões interessantes e não triviais, ou conhecimento a partir de documentos em textos não-estruturados. Moura [65] descreve a Mineração de Textos como sendo uma área de pesquisa tecnológica cujo objetivo é a busca por padrões, tendências e regularidades em documentos escritos em linguagem natural. Para Wives [114], a Mineração de Textos pode ser entendida como a aplicação de técnicas de KDD sobre dados extraídos de textos e complementada com qualquer outra técnica que visa descobrir conhecimento em dados não-estruturados.

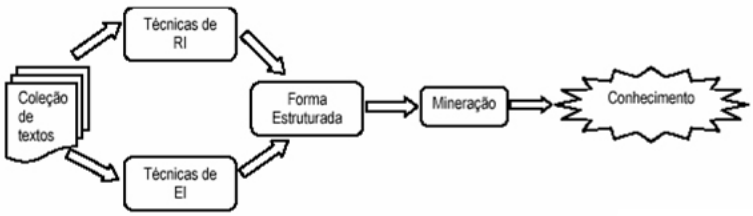


Figura 2.6: *Etapas da mineração de textos*

Observa-se na Figura 2.6 que as técnicas de Recuperação de Informação e de Extração de Informações são utilizadas sobre os documentos selecionados, a fim de que seja gerada uma forma estruturada e, então, seja aplicada a técnica de Mineração de Dados. Segundo Dixon [25], as principais etapas da Mineração de Textos são:

- **Recuperação de Informação:** momento de localização e recuperação dos documentos com base na solicitação do usuário. Nesta etapa é possível a extração de termos que representam os documentos, possibilitando a comparação direta entre

os termos e a consulta do usuário. A Recuperação de Informação auxilia no processo de indexação, como veremos na Seção 2.4.

- **Extração de Informação:** os itens relevantes são extraídos dos documentos possibilitando a sua representação em formato tabular. Nesta etapa podem ser extraídas informações semânticas e lexicais dos dados textuais. Isso será visto em mais detalhes na Seção 2.5.
- **Mineração de Informação:** aplicação das técnicas de Mineração de Dados sobre os resultados tabulados. Mais informações sobre as técnicas de Mineração de Dados podem ser vistas em [19], [43], [53] e [62].
- **Interpretação:** os resultados obtidos são analisados e interpretados. Os ajustes são feitos e caso se faça necessário, o processo se repete.

2.4 Recuperação de Informação

Manning et. al. [60] definem o processo de Recuperação da Informação - RI, do inglês *Information Retrieval - IR*, como a busca por um documento de natureza não estruturada, normalmente um texto, que consiga alcançar o objetivo de obter a informação dentre um grande conjunto de documentos, normalmente armazenados em computador.

Segundo Salton et. al. [89], o processo de RI necessita de técnicas que agilizem e facilitem o acesso aos dados. Tais técnicas são realizadas de forma manual ou automática por um processo chamado indexação. Segundo Han et. al. [43], a recuperação da informação dar-se-á pela entrada do usuário através de uma consulta e a busca geralmente ocorrerá através de palavras-chaves ou pelo cálculo da similaridade entre os documentos e a consulta.

A seguir, serão abordados pilares para a Recuperação de Informação: processo de indexação, os modelos de recuperação de informação e as medidas de eficácia.

2.4.1 Processo de Indexação

Salton et. al. [89] definem a indexação como o processo através do qual é criado um índice de palavras para cada documento, permitindo a busca dos documentos pelas palavras-chave que eles contém. Para melhorar o desempenho da busca é usado um índice invertido. A Figura 2.7 mostra a estrutura de um índice invertido: uma estrutura contendo os termos aponta para uma lista que contém os documentos com as palavras-chave.

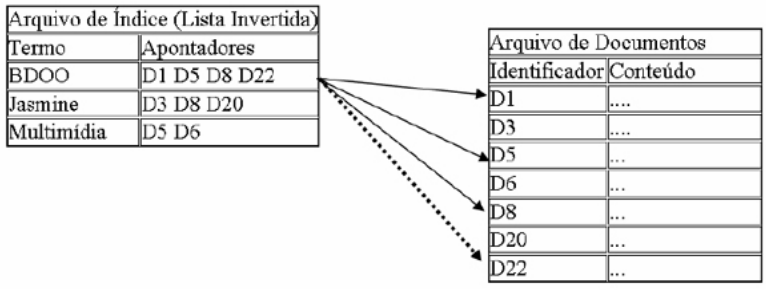


Figura 2.7: *Estrutura do Índice Invertido*

Em geral, o processo de indexação é composto pelas seguintes etapas:

Análise Léxica Consiste na subdivisão do documento textual em um conjunto de palavras (denominados *tokens*), convertidas para um mesmo tipo de letra (maiúsculo ou minúsculo), através da identificação de caracteres delimitadores. Neste processo, é comum a eliminação de dígitos numéricos, hífens, sinais de pontuação, acentos ou quaisquer outros caracteres especiais que não contribuam para a representatividade do documento.

Remoção de *Stopwords* Etapa na qual é feita a remoção de palavras consideradas irrelevantes para a representação do documento (*stopwords*), tais como: artigos, pronomes e preposições.

Stemming Consiste no processo da remoção das variações das palavras, deixando apenas os termos raízes. Por variações, entende-se: conjugações, plurais, gerúndios e sufixos.

Determinação de pesos É utilizada uma medida de frequência relativa, para atribuir um grau de relevância de um determinado termo em relação à coleção de documentos. A medida de cálculo do peso varia de acordo com o modelo de RI escolhido.

Seleção de termos-índice Define quais palavras serão utilizadas como elementos de indexação. Pode-se utilizar, por exemplo, dois ou mais termos para representar um conceito presente no documento.

Criação do Dicionário Léxico Consiste na criação de um dicionário léxico dos termos encontrados, de forma que permita, por exemplo, a expansão da consulta original.

2.4.2 Modelos de Recuperação de Informação

Os modelos de recuperação de informação funcionam como o mecanismo de busca para recuperação dos documentos, que se baseiam em uma consulta informada. Os modelos clássicos são: Booleano, Vetorial e Probabilístico. Outros modelos podem ser vistos em [60].

2.4.2.1 Modelo Booleano

O modelo Booleano é baseado na álgebra Booleana. A ideia consiste em representar a consulta como uma expressão booleana convencional formada pelos conectores lógicos “AND”, “OR” e “NOT”. A recuperação baseia-se no critério da decisão binária, assim, os documentos recuperados são aqueles que atendem a expressão.

2.4.2.2 Modelo Vetorial

No modelo Vetorial, tanto os documentos quanto as consultas são representadas como vetores de termos no espaço euclidiano $\mathbb{R}^{|T|}$, onde T é a quantidade de termos que constitui os vetores. Para cada termo, existe um valor associado que indica o peso, ou grau de relevância, do termo no vetor que representa o documento ou a consulta. O peso do termo t_i no documento d_j, denominado w_{ij}, é frequentemente calculado como se segue [90]:

$$w_{ij} = f_{ij} \times \log \frac{N}{n_i} \tag{2-1}$$

onde f_{ij} é a frequência do termo i no documento d_j, N é o número de documentos da coleção e n_i é o número de documentos em que o termo i ocorre. O valor w_{ij} também é conhecido como TF-IDF.

A parcela $\log \frac{N}{n_i}$ é denominada IDF (*Inverse Document Frequency*). O valor f_{ij}, também conhecido como TF (*Term Frequency*), é uma medida da importância do termo i no documento, ou seja, quanto mais frequente é um termo, mais importante ele é para o documento. O valor IDF é uma medida da importância do termo i na coleção, ou seja, quanto menos frequente é um termo, mais importante ele é na coleção [97].

A similaridade entre uma consulta q e um documento d pode ser calculada através do cosseno do ângulo formado pelos vetores de termos q e d, utilizando a seguinte expressão [18]:

$$sim(d,q) = \frac{\sum_{i=1}^{t} w_{id} \times w_{iq}}{\sqrt{\sum_{i=1}^{t} (w_{id})^2} \times \sqrt{\sum_{i=1}^{t} (w_{iq})^2}} \tag{2-2}$$

onde w_{id} é o peso do *i-ésimo* termo do vetor d e w_{iq} é o peso do *i-ésimo* termo do vetor q.

Na Figura 2.8, é ilustrada a medida do cosseno do ângulo θ entre o documento d_1 e a consulta q, onde:

- $\vec{v}(d_i)$ representa o documento d_i no espaço euclidiano $\mathbb{R}^2$;

- $\vec{v}(q)$ representa a consulta no espaço euclidiano $\mathbb{R}^2$.

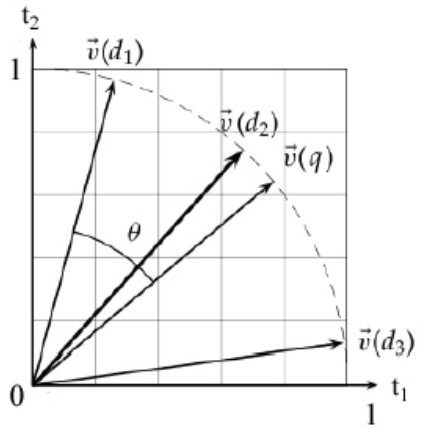


Figura 2.8: *Cosseno* θ *entre o documento* d_1 *e a consulta* q

Ao calcular o cosseno do ângulo θ entre um documento e a consulta, quanto mais próximo de 1 for o resultado, maior é a similaridade entre este documento com relação à consulta.

2.4.2.3 Modelo Probabilístico

O modelo probabilístico trabalha com conceitos provenientes da área de probabilidade e estatística. A base deste modelo está no princípio da ordenação probabilística (*Probability Ranking Principle*): dada uma consulta q e um documento d_j de uma coleção, tenta-se estimar a probabilidade do usuário considerar o documento d_j relevante à consulta q.

Este modelo assume que a probabilidade de relevância depende somente das representações da consulta e do documento. Este modelo também supõe a existência de um conjunto ótimo de documentos, que maximiza toda a probabilidade de relevância para o usuário. Os documentos deste conjunto são considerados relevantes para uma consulta q e os documentos que não estão neste conjunto são considerados não relevantes [73].

No modelo probabilístico, a similaridade entre um documento d_j e uma consulta q é definida por:

$$sim(d_j, q) = \frac{\mathrm{p}(Rel|d_j)}{\mathrm{p}(\overline{Rel}|d_j)} \tag{2-3}$$

onde:

- Rel: é o conjunto de documentos que foram estimados como relevantes para a consulta q, isto é, uma estimativa para conjunto ótimo de documentos;
- $\overline{Rel}$: é o conjunto complementar de Rel;

- P(Rel, d_j): é a probabilidade do documento d_j ser relevante para a consulta q;
- P($\overline{Rel}$, d_j): é a probabilidade do documento d_j não ser relevante para a consulta q.

Em uma situação prática, como não se sabe previamente quais documentos são relevantes, o conjunto ótimo de documentos relevantes deve ser inicialmente estimado e melhorado através de interações com o usuário.

2.4.3 Medidas de Eficácia

Através das medidas de eficácia, é possível verificar o grau de conformidade e satisfação com relação à resposta de uma determinada consulta. Os critérios comumente usados são as medidas de precisão (*precison*) e abrangência (*recall*) [20].

A precisão representa o percentual de documentos efetivamente recuperados numa consulta em relação ao total de respostas obtidas. A abrangência representa o percentual de documentos efetivamente recuperados por uma consulta em relação ao total de respostas previstas.

2.5 Extração de Informação - EI

O processo de Extração de Informação visa extrair informações relevantes dos textos através do isolamento de fragmentos signficativos. Assim, pode-se transformar a informação não estruturada em dados estruturados.

Figura 2.9: *Processo de Extração de Informação*

Conforme mostra a Figura 2.9, o processo de EI contempla três etapas:

Criação dos *Slots* A EI deve ser feita em um domínio específico onde se conheça as informações que podem estar contidas nos dados. Este conhecimento sobre o domínio deve ser expresso em forma de estruturas atributo-valor, que indicam as informações que devem ser extraídas dos textos. A essas estruturas dá-se o nome de *slots*.

Documento com *Tags* Após os *slots* serem criados, um analisador lexical verifica o texto a fim de preencher estes *slots*. Ao final, em cada posição do texto em que foi encontrada uma informação a ser extraída é inserida uma *tag*, ou marcação, do tipo SGML (*Standard Generation Markup Language*) [93].

Tabelas com campos pré-definidos, ou *templantes* Depois de serem marcados no texto os pontos que contém informações que podem ser extraídas, é utilizado um *template* para definir quais *slots* serão extraídos e essas informações são armazenadas em uma estrutura tabular, o que permitirá a aplicação de técnicas de mineração de dados.

O processo de Extração de Informação se utiliza das técnicas de Processamento de Linguagem Natural (PLN) [81] para análise do texto. Uma das grandes dificuldades do PLN reside na questão da ambiguidade léxica causada pela linguagem natural. Neste sentido, é utilizado o conceito de marcação POS (*Part-of-Speech*) para automaticamente atribuir tags, indicando a categoria morfossintática (verbo, sustântivo, adjetivo) das palavras dos textos. Com isto, é possível filtrar as palavras menos significativas por sua categoria morfossintática. Aos programas utilizados para realizar essa marcação POS dá-se o nome de *taggers*.

2.6 Ontologia

O termo ontologia tem origem na Filosofia e é relativo à existência do ser [34]. Entretanto, no campo da computação o termo possui outro significado. Ainda não há consenso na definição formal do termo ontologia [39]. Será adotada aqui uma das definições mais referenciadas definida por Gruber[34] [35], inicialmente no campo da Inteligência Artificial, como "a especificação explícita de uma conceitualização" e, mais recentemente, no contexto da Ciência da Computação e da Informação conforme o seguinte trecho [36]:

> "... uma Ontologia define um conjunto de representações primitivas com o qual se modela um domínio de conhecimento ou discurso..."

Apesar de não haver um consenso sobre o termo, diversos autores concordam que as ontologias são um meio de permitir o compartilhamento e a reutilização do conhecimento. Guarino apresenta uma definição formal para o conceito de Ontologia em [40].

Diversas são as áreas de aplicação das ontologias: Recuperação de Informações, Gestão do Conhecimento, Educação, Processamento de Linguagem Natural, Mineração de Dados, entre outras. Dentre elas, a Web Semântica tem obtido grandes avanços. Atualmente, as ontologias são uma recomendação do W3C [111] para a Web Semântica [9] como padrão de vocabulário comum para trocar dados, tornar o conhecimento reutilizável e facilitar a comunicação de sistemas heterogêneos [36].

A seguir serão abordados os tipos de ontologias, as linguagens utilizadas para representá-las e as metodologias existentes para construí-las.

2.6.1 Tipos de Ontologias

Segundo Heijs et. al. [46] e Guarino [39] podemos classificar as ontologias quanto a sua função em:

Ontologias Genéricas de Topo (ou Fundamentação) São ontologias que descrevem conceitos mais amplos. Não dependem de um problema específico (domínio). Geralmente representam conceitos da natureza, relativos ao espaço e ao tempo.

Ontologias de Domínio Representam conceitos de um domínio (área) específico, tais como: medicina, genética, computação. São os tipos de ontologias mais comuns.

Ontologias de Aplicação Descrevem conceitos que estão relacionados com um domínio específico (área) e com uma tarefa específica.

Ontologias de Representação Descrevem os conceitos que são usados para a representação do conhecimento.

Ontologias de Tarefa Descrevem conceitos que são usados por processos (tarefas e atividades) de uma maneira geral, sem a dependência de um domínio específico. Como exemplos, podem ser citados processos de compra e venda.

Em [5], é feito um resumo sobre as diversas classificações encontradas na literatura.

2.6.2 Linguagem

As Ontologias precisam ser descritas em alguma linguagem, para que possam então ser processadas pelas máquinas. Existem diversas linguagens para a representação das Ontologias. Elas variam de acordo com o seu poder de formalismo e expressividade. Dentre elas, podem ser citadas: SHOE (*Simple HTML Ontology Extensions*) [95], XOL (*Ontology Exchange Language*) [116], DAML (*DARPA Agent Markup Language*) [21] e OIL (*Ontology Inference Layer*) [71]. Estas duas últimas foram combinadas e formaram a DAML+OIL [22].

A OWL (*Web Ontology Language*) [4] é uma revisão da linguagem DAML+OIL [76]. Desde 10 de fevereiro de 2004, esta é a linguagem recomendada pelo W3C (*World Wide Web Consortium*) [107] para a representação de ontologias.

Em termos de sua expressividade para a representação de conteúdo semântico interpretável por máquinas, a OWL pode ser considerada como uma evolução das demais linguagens para representação de ontologias.

De acordo com o W3C, a linguagem OWL é projetada para ser utilizada pelas aplicações que precisam processar o conteúdo das informações, ao invés de apenas apresentar estas informações aos seres humanos.

As ontologias OWL podem ser classificadas em três espécies, de acordo com a sub-linguagem utilizada: OWL-Lite, OWL-DL e OWL-Full. A característica principal de cada sub-linguagem é a sua expressividade: a OWL-Lite é a menos expressiva, a OWL-Full é a mais expressiva e a expressividade da OWL-DL está entre a OWL-Lite e a OWL-Full.

OWL-Lite A OWL-Lite é a sub-linguagem sintaticamente mais simples. Destina-se a situações em que apenas são necessárias restrições e uma hierarquia de classe simples. Por exemplo, a OWL-Lite pode fornecer uma forma de migração para *tesauros*[1] existentes, bem como de outras hierarquias simples.

OWL-DL A OWL-DL é mais expressiva que a OWL-Lite e baseia-se em lógica descritiva, um fragmento de Lógica de Primeira Ordem, passível, portanto, de raciocínio automático. É possível assim computar automaticamente a hierarquia de classes e verificar inconsistências na ontologia.

OWL-Full A OWL-Full é a sub-linguagem mais expressiva. Destina-se a situações em que a alta expressividade é mais importante do que garantir a decidibilidade ou completeza da linguagem. Não é possível efetuar inferências em ontologias OWL-Full.

2.6.3 Metodologias

O processo de desenvolvimento de uma Ontologia varia de metodologia para metodologia, mas em geral consiste em: determinar o domínio e o escopo, definir e organizar as classes na ontologia em uma taxonomia (subclasse/superclasse), definir atributos, definir relações, definir instâncias (elementos) e definir axiomas (sentenças que são sempre verdadeiras).

Segundo Gruber [35], a Ontologia deve seguir os seguintes princípios:

- *Clareza:* uma ontologia deve, de forma efetiva, comunicar o significado pretendido dos termos definidos. Na definição da ontologia, deve-se ter a objetividade de definir apenas o que se presume ser útil na resolução da classe de problemas a ser atingida. Definições completas, com condições necessárias e suficientes, devem preceder definições parciais. Todas as definições devem ser documentadas em uma linguagem natural;
- *Coerência:* as inferências derivadas da ontologia devem ser corretas e consistentes com as definições formais e informais;

[1] Tesauro, também conhecido como dicionário de idéias afins, é uma lista de palavras com significados semelhantes, dentro de um domínio específico de conhecimento. [5]

- *Extensibilidade:* uma ontologia deve permitir a definição de novos termos, através de extensões e especializações, sem a necessidade de revisão da teoria existente, que consiste na revisão lógica automática de uma base de conhecimento em busca de contradições;
- *Codificação mínima:* devem ser especificados conceitos genéricos independentes de padrões estabelecidos para mensuração, notação e codificação, garantindo a extensibilidade. A codificação deve ser minimizada, pois os agentes que compartilham o conhecimento podem ser implementados em diferentes sistemas de representação e estilos de representação;
- *Compromisso ontológico mínimo:* apenas o conhecimento essencial deve ser incluído, gerando a menor teoria possível acerca de cada conceito, e permitindo a criação de conceitos novos, mais especializados ou estendidos, maximizando o reuso.

Existem diversas metodologias que podem ser usadas para a construção de uma Ontologia, destacando-se: *Cyc* [87], *Grüninger e Fox* [37], *Uschold e King* [104], *Kactus* [8], *Methontology* [28], *Sensus* [101], *Ontology Development* 101 [70] e *On-To-Knowledge* [100].

A escolha de qual metodologia usar dependerá de diversas questões, tais como: o grau de formalidade exigido, o domínio do assunto e o tipo da ontologia que se deseja modelar.

2.7 Análise Criminal

Gottlieb et. al. [33] definem a Análise Criminal como a atividade sistêmica que visa identificar padrões de crimes, bem como suas relações e tendências, servindo como base de apoio para as decisões operacionais e administrativas do alto comando policial. Neste contexto, segundo Dantas et. al. [24], o termo padrão refere-se ao grau de ocorrência de uma determinada característica em diversos crimes. Já a tendência se refere a uma medida quantitativa geral indicando o possível estado de um determinado evento, como a diminuição da criminalidade.

Percebe-se que os eventos criminais, em sua grande maioria, ao contrário do que se imagina, não são aleatórios. Assim, existe uma lógica entre a ocorrência dos crimes e o meio no qual estão inseridos [23].

A Análise Criminal apoia-se fundamentalmente nas informações disponíveis, quer sejam os eventos de crime ocorridos ou os registros históricos das soluções. Devido ao grande volume de informações que necessitam ser trabalhadas, torna-se praticamente impossível fazê-lo de forma manual. Neste sentido, com o advento dos computadores, foi a partir da década de 80 que intensificou-se o processo de Análise Criminal.

Magalhães [59] ressalta a importância da distinção entre o raciocínio puramente jurídico e o trabalho de Análise Criminal. Observa-se que é necessário mais do que o conhecimento legal sobre o crime, é necessário o conhecimento do comportamento humano dos fatos, característica que exige do analista um conhecimento multidisciplinar.

Ainda segundo Magalhães [59], é importante observar que as ações resultantes da Análise Criminal não visam eliminar o problema da criminalidade, uma vez que a dinâmica do crime está em constante movimentação e evolução tornando a atividade de Análise Criminal perene, mas possibilita a criação de políticas públicas que responderão de maneira mais eficaz aos novos eventos criminais.

Pode-se resumir o trabalho da Análise Criminal nas seguintes etapas:

1. Sistematizar e analisar os dados buscando identificar padrões;
2. Analisar os padrões visando identificar suas causas;
3. Identificar formas de intervir nas causas, visando mitigar os incidentes;
4. Avaliar o impacto da intervenção e, caso necessário, iniciar o processo novamente.

Comumente, divide-se a Análise Criminal em três segmentos, de acordo com a finalidade [24]:

Análise Criminal Estratégica - ACE Voltada para a determinação de padrões de delinquência [2]. Sua utilidade principal é na área de prevenção policial. Visa analisar a criminalidade a longo prazo. Dentre seus focos estão [59]:

- Criação de políticas públicas;
- Produção de conhecimento;
- Integração com outras Secretarias na construção de ações de Segurança Pública;
- Direcionamento de investimentos;
- Formulação do plano orçamentário;
- Controle e acompanhamento de ações e projetos;
- Formulação de indicadores de desempenho.

Análise Criminal Tática - ACT É geralmente utilizada no apoio à investigação criminal, principalmente quando existe um grande número de informações que devem ser trabalhadas metodicamente. Outra utilidade da Análise Tática é a descoberta de padrões em crimes repetidos. É voltada para uma análise a médio prazo, e dentre seus focos estão [59]:

[2] A delinquência é um problema comportamental caracterizado por realizações de atos criminosos em indivíduos de menoridade penal.

- Gerar informações para orientar as atividades do policiamento ostensivo nas atividades preventivas e repressivas;
- Gerar informações para subsidiar a polícia investigativa nas soluções das ocorrências criminais, visando identificar a autoria e materialidade dos delitos.

Análise Criminal Administrativa - ACA É um meio de prover os Gestores da área de Segurança Pública informações sobre questões sociais, geográficas, econômicas, dentre outras, que tenham relevância para o enfrentamento da criminalidade. Alguns de seus focos são [59]:

- O fornecimento de informações sumarizadas para os diversos públicos (gerentes, policiais, diretores, etc.);
- A elaboração de estatísticas descritivas;
- A elaboração sobre tendências criminais;
- As comparações entre períodos e locais.

Diversas são as ferramentas que auxiliam os analistas criminais na tarefa de análise, dentre elas: Regressão e Relação entre Variáveis, Análise de Movimentos, Sistemas de Informações Geográficas e Mineração de Dados [59], sendo a última o foco deste trabalho.

CAPÍTULO 3

Ontologias, Mineração de Dados e Textos

Este capítulo aborda os conceitos relativos às Ontologias e como elas são utilizadas no processo de mineração de textos. A Seção 3.1 apresenta como as Ontologias podem ser utilizadas no processo de mineração. A Seção 3.2 mostra os principais conceitos da linguagem OWL, utilizada para representação das Ontologias. A Seção 3.3 aborda a definição de "Conceitos", utilizados neste trabalho para mapear o conhecimento dos especialistas. Por fim, na Seção 3.4 serão abordadas as formas utilizadas para o cálculo de similaridade entre Ontologias e textos.

3.1 Mineração de Dados apoiada por Ontologia

As Ontologias, introduzidas na mineração pela primeira vez no começo de 2000, podem ser usadas de diferentes formas [68]: Ontologias de Domínio e de Conhecimento, Ontologias para o processo de Mineração ou Ontologias de Metadados.

A primeira organiza os conhecimentos sobre um domínio e desempenha um papel importante no processo de mineração. A segunda representa a descrição do processo de mineração e auxilia na escolha da melhor tarefa para o problema. A terceira armazena o conhecimento sobre os itens e os relacionamentos que compõem um repositório de dados.

A Figura 3.1 apresenta um *framework* geral para esse processo. O processo começa com a extração dos dados a serem minerados, podendo, para isso, usar um *data warehouse* ou outras fontes de dados. Estas fontes de dados servem de subsídio para a ontologia de metadados. Com os dados selecionados, uma ontologia de domínio pode ser usada para preparar os dados. Em seguida, os algoritmos de mineração são aplicados. Uma ontologia para a mineração de dados pode ser usada. O resultado do algoritmo pode ser utilizado para a visualização ou para a tomada de decisão.

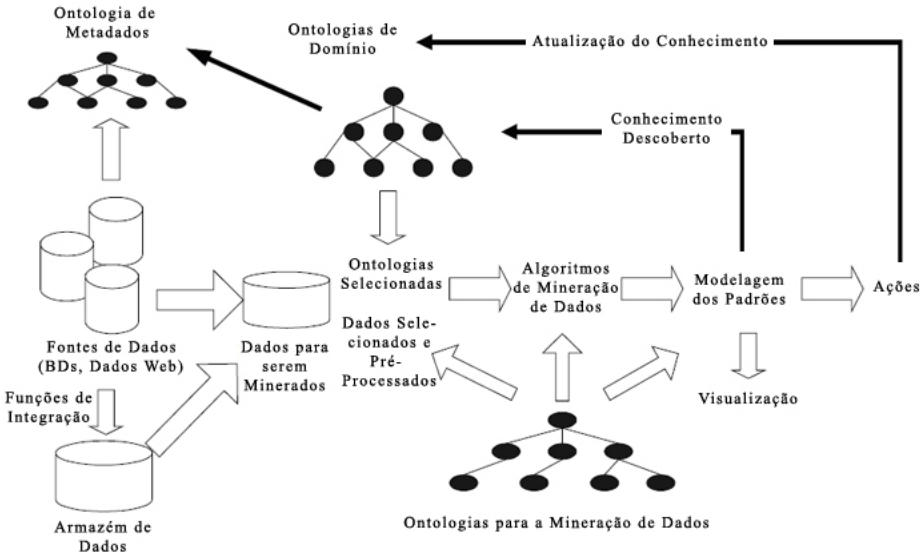


Figura 3.1: *Framework geral para a integração entre Mineração e Ontologia*

No contexto deste trabalho, a Ontologia dará suporte à extração de informações durante o processo de mineração de textos. Mais informações sobre a aplicação de Ontologias no processo de mineração podem ser vistas em [6], [7], [16], [26], [30], [55], [61], [72], [77], [106] e [115].

3.2 Padrão OWL

A OWL (*Web Ontology Language*) foi projetada para prover uma linguagem de ontologia que pudesse ser usada para descrever, de modo natural, classes e relacionamentos em documentos e aplicações Web. Os termos usados em uma ontologia devem ser escritos de maneira que possam ser interpretados sem ambiguidade e usados por agentes de software.

O documento que descreve uma ontologia possui uma sintaxe XML [45] e segue as definições propostas pelo *Web Ontology Language Guide* [76]. A seguir, serão apresentados os principais conceitos utilizados na criação de uma Ontologia.

3.2.1 Elementos Básicos

Os elementos básicos para construção de uma ontologia OWL são as classes, as instâncias das classes (também chamadas de indivíduos) e os relacionamentos entre estas instâncias (as propriedades).

3.2.1.1 Classes

As classes fornecem um mecanismo de abstração para agrupar recursos com características similares, ou seja, uma classe define um grupo de indivíduos que compartilham propriedades em comum. Os indivíduos de uma classe são chamados instâncias da classe.

Cada indivíduo na OWL é membro da classe “owl:Thing”. Deste modo, ela é superclasse de todas as classes OWL definidas pelos usuários. Além disso, existe a classe “owl:Nothing” (não possui instâncias), que é uma subclasse de todas as classes OWL. Uma classe é sintaticamente representada como uma instância nomeada da “owl:Class”, que é uma subclasse da “rdfs:Class”. A sintaxe utilizada para declaração de uma classe é apresentada abaixo:

```
<owl:Class rdf:about="#Alcool"/>
```

A linguagem OWL permite a criação de hierarquia de classes, em que as classes inferiores herdam as características das classes superiores. A seguir, é mostrada a sintaxe utilizada:

```
<owl:Class rdf:about="#Alcool">
    <rdfs:subClassOf rdf:resource="#Motivação"/>
</owl:Class>
```

3.2.1.2 Instâncias

As instâncias representam a materialização de uma classe, porém, com os atributos definidos. Os indivíduos são descritos conforme abaixo:

```
<Alcool rdf:about="Etanol"/>
```

3.2.1.3 Propriedades

As propriedades, que são relações binárias, podem ser usadas para estabelecer relacionamentos entre indivíduos ou entre indivíduos e valores de dados. Estes relacionamentos permitem afirmar fatos gerais sobre os membros das classes e podem também especificar fatos sobre indivíduos. Temos duas propriedades:

Datatype Properties Representam a relação entre indivíduos e valores de dados.
Object Properties Representa a relação entre indivíduos.

Ainda é possível especificar características das propriedades. A OWL implementa:

TransitiveProperty As propriedades transitivas indicam que: se A é subordinado a B e B é subordinado a C, então A é subordinado a C.

SymmetricProperty As propriedades simétricas indicam que: se A tem ligação com B, então B tem ligação com A.

FunctionalProperty As propriedades funcionais indicam que o indivíduo só pode assumir um único valor.

InverseOf As propriedades "Inversas De" indicam aquelas que são inversas de outras, por exemplo: a propriedade inversa de pai é filho e a de filho é pai.

Além das propriedades apresentadas, a OWL permite a definição de restrições que podem ser aplicadas a estas propriedades:

allValuesFrom Indica que todas as instâncias de uma classe que possuem determinada propriedade sejam da classe definida.

someValuesFrom Indica que pelo menos uma das classes ligadas a uma propriedade sejam da classe definida.

hasValue Indica que uma classe será membro de outra sempre que tiver a propriedade indicada.

maxCardinality Indica a quantidade máxima de indivíduos que uma mesma classe pode ter.

minCardinality Indica a quantidade mínima de indivíduos que uma mesma classe pode ter.

cardinality Indica a quantidade exata de indivíduos que uma mesma classe pode ter.

3.3 Definição de Conceitos

A proposta deste trabalho utiliza-se dos "Conceitos", definidos em uma Ontologia, para representar o conhecimento e permitir a extração automática de informações dos dados textuais. O Dicionário Aurélio [29] define conceito como: "É a representação de um objeto pelo pensamento, por meio de suas características gerais". Segundo Loh [57], a abordagem por conceito é aplicada com sucesso em diversas áreas.

Em geral, em um processo de mineração de textos, os documentos são transformados em vetores de palavras (também chamados de vetor de termos). Este processo é necessário pois possibilita a redução da dimensionalidade dos documentos (apenas as palavras mais significativas dos textos são utilizadas), e permite um processamento computacional mais eficaz.

Porém, o uso simples dos termos pode trazer interpretações ambíguas na contextualização dos documentos. Por exemplo, um texto que contenha a palavra "crime" não necessariamente trata de um crime. É necessário verificar outros elementos que compõem

um crime, para então afirmar se, de fato, o texto é relativo a um crime. A Figura 3.2 ilustra um exemplo.

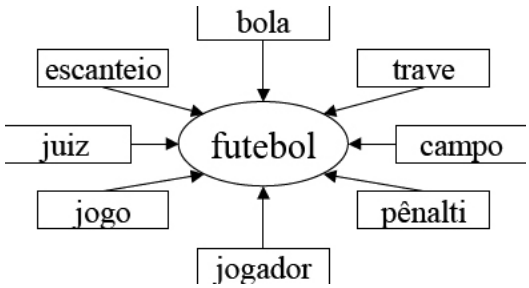


Figura 3.2: *Exemplo de um Conceito com seus termos relacionados*

Em geral, os conceitos são compostos por um identificador e um conjunto de termos que o descrevem (*hints*). O identificador dá a ideia geral do conceito e o conjunto de termos são os elementos que indicam a presença daquele conceito em um documento [114]. Na Figura 3.3 pode-se ver o exemplo da estrutura de um conceito.

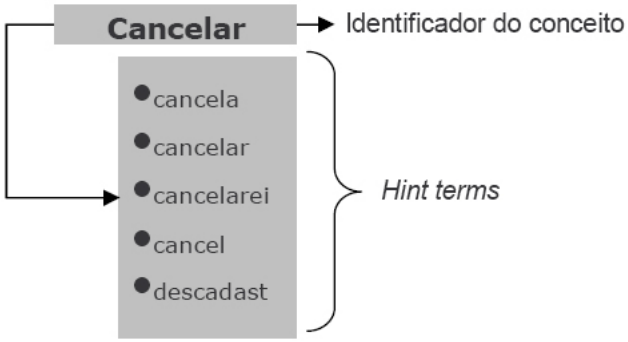


Figura 3.3: *Estrutura de um Conceito*

Diversas estruturas podem ser utilizadas para a representação do conceito (modelo vetorial, modelo contextual), mas segundo Guarino [38] as ontologias são as estruturas naturais para tal representação. Wives [114] cita outras formas de representação.

3.4 Cálculo de Similaridade entre Ontologia e Texto

Determinar a similaridade entre duas entidades (documentos, termos, conceitos) é um dos principais desafios para a mineração de textos. O processo de Recuperação da Informação possui, em suas etapas, este cálculo.

A escolha da medida de similaridade ideal está intimamente ligada ao domínio do problema em questão. Neste trabalho, o cálculo da similaridade foi utilizado para identificar no texto os conceitos presentes. A relação dos possíveis conceitos foi obtida através da Ontologia.

Após a recuperação dos conceitos armazenados na Ontologia, um vetor de termos para cada conceito é gerado. Para cada texto também é criado um vetor de termos. Realiza-se o cálculo da similaridade utilizando os vetores dos conceitos e o vetor do texto a fim de identificar quais conceitos realmente estão presentes no texto. A Figura 3.4 mostra a visão geral do cálculo de similaridade.

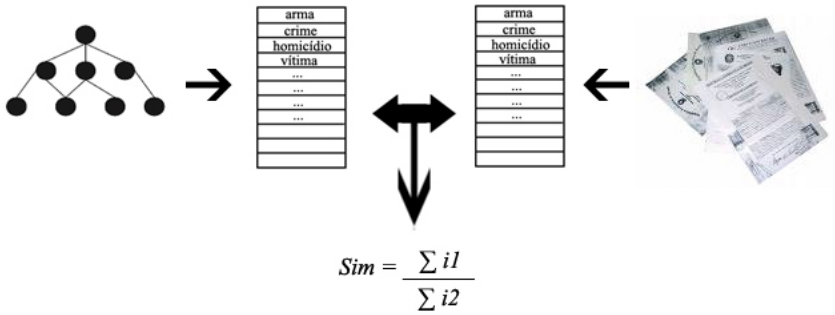


Figura 3.4: *Processo de cálculo de similaridade entre texto e Ontologia*

Segundo Aldenderfer et. al. [3], as medidas de similaridade podem ser classificadas em quatro grupos: medidas de distância, coeficientes de correlação, coeficientes de associação e medidas probabilísticas de similaridade. Ainda segundo ele, a similaridade deve ser medida de uma forma objetiva, reaplicável e quantitativa.

Pela sua alta empregabilidade e facilidade de aplicação, serão destacados dois tipos de medidas: as medidas de distância e os coeficientes de associação.

3.4.1 Medidas de distância

As medidas de distância se associam ao espaço euclidiano, em que cada coordenada corresponde a uma palavra. A similaridade é calculada com base na proximidade dos pontos, conforme mostra a Figura 3.5. O ponto “A” possui ambos os termos x e y. Esses termos estão contidos no ponto “B”, apesar de estarem em uma relevância baixa. Já o ponto “C” não contém um dos termos.

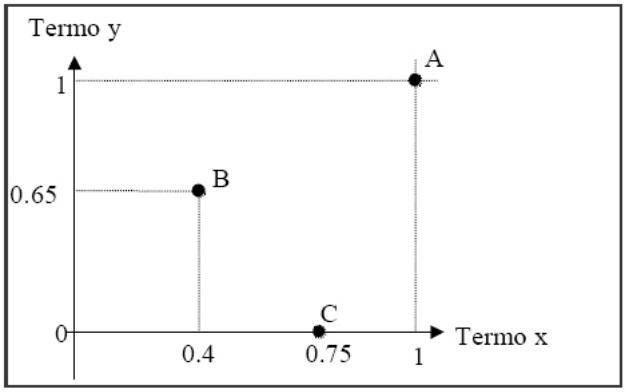


Figura 3.5: *Espaço euclidiano com os pontos*

Existem diversas formas de calcular a distância em um espaço euclidiano, as mais utilizadas são: a distância euclidiana e a função do coseno (*cosine*).

A distância euclidiana, apresentada na equação 3-1, cálcula a distância entre dois documentos i e j, com k termos. Quanto mais próximos os objetos estiverem, mais próximo de zero será o d_{ij}.

$$d_{ij} = 1 - \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2} \tag{3-1}$$

A função do coseno (utilizadas no modelo vetorial de recuperação de informaçao, conforme mostrado na seção 2.4.2), apresentada na equação 3-2, calcula a distância baseando-se no vetor de termos de cada documento. Assim, d e q são vetores de termos e w são os pesos dos termos em cada vetor.

$$sim(d,q) = \frac{\sum_{i=1}^{t} w_{id} \times w_{iq}}{\sqrt{\sum_{i=1}^{t} (w_{id})^2} \times \sqrt{\sum_{i=1}^{t} (w_{iq})^2}} \tag{3-2}$$

3.4.2 Coeficiente de Associação

Os coeficientes de associação são utilizados para estabelecer a similaridade entre termos de forma binária, ou seja, levando-se em consideração sua presença ou ausência no texto. Dentre os diversos coeficientes, destacamos os coeficientes *Jaccard* e *Overlap* [60].

No coeficiente *Jaccard*, apresentado na equação 3-3, aplicado em dois documentos X e Y, teríamos em "a" a quantidade de termos presentes em ambos os documentos, em "b" os termos presentes somente no documento X e em "c" os termos presentes somente no documento Y.

$$s = \frac{a}{a+b+c} \tag{3-3}$$

Já no coeficiente *Overlap*, apresentado na equação 3-4, aplicado em dois documentos X e Y, teríamos em "a" a quantidade de termos presentes em ambos os documentos, em "b" os termos presentes somente no documento X e em "c" os termos presentes somente no documento Y.

$$s = \frac{a}{min(b,c)} \tag{3-4}$$

Em ambos coeficientes, a similaridade varia de 0 a 1, onde 0 representa ausência de similaridade e 1 a similaridade total. A diferença central entre os dois coeficientes reside no fato de que o *Overlap* não leva em consideração os termos diferentes que existem entre os documentos. Assim, independente de quantos termos diferentes existam no documento Y, caso os termos do documento X forem encontrados, a similaridade será igual a 1. A escolha de qual medida usar dependerá do contexto inserido e da natureza dos documentos analisados.

CAPÍTULO 4

Regras de Associação

Neste capítulo são detalhados os conceitos relativos às Regras de Associação. A Seção 4.1 define, de forma descritiva, as regras de associação e, na Seção 4.2, é feita uma definição formal. Na Seção 4.3, é mostrado o algoritmo *Apriori*, utilizado para geração das regras. Na seção 4.4, são abordadas as medidas de interesse objetivas e subjetivas.

4.1 Mineração de Itens Frequentes

Tradicionalmente, os métodos de mineração de dados são divididos em aprendizado supervisionado (preditivo) e não-supervisionado (descritivo) [19] [27] [43]. Apesar do limite dessa divisão ser tênue (alguns métodos preditivos podem ser descritivos e vice-versa), ela ainda é interessante para fins didáticos [27]. Entretanto, já existem variações entre os dois tipos de aprendizados, como já foi relatado por Seliya [92] e Wang [109]], que propõem abordagens semi-supervisionadas.

A diferença entre os métodos de aprendizado supervisionados e não-supervisionados está no fato de que estes não precisam de uma categorização para cada registro, não sendo necessário um atributo alvo. Os autores classificam os diversos métodos de formas diferentes. Neste trabalho, será usada classificação adotada por Han et al. [43] para descrever o método de Regras de Associação.

A Mineração de Regras de Associação é uma das técnicas mais conhecidas de mineração de dados, devido ao problema da Análise da Cesta de Compras, que consiste em identificar o relacionamento dos itens mais frequentes em um determinado conjunto de dados, e permite obter resultados do tipo: SE *compra leite e pão* TAMBÉM *compra manteiga*. Esta construção recebe o nome de Regra de Associação (*Association Rules*). Na Figura 4.1 pode ser visto um exemplo de algumas regras.

Regra 1: SE *idade* = *jovem* AND *estudante* = *não* ENTÃO *compra computadores* = *não*
Regra 2: SE *idade* = *jovem* AND *estudante* = *sim* ENTÃO *compra computadores* = *sim*
Regra 3: SE *idade* = *média* ENTÃO *compra computadores* = *sim*
Regra 4: SE *idade* = *adulto* AND *avaliação de crédito* = *excelente* ENTÃO *compra computadores* = *sim*
Regra 5: SE *idade* = *adulto* AND *avaliuação de crédito* = *ruim* ENTÃO *compra computadores* = *não*

Figura 4.1: *Regra de associação*

Introduzido por Agrawal, Imielinski e Swami [1], a principal técnica de mineração de regras de associação, chamada Mineração de Itens Frequentes (*Frequent Itemset Mining*), consiste de duas etapas: primeiro, um conjunto de itens frequentes (*Frequent Itemset*) é criado, respeitando um valor mínimo de frequência para os itens (suporte e confiança); depois, as regras de associação são geradas pela mineração desse conjunto.

Para garantir resultados válidos, os conceitos de suporte e confiança são utilizados em cada regra produzida. A medida de suporte indica o percentual de registros (dentre todo o conjunto de dados) que se encaixam nessa regra. Já a confiança mede o percentual de registros que atendem especificamente a regra, por exemplo, o percentual de quem compra leite e pão e também compra manteiga.

Um dos mais tradicionais algoritmos de mineração utilizando a estratégia de itens frequentes é o *Apriori* [2]. Diversas variações deste algoritmo, envolvendo o uso de técnicas de *hash*, redução de transações, particionamento e segmentação, podem ser encontradas em [2].

4.2 Definição Formal

Em Agrawal et. al [2], temos que: seja $I = i_1, i_2, ..., i_m$ um conjunto de itens e D um conjunto de transações, onde cada transação T é um conjunto de itens tal que $T \subseteq I$. Associada com cada transação está um identificador único, chamado TID. Diz-se que a transação T contém X (um conjunto de itens em I), se $X \subseteq T$.

Uma regra de associação é uma implicação do tipo $X \Rightarrow Y$, onde $X \subset I$, $Y \subset I$ e $X \cup Y = \oslash$. A regra $X \Rightarrow Y$ aparece nas transações D com uma confiança c se $c\%$ das transações em D contém X e também Y. A regra $X \Rightarrow Y$ tem suporte s no conjunto de transações D se $s\%$ das transações em D contém $X \cup Y$.

Assim, dado um conjunto de transações D, o problema da mineração das regras de associação consiste em gerar todas as regras de associação que tiverem, respectivamente, suporte maior que o especificado pelo usuário (*minsup*) e confiança maior que a especificada pelo usuário ($minconf$).

Em um regra de associação do tipo $X \Rightarrow Y$, X é chamado de antecedente e Y de consequente.

4.3 Algoritmo Apriori

Esse algoritmo foi proposto em 1994 pela equipe de pesquisa do Projeto QUEST, da IBM, que originou o software *Intelligent Miner*. Trata-se de um algoritmo que resolve o problema da mineração de itens frequentes. É comumente utilizado para geração das regras de associação, conforme as propriedades descritas na Seção 4.2.

O algoritmo Apriori é dividido em duas etapas:

1. Encontrar todos os conjuntos de itens freqüentes, que satisfazem a condição de suporte mínimo;
2. A partir do conjunto de itens freqüentes, gerar as regras de associação, que satisfazem a condição de confiança mínima.

O algoritmo Apriori possui uma propriedade muito útil para geração do conjunto dos itens frequentes. É chamada Propriedade Apriori, ou antimonotonia do suporte, e pode ser definida como: Seja X e Y dois *itemsets* tal que $X \subseteq Y$. Se X é frequente então Y também é frequente. Assim, dada uma sequencia de *itemset* descobertos variando de 1 até k, um *k-itemset* somente será frequente se todos os seus *(k-1)-itemsets* também forem frequentes.

Como o algoritmo Apriori é executado de forma iterativa, ou seja, o *2-itemset* é calculado a partir do *1-itemset* e assim sucessivamente, não é necessário calcular o suporte do *k-itemset*, pois basta usar a propriedade da antimonotonia e verificar se os *itemset* anteriores são frequentes.

4.3.1 Etapas do algoritmo *Apriori*

Tomemos a Tabela 4.1, que contém algumas transações que representam os produtos adquiridos em compras feitas em um supermercado.

Inicialmente, o usuário deve definir os valores mínimos de suporte e confiança. Por exemplo: $minsup = 0,3$ e $minconf = 0,8$.

No primeiro passo, são identificados os *k-itemsets* frequentes (suporte $>$ $minsup$). Conforme mostra as tabelas 4.2, 4.3 e 4.4.

Transação	Leite	Café	Cerveja	Pão	Manteiga	Arroz	Feijão
1	Não	Sim	Não	Sim	Sim	Não	Não
2	Sim	Não	Sim	Sim	Sim	Não	Não
3	Não	Sim	Não	Sim	Sim	Não	Não
4	Sim	Sim	Não	Sim	Sim	Não	Não
5	Não	Não	Sim	Não	Não	Não	Não
6	Não	Não	Não	Não	Sim	Não	Não
7	Não	Não	Não	Sim	Não	Não	Não
8	Não	Não	Não	Não	Não	Não	Sim
9	Não	Não	Não	Não	Não	Sim	Sim
10	Não	Não	Não	Não	Não	Sim	Não

Tabela 4.1: *Exemplo de transações*

1-itemsets	Suportes
Leite	0,2
Café	0,3
Cerveja	0,2
Pão	0,5
Manteiga	0,5
Arroz	0,2
Feijão	0,2

Tabela 4.2: *1-itemsets*

2-itemsets	Suportes
Café, Pão	0,3
Café, Manteiga	0,3
Pão, Manteiga	0,4

Tabela 4.3: *2-itemsets*

A lista obtida com os *k-itemset* frequentes >= 2 contém as seguintes relações:

```
{Café, Pão}
{Café, Manteiga}
{Pão, Manteiga}
{Café, Pão, Manteiga}
```

No segundo passo são geradas as regras de associação. Assim, é necessário gerar as regras para cada conjunto de *itemset* frequente encontrado no passo anterior. Tem-se então, as seguintes regras:

```
Itemset(Café, Pão)
  café => pão (Confiança: 1,0)
  pão => café (Confiança: 0,6)

Itemset(Café, Manteiga)
  café => manteiga (Confiança: 1,0)
```

3-itemsets	Suportes
Café, Pão, Manteiga	0,3

Tabela 4.4: *3-itemsets*

```
  manteiga => café (Confiança: 0,6)

Itemset{Pão, Manteiga}
  pão => manteiga (Confiança: 0,8)
  manteiga => pão (Confiança: 0,8)

Itemset{Café, Pão, Manteiga}
  café, pão => manteiga (Confiança: 1,0)
  café, manteiga => pão (Confiança: 1,0)
  manteiga, pão => café (Confiança: 0,75)
  café => pão, manteiga (Confiança: 1,0)
  pão => café, manteiga (Confiança: 0,6)
  manteiga => café, pão (Confiança: 0,6)
```

Por fim, aquelas regras cuja confiança for inferior à confiança mínima definida pelo usuário são eliminadas. Tem-se, então, o seguinte conjunto de regras:

```
café => pão
café => manteiga
manteiga => pão
pão => manteiga
café, pão => manteiga
café, manteiga => pão
café => pão, manteiga
```

4.3.2 Algoritmo *Apriori*

Agrawal et. al [2] propõem a seguinte implementação do algoritmo Apriori 4.1.

Algoritmo 4.1: Algoritmo Apriori

```
Data: D
Result: L
L_1 ← 1 − itemsets ;                          // busca os k-itemset onde k = 1
for k ← 2 to L_{k−1} = ⊘ do
    k ← k + 1;
    C_k ← apriori-gen(L_{k−1}) ;              // Busca um novo k-itemset
    for t ∈ D do
        C_t ← subset(C_k, t);
        for c ∈ C_t do
            c.count++ ;                       // Calcula o suporte
    L_k ← c ∈ C_k | c.count ≥ minsup;
L ← ⋃ L_k ;                  // Faz a união dos itemsets selecionados
MostrarRegras(L);
```

4.4 Medidas de Interesse

As medidas de interesse servem para garantir a aplicabilidade de uma regra gerada, evitando que regras aleatórias, ou sem expressividade, sejam produzidas. Diversos são os mecanismos para auxiliar nesta tarefa, destacamos aqui as Medidas de Interesse Objetivas e as Medidas de Interesse Subjetivas [41].

O interesse no estudo destas medidas tem crescido, uma vez que o modelo Suporte/Confiança, que também é uma medida de interesse objetiva, tem apresentado resultados insatisfatórios, gerando um número grande de regras e, em alguns casos, regras que não são relevantes ou úteis.

4.4.1 Medidas de Interesse Objetivas

As medidas de interesse objetivas empregam índices estatísticos para avaliar a força de uma regra. Serão abordadas algumas das principais medidas de interesse objetivas, porém, outras medidas podem ser encontradas em [103] [41].

4.4.1.1 Modelo Suporte e Confiança

O Modelo Suporte/Confiança tem recebido muitas críticas ao longo dos últimos anos. O número de regras geradas pelo modelo geralmente é muito grande, dificultando o

processo de análise por parte do usuário.

O modelo típico para mineração de regras de associação em bases de dados consiste em encontrar todas as regras que possuam suporte e confiança maiores ou iguais, respectivamente, a um suporte mínimo (*minsup*) e uma confiança mínima (*minconf*), especificados pelo usuário.

O suporte de um conjunto de itens D, *sup(D)*, representa o percentual de transações T tal que $D \subseteq T$. Assim, o suporte da regra $X \Rightarrow Y$ é dado por $sup(X \cup Y)$. A confiança representa o percentual de transações que contém X e que também contém Y, assim, $conf(X \Rightarrow Y) = \frac{sup(X \cup Y)}{sup(X)}$.

4.4.1.2 *Lift*

A medida de *Lift*, apresentada por Brian et. al. [12], também conhecida como *interest*, é utilizada para avaliar a dependência entre o antecedente e o consequente.

A fórmula usada para o cálculo é dada pela fórmula:

$$Lift(X \Rightarrow Y) = \frac{Conf(X \Rightarrow Y)}{Sup(Y)} \tag{4-1}$$

A medida tende a um ($Lift(X \Rightarrow Y) = 1$), caso X e Y sejam independentes. Se $Lift(X \Rightarrow Y) > 1$, então X e Y são positivamente dependentes. Se $Lift(X \Rightarrow Y) < 1$, então X e Y são negativamente dependentes. Quanto maior o *lift* menos obvia é a regra.

4.4.1.3 Novidade

Introduzida por Shapiro et. al. [82], esta medida, também chamada de *Rule Interest - RI* ou *leverage*, indica o valor da diferença entre o suporte real e o suporte esperado. Entende-se por suporte esperado a soma dos suportes dos itens que compõem a regra, ou seja: $Sup(X \Rightarrow Y) = Sup(X \cup Y)$. A fórmula usada para o cálculo é dada por:

$$Novidade(X \Rightarrow Y) = Sup(X \Rightarrow Y) - SupEsp(X \Rightarrow Y) \tag{4-2}$$

Se $Novidade(X \Rightarrow Y) = 0$, então X e Y são independentes. Se $Novidade(X \Rightarrow Y) > 0$ então X e Y são positivamente dependentes. Se $Novidade(X \Rightarrow Y) < 0$ então X e Y são negativamente dependentes. A medida de *levarage* possui variação de -0.25 e 0.25. Quanto maior a medida, mais interessante, inovadora e não-usual é a regra.

4.4.1.4 Convicção

Também introduzida por Brin et. al. [12], a medida de convicção representa o quão significativa é a associação entre o antecedente e o consequente. A ideia é avaliar a regra como uma implicação. Utiliza-se a seguinte fórmula para o cálculo desta medida:

$$Conv(X \Rightarrow Y) = \frac{Sup(X)xSup(\neg Y)}{Sup(X \cup \neg Y)} \tag{4-3}$$

A medida de convicção possui as seguintes características:

- São levados em conta, tanto o suporte do antecedente quanto o do consequente;
- Caso X e Y sejam independentes, $Conv(X \Rightarrow Y) = 1$;
- Caso sempre ocorra X e Y, $Conv(X \Rightarrow Y) = \infty$.

Brin et. al. [12] realizaram uma avaliação através da mineração de uma base de dados censitários com mais de 20.000 regras de associação. Foi constatado que o valor de convicção das melhores regras varia de 1,01 a 5 e que regras com valor de convicção acima de 5 representavam informações óbvias ou ilusórias.

4.4.2 Medidas de Interesse Subjetivas

As medidas de interesse subjetivas consideram principalmente a opinião de um analista para determinar a força da regra. Silberschatz e Tuzhilin [96] classificaram as medidas de interesse subjetivas baseando-se em dois conceitos principais:

Inesperabilidade (*Unexpectedness*) O conhecimento é interessante se ele é novo para o usuário ou contradiz seu conhecimento prévio.

Utilidade (*Actionability*) O conhecimento é interesse se o usuário pode tomar alguma decisão baseado nele.

A grande maioria das pesquisas concentra-se na avaliação da inesperabilidade, pelo fato da avaliação da utilidade ser muito abstrata e pessoal. Além disto, com o foco na inesperabilidade espera-se atingir a utilidade [99].

Liu et. al. [56] propõem quatro medidas para identificar as regras de associação esperadas e inesperadas, levando em consideração o conhecimento prévio de um especialista sobre um domínio. As medidas são definidas em relação ao grau com que o antecedente e o consequente das regras encontradas conferem com o antecedente e o consequente informados pelo usuário. Os valores variam de 0, sem conformidade, até 1, conformidade total.

Seja A_{ij} o fator que mede o quanto o antecedente i da regra está em conformidade com o antecedente j fornecido, e C_{ij} o fator que mede o quanto o consequente i da regra está em conformidade com o consequente j fornecido, definem-se as medidas como:

Conformidade Indica se a regra está em conformidade com o esperado pelo especialista.

$$conf_{ij} = A_{ij} \cdot C_{ij} \tag{4-4}$$

Antecedente Inesperado Indica se o antecedente é inesperado.

$$antInes_{ij} = \begin{cases} 0, & C_{ij} - A_{ij} \leq 0 \\ C_{ij} - A_{ij}, & C_{ij} - A_{ij} > 0 \end{cases} \tag{4-5}$$

Consequente Inesperado Indica se o consequente é inesperado.

$$consInes_{ij} = \begin{cases} 0, & A_{ij} - C_{ij} \leq 0 \\ A_{ij} - C_{ij}, & A_{ij} - C_{ij} > 0 \end{cases} \tag{4-6}$$

Antecedente e Consequente Inesperados Indica se o antecedente e o consequente são inesperados.

$$antConInes_{ij} = 1 - \max[conf_{ij}, antInes_{ij}, consInes_{ij}] \tag{4-7}$$

A forma de cálcular os valores do A_{ij} e do C_{ij} dependem de como o conhecimento será fornecido pelo especialista. Segundo Liu et. al. [56], o conhecimento pode ser especificado na forma de impreciso ou de uma impressão geral. O conhecimento impreciso possibilita ao especialista informar suposições que ele acredita serem verdadeiras, e a impressão geral informa a relação que o especialista acredita existir entre os itens das regras.

4.4.2.1 Impressão Geral

Assim, seja *nAnt* o número de itens do antecedente da regra descoberta e *nCon* o número de itens do consequente da mesma. Seja *contAnt* o número de itens do antecedente que conferem com aqueles informados pelo especialista e *contCon* o número de itens do consequente que conferem com aqueles informados pelo especialista. Seja $nItens$ o número total de itens da regra informada pelo especialista (somando antecedente e consequente). Seja $nContItens$ o número total de itens da regra gerada (somando antecedente e consequente) que conferem com aquelas informadas pelo especialista (somando antecedente e consequente). E se $nItens = 0$ então $\frac{nContItens}{nItens} = 1$. Temos que:

Se

$$\frac{contAnt}{nAnt} > \frac{contCon}{nCon} \tag{4-8}$$

Então

$$A_{ij} = \min(\frac{contAnt}{nAnt}, \frac{nContItens}{nItens}) \tag{4-9}$$

$$C_{ij} = \frac{contCon}{nCon} \tag{4-10}$$

Senão

$$C_{ij} = \min(\frac{contCon}{nCon}, \frac{nContItens}{nItens}) \tag{4-11}$$

$$A_{ij} = \frac{contAnt}{nAnt} \tag{4-12}$$

4.4.2.2 Conhecimento Impreciso

Assim, seja *nAnt* o número de itens do antecedente da regra descoberta e *nCon* o número de itens do consequente da regra descoberta. Seja *contAnt* o número de itens do antecedente que conferem com aqueles informados pelo especialista e *contCon* o número de itens do consequente que conferem com aqueles informados pelo especialista. Seja $nItensAnt$ o número de itens do antecedente na regra informada pelo especialista e $nItensCon$ o número de itens do consequente na regra informada pelo especialista. Seja $nContItensAnt$ o número de itens do antecedente da regra informada pelo especialista que conferem com aqueles da regra gerada e $nContItensCon$ o número de itens do consequente da regra informada pelo especialista que conferem com aqueles da regra gerada. E se $nItensAnt = 0$ ou $nItensCon = 0$ então $\frac{nContItensAnt}{nItensAnt} = 1$ ou $\frac{nContItensCon}{nItensCon} = 1$. Temos que:

$$A_{ij} = \min(\frac{contAnt}{nAnt}, \frac{nContItensAnt}{nItensAnt}) \tag{4-13}$$

$$C_{ij} = \min(\frac{contCon}{nCon}, \frac{nContItensCon}{nItensCon}) \tag{4-14}$$

CAPÍTULO 5

Metodologia Proposta

Neste capítulo, é proposta uma metodologia para auxiliar na geração de regras de associação envolvendo dados estruturados e não-estruturados. Na Seção 5.1 são apresentados alguns trabalhos correlatos. Na Seção 5.2 a metodologia proposta é detalhada.

5.1 Trabalhos Relacionados

A seguir, destacam-se alguns trabalhos da bibliografia pesquisada que tem relação com a proposta deste trabalho.

5.1.1 *Interestingness Analysis System - IAS*

Liu et. al. [56] propõem um sistema chamado *Interestingness Analysis System - IAS* com o objetivo de automatizar o processo de análise das regras de associação produzidas.

A ferramenta proposta utiliza uma técnica de pós-processamento interativa e iterativa, ou seja, ela atua após a geração das regras e de forma a permitir a interatividade com o usuário. Basicamente é composta de três componentes:

1. Uma linguagem que permite a representação do conhecimento do usuário;
2. Um sistema de análise de interesse que permite analisar as regras de associação produzidas identificando os valores de conformidade, antecedente inesperado, consequente inesperado e antecedente e consequente inesperado;
3. Um sistema de visualização que permite a detecção visual de regras interessantes.

Segundo a proposta, o IAS funciona de maneira circular: o usuário modela o conhecimento sobre um determinado domínio; o sistema, após gerar as regras, identifica aquelas que sejam interessantes com base no conhecimento informado; o usuário analisa as regras de forma visual, removendo aquelas irrelevantes, faz os ajustes necessários e inicia novamente o processamento. O processo continua até que todas as regras geradas sejam consideradas relevantes.

O sistema IAS difere da metodologia proposta neste trabalho, pois, apesar de trabalhar com as medidas de interesse subjetivas, não leva em consideração dados não-estruturados e utiliza-se apenas da medida de interesse objetiva Suporte/Confiança.

5.1.2 Extração de regras de associação de dados textuais

Santos [91] propõe a extração das regras de associação diretamente de textos. O método proposto visa analisar as relações implícitas entre as palavras de um conjunto de documentos. Assim, as regras de associação geradas são baseadas no relacionamento existente entre os termos dos documentos minerados.

A ideia consiste em extrair de cada texto um conjunto de palavras, que serão dispostas de maneira tabular, e então o algoritmo de mineração de regras de associação *Apriori* será executado.

Para extração do conjunto de palavras que representará cada documento, foram utilizadas as técnicas de Processamento de Linguagem Natural (PLN) a fim de identificar todos os substantivos presentes nos textos. Após o processamento de todos os textos, a medida TF-IDF foi utilizada para identificar quais palavras de fato eram relevantes.

Para validação das regras obtidas, foi utilizada a plataforma *WordNet* [63] como uma medida de interesse. Assim, a relação entre as palavras foi confrontada com os termos do *WordNet* e foi verificado se, de fato, as regras produzidas eram interessantes. Para isso, calculou-se o número de arestas que ligam duas palavras na hierarquia do *WordNet*. Quanto maior o número de arestas mais interessante é a regra.

Esta proposta, apesar de trabalhar com a mineração de dados não-estruturados, não possibilita a combinação com dados estruturados. Além disso, são extraídos de forma direta dos textos apenas os substantivos. O conhecimento do usuário sobre um determinado domínio não é levado em consideração. Outra característica não abordada pela proposta refere-se à não utilização das medidas de interesse objetivas e subjetivas.

5.1.3 Utilização de uma Ontologia na melhora do valor de suporte

Chen et. al. [17] propõem uma abordagem para melhorar o valor da medida de interesse objetiva “Suporte” das regras de associação mineradas.

A proposta sugere a utilização de uma Ontologia de domínio que automatiza o processo de análise das regras de associação produzidas e, de acordo com a relação do antecedente e consequente mapeada, sugere a substituição das regras mais específicas por regras mais genéricas.

Assim, com regras mais genéricas, espera-se aumentar o valor de suporte das regras de tal forma que aquelas com valores inferiores aos mínimos definidos sejam eliminadas sem a necessidade de intervenção humana.

Apesar de trabalhar com a mineração de regras de associação, a proposta não visa minerar dados estruturados e não-estruturados, bem como não utiliza outras medidas de interesse objetivas e subjetivas para filtrar as regras.

5.1.4 RulEE-SEAR

Sinoara [99] propõe uma metodologia para identificação de regras de associação interessantes através da combinação de medidas de interesse objetivas e subjetivas na filtragem das regras de associação produzidas. Utiliza-se o algoritmo *Apriori* para geração das mesmas.

Para validação do método proposto, foi desenvolvida uma ferramenta chamada RulEE-SEAR (*Rule Exploration Environment - Subjective Exploration of Association Rules*), baseada no ambiente RulEE (*Rule Exploration Environment*). O ambiente RulEE possibilita a filtragem das regras de associação por meio das medidas de interesse objetivas. A proposta deste trabalho foi agregar ao resultado do ambiente RulEE uma nova filtragem utilizando as medidas de interesse subjetivas.

O trabalho, apesar de combinar as medidas de interesse objetivas e subjetivas, não aborda o processo de mineração em dados estruturados e não-estruturados.

5.1.5 Mineração em dados estruturados e não-estruturados

Barth et. al. [6] propõem um sistema para processamento de dados armazenados em fontes estruturadas e não-estruturadas. A recuperação dos dados ocorre através de uma consulta inicial feita pelo usuário, que é então expandida com o auxílio de uma Ontologia de domínio.

Os dados recuperados são submetidos a algoritmos de agrupamento (*cluster*) e os grupos identificados são representados através de entidades nomeadas. Com os resultados obtidos, a ferramenta gera os vínculos[1] entre as entidades, o que pode ser analisado de forma gráfica.

Para validação do sistema proposto, foi utilizado um ambiente de investigação criminal contendo dados de escutas telefônicas, boletins de ocorrências e notícias da Internet.

Apesar de trabalhar com dados estruturados e não-estruturados a proposta deste trabalho não utiliza o conhecimento do usuário para o processamento dos dados textuais. Além disso, trabalha com a técnica de agrupamento ao invés das regras de associação.

[1] Análise de vínculo é o estudo a respeito dos alvos e suas relações.

5.1.6 Classificação de jurisprudências

Morais [64] propõe um sistema para classificação automática de textos. Inicialmente, os documentos são recuperados baseando-se em uma consulta feita pelo usuário. Em seguida, é calculada a similaridade entre os documentos recuperados e as Ontologias de domínio criadas. Assim, é possível classificar um determinado documento baseando-se na similaridade com a Ontologia.

Para validação da ferramenta proposta, foi utilizada a base de dados de jurisprudências[2] do Tribunal de Justiça de Goiás e as Ontologias criadas, cada uma contendo o conhecimento sobre determinada lei. Assim, foi possível classificar de forma automática as jusrisprudências quanto à matéria tratada por elas.

Apesar de utilizar as Ontologias no processo de análise dos textos, este trabalho não realiza a mineração de regras de associação, bem como não trabalha com a mineração de dados estruturados e não-estruturados.

5.1.7 Análise de diagnósticos médicos

Em [47], Holzinger et. al. utilizam a mineração de textos para auxiliar na análise de diagnósticos médicos. A proposta consiste em desenvolver um sistema que analisa, através da mineração de dados, as observações feitas por especialistas sobre as imagens de ressonâncias magnéticas, e, com auxílio de Ontologias que mapeiam o conhecimento sobre diversas doenças, identificam correlações entre as informações mineradas e as Ontologias.

O trabalho propõe uma aplicação voltada para Internet, que permite aos usuários, além de solicitar o processo de análise descrito acima, configurar o ambiente da aplicação, possibilitando a inclusão de termos e doenças, bem como obter um histórico das análises realizadas.

Apesar de trabalhar com a mineração de dados não-estruturados e utilizar-se de uma Ontologia, o sistema proposto não visa minerar regras de associação.

5.2 Metodologia Proposta

Este trabalho propõe uma metodologia para minerar regras de associação em repositórios que contenham dados estruturados e não-estruturados. No tratamento dos dados não-estruturados, uma Ontologia é utilizada para identificar informações relevantes. Para filtragem das regras de associação produzidas é utilizada uma combinação entre as medidas de interesse objetivas *lift*, *convicção* e *novidade*, e as medidas de interesse

[2] O hábito de interpretar e aplicar as leis aos fatos concretos, para que, assim, se decidam as causas.

subjetivas *conformidade*, *antecedente inesperado*, *consequente inesperado* e *antecedente e consequente inesperado*.

A metodologia proposta pode ser aplicada a qualquer área, desde que se tenha a necessidade da combinação de dados estruturados e não-estruturados, além do conhecimento de especialistas para criação da Ontologia. Apesar de não ser objeto da metodologia proposta, a criação da Ontologia faz-se necessária para viabilizar sua aplicação em situações reais.

Apesar de não propor um novo algoritmo de mineração de regras de associação, a metodologia proposta visa orquestrar de forma inovadora algumas tecnologias consolidadas. Destaca-se o uso da mineração de dados, de textos e as ontologias. O fiuxo proposto permite ao usuário, além da automatização do processo de geração de regras de associação utilizando dados estruturados e não-estruturados, a interação direta com os resultados obtidos.

Espera-se que seja possível combinar dados do tipo estruturado e não-estruturado e identificar padrões implícitos através da mineração de regras de associação. Além disto, a combinação das medidas de interesse objetivas e subjetivas para filtragem dos resultados possibilita que o usuário utilize o conhecimento pré-existente sobre um determinado domínio.

Apesar da existência de diversos algoritmos para mineração das regras de associação, optou-se por trabalhar com o algoritmo *Apriori* por sua robustez computacional e aos bons resultados obtidos com sua aplicação em repositórios contendo dados qualitativos e estruturados [15].

Devido à grande quantidade de medidas de interesse disponíveis, optou-se por trabalhar em duas linhas: medidas de interesse objetivas, visando obter regras gerais, confiáveis e não redundantes, e medidas de interesse subjetivas, visando obter regras novas e úteis [54]. Neste sentido, para escolha das medidas de interesse objetivas buscou-se analisar a independência, a dependência positiva e a dependência negativa entre os itens [32]. Para escolha das medidas subjetivas foram analisadas a inesperabilidade e a conformidade [31].

A combinação das medidas objetivas e subjetivas visa proporcionar aos usuários a geração das regras através de duas ópticas: com as medidas objetivas pode-se avaliar as regras independente do domínio e, com as medidas subjetivas, aquelas regras que, mesmo interessantes, não são aplicáveis a um determinado domínio [99]. Além disto, a combinação das medidas possibilita que as regras sejam filtradas de maneira interativa, uma vez que, de acordo com os resultados, novos valores são definidos, e participativa, uma vez que o usuário pode infiuenciar o resultado final com conhecimentos pré-existentes.

Optou-se por representar o conhecimento na Ontologia através de conceitos. A representação do conhecimento através dos conceitos permite diminuir o problema

da ambiguidade na interpretação semântica [114]. Assim, um conceito é representado por termos que podem melhor refletí-lo. Por exemplo: o conceito "Arma de Fogo" é representado pelos termos "Disparo", "Bala", "Tiros", "PAF (Projétil de Arma de Fogo)", "Alvejada", etc. Desta forma, para que o conceito seja considerado presente em um texto, ele deve conter um número mínimo de termos ao invés de apenas o termo chave "Arma de Fogo".

Diferente de outras abordagens, esta proposta pretende automatizar a integração dos dados estruturados e não-estruturados utilizando-se de uma Ontologia de domínio que expressa, através dos conceitos, o conhecimento do usuário sobre um assunto. A Figura 5.1 apresenta o fluxo geral da metodologia proposta.

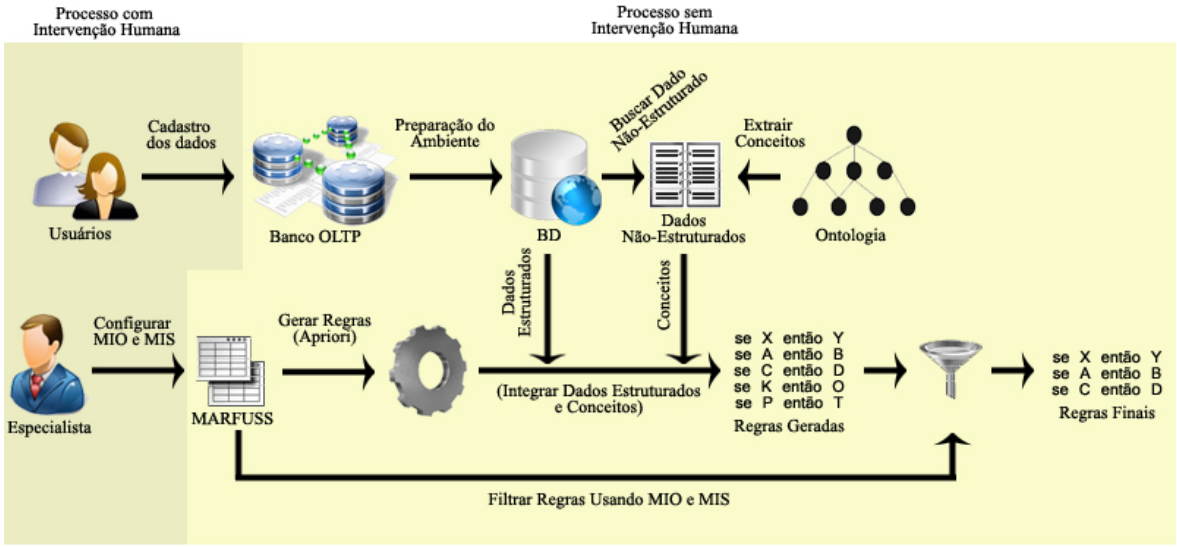


Figura 5.1: *Fluxo geral da metodologia proposta*

O método proposto é composto de duas etapas:

Preparação do ambiente Esta etapa compreende a coleta dos dados nos diversos repositórios disponíveis (banco de dados, página da internet, arquivos de textos, *webservices*, bibliotecas digitais, etc.) e as tarefas de limpeza, redução e transformação destes dados. Como resultado desta etapa, cria-se um repositório central, com os dados padronizados e que será utilizado no processo de mineração.

Processamento Nesta etapa, inicialmente são extraídos os termos mais significativos dos dados não-estruturados (documentos) com o auxílio de uma Ontologia. Em seguida, os termos obtidos são combinados com outros dados estruturados. Por fim, as regras de associação são mineradas e filtradas com base na combinação das medidas de interesse objetivas e subjetivas.

A Figura 5.2 apresenta uma visão geral da metodologia proposta.

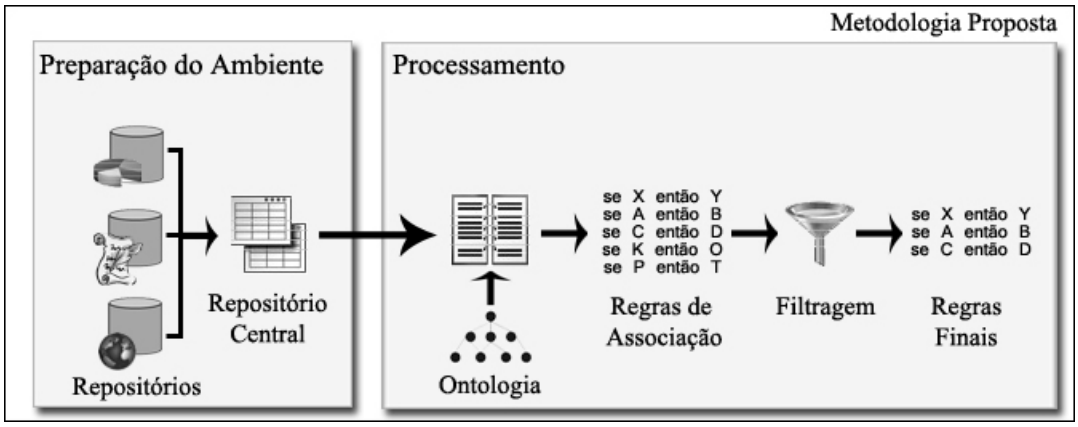


Figura 5.2: *Visão geral da metodologia proposta*

A etapa "Preparação do Ambiente" funciona como uma espécie de pré-processamento dos dados, eliminando as inconsistências e criando um repositório próprio para que as regras de associação possam ser mineradas.

Na etapa "Processamento", utiliza-se uma Ontologia para mapear o conhecimento do especialista sobre um determinado domínio, através dos conceitos, possibilitando, assim, a análise automatizada dos dados não-estruturados.

A seguir, são detalhadas as duas etapas da metodologia proposta.

5.2.1 Preparação do ambiente

Nesta etapa, dados obtidos de diversas fontes são consolidados. Como os dados podem ser obtidos de diversas fontes, tais como banco de dados, arquivos textos e documentos, é comum que o formato dos dados em cada uma dessas fontes não seja uniforme. Assim, antes de executar o algoritmo de mineração de regras de associação é necessário realizar algumas atividades visando padronizar estes dados.

A Figura 5.3 mostra o processo de preparação do ambiente. Inicialmente, os dados recuperados das diversas fontes são integrados em um repositório central. Durante a integração, os dados são lidos e armazenados em uma nova estrutura, criando valores de domínio para cada atributo gerado. Por exemplo, para o campo "SEXO" teríamos os valores "Masculino", "Feminino".

Após a integração, são executadas algumas atividades visando melhorar a qualidade dos dados: uma análise do repositório criado, a fim de eliminar atributos desnecessários e reduzir a dimensionalidade dos dados, uma limpeza nos dados visando eliminar valores incorretos, tais como nulos e vazios e, por fim, a transformação dos valores quantitativos em qualitativos, uma vez que os algoritmos de mineração de regras de associação necessitam deste formato. A transformação mapeia valores numéricos em descritivos, como por exemplo: o campo idade que possui o valor "32", será mapeado para

"Adulto". Ao final, a estrutura criada estará preparada para execução dos algoritmos de mineração.

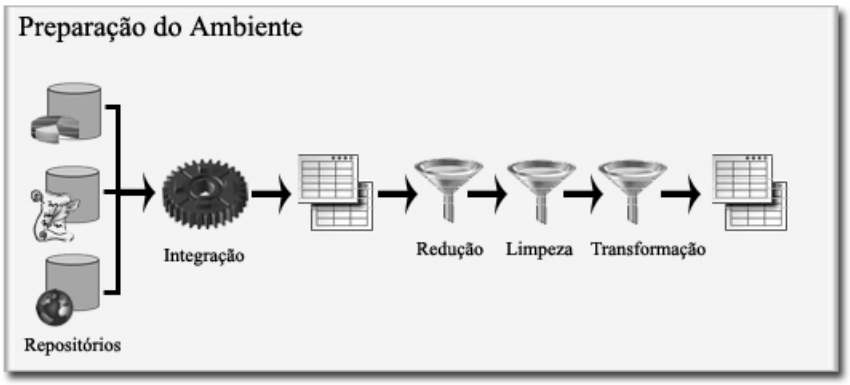


Figura 5.3: *Etapa: Preparação do ambiente*

Apesar de consolidar os atributos dos diversos repositórios e realizar algumas atividades visando melhorar a qualidade dos dados armazenados, nesta etapa, apenas os dados estruturados são processados. Os atributos textuais (não-estruturados) serão processados na etapa seguinte. Diversas técnicas e ferramentas de ETL (*Extract, Transform and Load*) [84] podem ser utilizadas para esta etapa.

5.2.2 Processamento

Nesta etapa, é realizada a mineração dos registros em busca das regras de associação. No entanto, antes da execução dos algoritmos, faz-se necessário processar os dados não-estruturados de maneira a extrair os conceitos presentes. Uma vez extraídos, esses valores são convertidos para uma forma estruturada (em que atributo=valor) e combinados com os demais dados estruturados do repositório criado na etapa anterior. Por fim, os algoritmos de mineração de regras de associação são executados. Sob o resultado gerado, são utilizadas as medidas de interesses para filtragem das regras irrelevantes. A Figura 5.4 ilustra este processo.

Durante a etapa de processamento, duas atividades chaves são executadas para que seja possível automatizar o processo de geração das regras de associação: extração de conceitos e filtragem das regras. Estes pontos serão analisados a seguir.

5.2.2.1 Extração de conceitos

Nesta atividade os conceitos definidos na Ontologia são identificados nos dados não-estruturados e recuperados. O algoritmo seguinte descreve os passos desta atividade:

```
1 ArquivoOntologia = Ler arquivo ontologia.owl
```

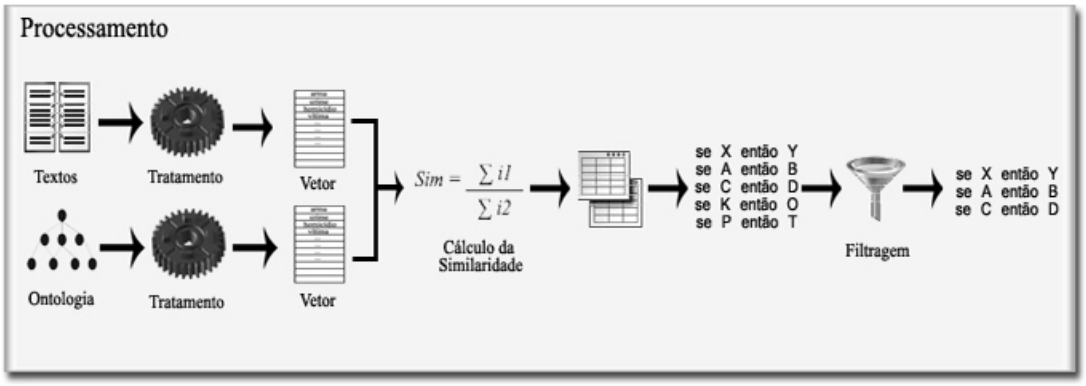


Figura 5.4: *Etapa: Processamento*

```
2  ListaStopWords = Ler arquivo stopWords.txt
3
4  VetorConceitos = []
5
6  ListaConceitos = BuscarListaConceitos(ArquivoOntologia)
7
8  Para cada item da ListaConceitos
9    VetorTermos = BuscarTermosConceito(ListaConceitos[item])
10   Para cada item do VetorTermos
11     Se VetorTermos[item] presente na ListaStopWords
12       VetorTermos[item] = nulo
13     Senão
14       VetorTermos[item] = Stemming(VetorTermos[item])
15     Fim se
16   Fim para
17   Adicionar VetorTermos em VetorConceitos
18 Fim para
19
20 Para cada registro do repositório
21   VetorConceitosPresente = []
22   DadoNaoEstruturado = Buscar campo não-estruturado
23   VetorTermosDoc = Separar cada palavra do DadoNaoEstruturado
24
25     Para cada item do VetorTermosDoc
26       Se VetorTermosDoc[item] presente na ListaStopWords
27         VetorTermosDoc[item] = nulo
28       Senão
29         VetorTermosDoc[item] = Stemming(VetorTermosDoc[item]);
30       Fim se
31     Fim para
32
33     Para cada item do VetorConceitos
34       minSimilaridade = Buscar o valor mínimo especificado
```

```
35                          na Ontologia para a similaridade
36          ValorSimilaridade = Overlap(VetorConceitos[item], VetorTermosDoc)
37          Se ValorSimilaridade >= minSimilaridade
38            Adiciona VetorConceitos[item] no VetorConceitosPresentes
39          Fim Se
40        Fim para
41  Fim para
```

O algoritmo recebe como entrada dois arquivos que contém, respectivamente, a definição da Ontologia e uma lista de palavras irrelevantes (*stopwords*).

Com base no arquivo que contém a Ontologia, os conceitos definidos são extraídos. Para cada conceito é gerado um vetor de termos que contém as palavras que o identificarão nos textos. Os dados não-estruturados também são processados e para cada um é gerado um vetor de termos. Sobre as palavras dos vetores gerados é realizado o *stemming*, visando eliminar as variações ortográficas das palavras, e a remoção de *stopwords*, visando eliminar as palavras irrelevantes.

Uma vez que os vetores de termos tenham sido tratados, a similaridade é calculada. O coeficiente *Overlap* é calculado sob o vetor de termos do documento e todos os vetores de termos dos conceitos. Um conceito é considerado presente se o vetor de termos que o representa possui valor de similaridade acima do mínimo definido.

Considerando que a Tabela 5.1 representa o vetor de termos do documento, em que cada célula é uma posição do vetor, e que as Tabelas 5.2, 5.3 e 5.4 representam, cada uma, o vetor de termos de cada conceito, o cálculo da similaridade com o coeficiente *Overlap*, utilizando a equação 3-4, se daria da seguinte forma:

- A similaridade entre o vetor de termos do documento e o conceito “ArmaDeFogo” será: $0/0 = 0$, uma vez que nenhum dos termos do vetor do conceito é encontrado no vetor de termos do documento;
- A similaridade entre o vetor de termos do documento e o conceito “Homicídio” será: $2/8 = 0.25$, sendo “2” a quantidade de termos comum entre os vetores e “8” o tamanho do menor vetor;
- A similaridade entre o vetor de termos do documento e o conceito “Vingança” será: $1/3 = 0.33$, sendo “1” a quantidade de termos comum entre os vetores e “3” o tamanho do menor vetor;

Após testes realizados com outras medidas de similaridade (*Jaccard* e *cosine*), a medida *Overlap* foi escolhida por não levar em consideração a quantidade de termos diferentes existente entre os vetores. Esta característica é fundamental uma vez que no contexto deste trabalho o objetivo é identificar se o vetor de termos do conceito está contido no vetor de termos do documento, e não se ambos os vetores são iguais.

Comparece	nesta	Delegacia	comunicando-nos	encontra-se	residencia
surpreendida	chegada	residencia	equipe	policia	militar
policiais	informaram	haviam	assassinado	sobrinho	Rua
Setor	Pedro	Ludovico	informa	usuario	drogas
estando	totalmente	dependente	vicio	Expediu-se	guia
requisição	exame	cadavérico	Registrou-se	devidos	fins

Tabela 5.1: *Vetor de termos do documento*

Atirar	Bala	Baleado
Disparos	PAF	Tiros

Tabela 5.2: *Vetor de termos do conceito "ArmaDeFogo"*

Assassinado	Autor	Cadáver
Corpo	Morrer	Morte
Óbito	Vítima	

Tabela 5.3: *Vetor de termos do conceito "Homicídio"*

Drogas	Entorpecentes	Tráfigo

Tabela 5.4: *Vetor de termos do conceito "Vingança"*

5.2.2.2 Filtragem das regras

Nesta atividade as regras de associação são geradas e filtradas. O seguinte algoritmo mostra os passos necessários:

```
1  minLift, minNovidade, minConvicacao = valor definido pelo usuário
2  minconf, minAntInes,
3      minConsInesp, minAntConInes = valor definido pelo usuário
4
5  ListaRegrasFinais = []
6  ListaRegras = Apriori()
7
8  Para cada item da ListaRegras
9    Se regra.lift < minLift OU
10       regra.novidade < minNovidade OU
11       regra.convicção < minConviccao
12     ListaRegras[item] = nulo
13   Senão
14     Se regra.conformidade < minConf OU
15        regra.antecedenteInesperado < minAntInes OU
16        regra.consequenteInesperado < minConsInesp OU
17        regra.anteConsInesperado < minAntConInes
18       ListaRegras[item] = nulo
19     Senão
20       Adiciona ListaRegras[item] em ListaRegrasFinais
21     Fim se
```

```
22    Fim Se
23  Fim para
24
25  Imprime ListaRegrasFinais
```

O algoritmo recebe como entrada os valores mínimos definidos pelo usuário para as medidas de interesse objetiva e subjetiva.

Após a geração das regras através do algoritmo *Apriori*, os valores calculados das medidas de interesse objetivas são confrontados com os valores mínimos definidos pelo usuário. Aquelas regras com valores inferiores aos mínimos definidos são descartadas. Para as regras que satisfizerem esta condição, os valores das medidas de interesse subjetivas serão calculados. Da mesma forma, estes valores serão confrontados com os valores mínimos definidos pelo usuário e, caso estejam abaixo do mínimo, a regra será descartada. O processo se repetirá até que todas regras tenham sido analisadas.

Optou-se por utilizar as medidas objetivas novidade, interesse e convicção, além do suporte e confiança, por serem métricas consolidadas junto às regras de associação. Para as medidas subjetivas, foram escolhidas conformidade, antecedente inesperado, consequente inesperado e antecedente/consequente inesperado visando obter regras inesperadas e úteis a um determinado domínio. Outras medidas podem ser vistas em [31].

Ao final, apenas as regras que atenderem às condições de valores mínimos definidos pelo usuário serão consideradas e apresentadas ao usuário. Observa-se que, para geração das regras, foram levados em consideração tanto os dados estruturados, resultantes da etapa “Preparação do Ambiente”, quanto os conceitos extraídos dos textos nesta etapa.

CAPÍTULO 6

Desenvolvimento do Sistema

Este capítulo apresenta as ferramentas utilizadas em um estudo de caso real para implementar as etapas da metodologia propostas no Capítulo 5, possibilitando assim sua validação. A Seção 6.1 detalha o Estudo de Caso proposto para aplicação da ferramenta. Na Seção 6.2 são abordadas as ferramentas utilizadas para o desenvolvimento da ferramenta. Por fim, na Seção 6.3 o funcionamento do sistema é mostrado.

6.1 Estudo de Caso

Como estudo de caso, foram utilizados os dados de boletins de ocorrência da Secretaria de Segurança Pública do Estado de Goiás. A escolha deste contexto deve-se à familiaridade e possibilidade de acesso às informações. Além disso, existe, por parte da instituição, a demanda por soluções que auxiliem os Analistas Criminais no processo de análise das ocorrências.

Todo o processo de análise das ocorrências policiais tem início com o registro das ocorrências. Um cidadão, quando vítima de um crime, comparece a uma delegacia para registrar um boletim de ocorrência. O cidadão, diante de um escrivão (servidor público com competência legal para tal ato), digita as informações fornecidas pelo cidadão em um sistema chamado SPP - Sistema de Procedimentos Policiais (mostrado na Figura 6.1).

Através do SPP é possível registrar uma série de procedimentos policiais, tais como Boletim de Ocorrência (BO), Termo Circunstanciado de Ocorrência (TCO), Auto de Prisão em Flagrante (APF), etc. Cada procedimento é destinado a uma situação específica.

Dentre os procedimentos policias disponíveis, escolheu-se trabalhar com os Boletins de Ocorrência, pois, como define o Manual de Polícia Judiciária [86], “é o documento utilizado pelos órgãos da Polícia Civil para o registro da notícia do crime, ou seja, aqueles fatos que devem ser apurados através do exercício da atividade de Polícia Judiciária” e “presta-se fielmente à descrição do fato, registrando horários, determinando locais, relacionando veículos e objetos, descrevendo pessoas envolvidas, identificando partes etc”.

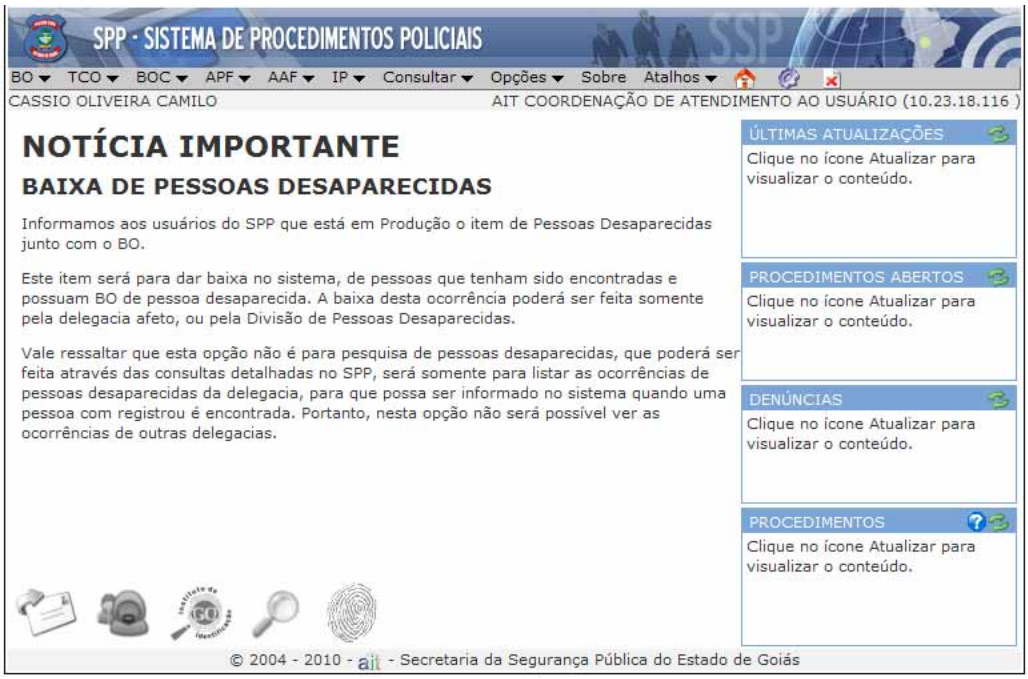


Figura 6.1: *Tela inicial do sistema SPP*

O SPP é o sistema responsável por receber os dados digitados por um escrivão e armazená-los em uma banco de dados. Os dados armazenados englobam tanto a parte estruturada quanto a não-estruturada. Na parte estruturada, são registrados os dados que possuem valores definidos, tais como sexo=masculino, estado_civil=solteiro, etc. A parte não-estruturada, chamada histórico, contém o relato do cidadão sobre o fato. A Figura 6.2 mostra a tela inicial para o cadastro de uma ocorrência.

Através do Boletim de Ocorrência é possível registrar diversos fatos (ou crimes). A classificação destes eventos recebe o nome de tipificação. Assim, no contexto deste trabalho são utilizados os crimes tipificados como “Homicídio Doloso”. O Código Penal Brasileiro [105] define no Art. 121 “Homicídio” como “Matar alguém” e no Art. 18 “Doloso” como “I - doloso, quando o agente quis o resultado ou assumiu o risco de produzí-lo”.

Um exemplo de boletim de ocorrência pode ser visto na Figura 6.3. Os atributos que identificam autor e vítima foram borrados a fim de preservar a integridade e a confidencialidade das partes. Observa-se na figura a parte estruturada (dados do fato e vítima) e a não-estruturada (histórico).

A decisão em se trabalhar com a tipificação “Homicídio Doloso” se deve à grande repercussão e comoção social que este tipo de crime gera na população. Por sua complexidade e importância é um dos principais objetos de estudo da Análise Criminal.

Após a conclusão do registro da ocorrência, a mesma já poderá ser utilizada como material de análise para o Analista Criminal.

Na Secretaria de Segurança Pública do Estado de Goiás, atualmente, o processo

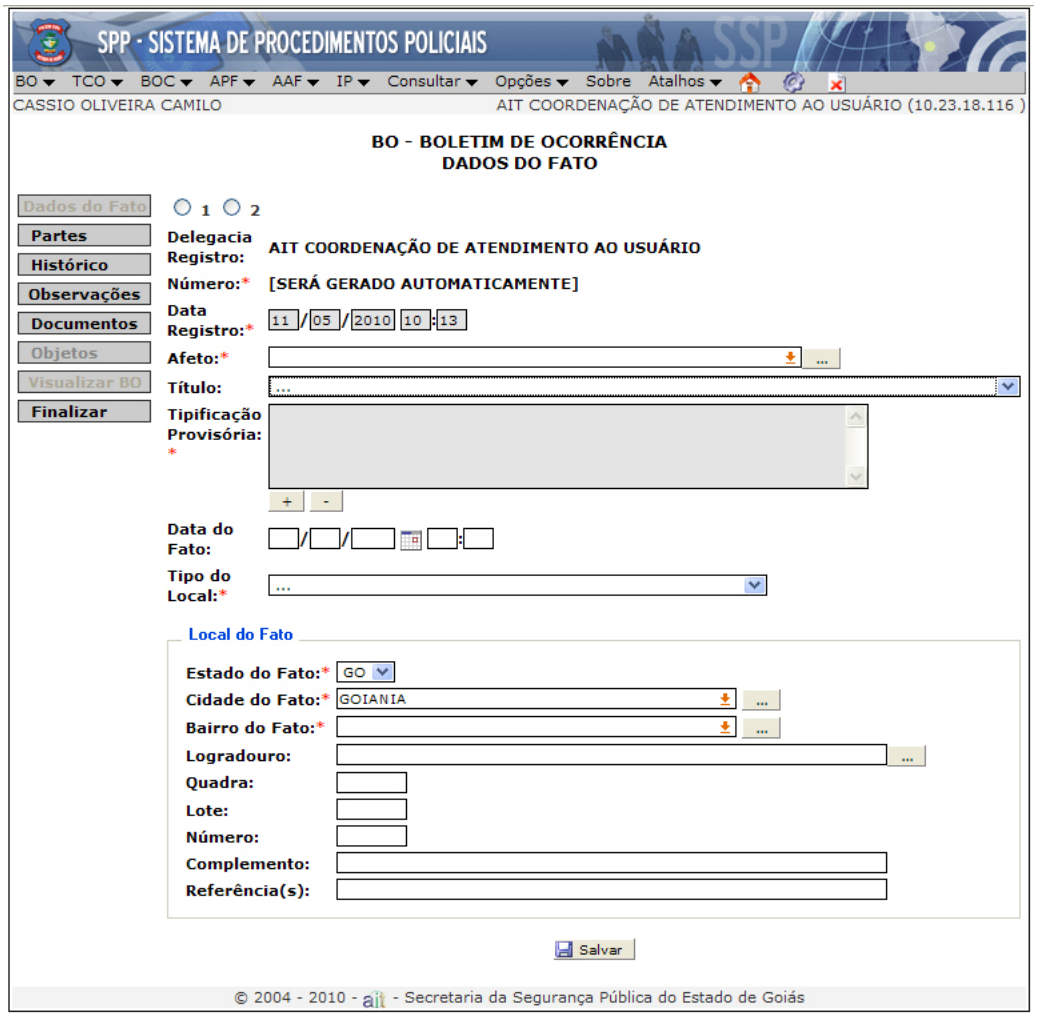
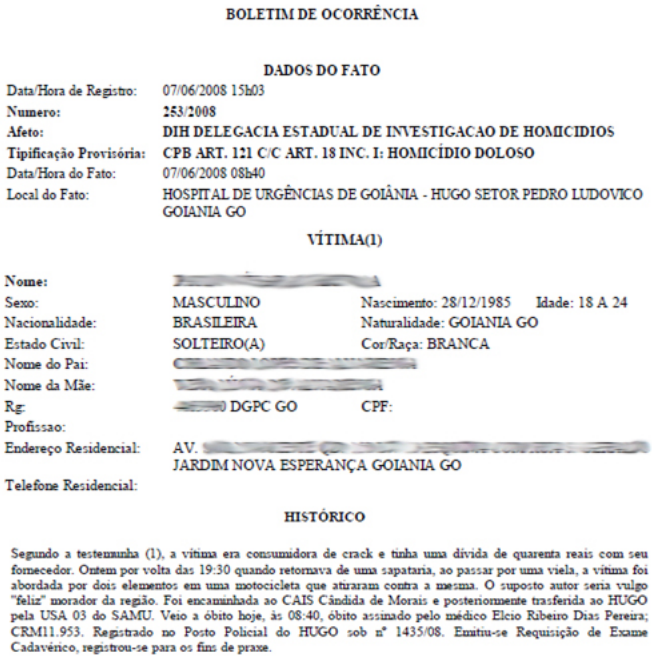


Figura 6.2: *Tela de cadastro de um boletim de ocorrência no sistema SPP*

de análise das ocorrências é feito de forma manual, sendo necessária a leitura de cada ocorrência para que se faça a tabulação dos dados. Em geral, a leitura é feita visando identificar:

- O comportamento do suposto autor quando do ocorrido: se fugiu, se prestou socorro, etc;
- O local onde ocorreu o evento;
- O meio em que ocorreu o crime, tal como se foi por arma de fogo ou branca, ou se foi um acidente, etc;
- A possível motivação para o crime: se passional, briga, fútil, drogas, etc;

O processo de análise começa com a escolha, por parte de analista criminal, do período desejado para realizar a interpretação dos dados, podendo compreender uma semana, um mês ou até um ano. Além do período, escolhe-se o tipo de crime, uma vez que a tipificação influencia o processo de análise. Após definir o período e o tipo de crime,



BOLETIM DE OCORRÊNCIA

DADOS DO FATO

Data/Hora de Registro:	07/06/2008 15h03
Numero:	253/2008
Afeto:	DIH DELEGACIA ESTADUAL DE INVESTIGACAO DE HOMICIDIOS
Tipificação Provisória:	CPB ART. 121 C/C ART. 18 INC. I: HOMICÍDIO DOLOSO
Data/Hora do Fato:	07/06/2008 08h40
Local do Fato:	HOSPITAL DE URGÊNCIAS DE GOIÂNIA - HUGO SETOR PEDRO LUDOVICO GOIANIA GO

VÍTIMA(1)

Nome:		
Sexo:	MASCULINO	Nascimento: 28/12/1985 Idade: 18 A 24
Nacionalidade:	BRASILEIRA	Naturalidade: GOIANIA GO
Estado Civil:	SOLTEIRO(A)	Cor/Raça: BRANCA
Nome do Pai:		
Nome da Mãe:		
Rg:	DGPC GO	CPF:
Profissao:		
Endereço Residencial:	AV. JARDIM NOVA ESPERANÇA GOIANIA GO	
Telefone Residencial:		

HISTÓRICO

Segundo a testemunha (1), a vítima era consumidora de crack e tinha uma dívida de quarenta reais com seu fornecedor. Ontem por volta das 19:30 quando retornava de uma sapataria, ao passar por uma viela, a vítima foi abordada por dois elementos em uma motocicleta que atiraram contra a mesma. O suposto autor seria vulgo "feliz" morador da região. Foi encaminhada ao CAIS Cândida de Morais e posteriormente trasferida ao HUGO pela USA 03 do SAMU. Veio a óbito hoje, às 08:40, óbito assinado pelo médico Elcio Ribeiro Dias Pereira; CRM11.953. Registrado no Posto Policial do HUGO sob nº 1435/08. Emitiu-se Requisição de Exame Cadavérico, registrou-se para os fins de praxe.

Figura 6.3: *Exemplo do Boletim de Ocorrência contendo a parte estruturada (dados do fato, vítima) e não-estruturada (histórico)*

os boletins de ocorrências são recuperados, através de relatórios do sistema, e analisados individualmente.

Para cada ocorrência, o analista transcreve os dados estruturados para uma planilha. Durante esta transcrição, alguns dados são transformados, como por exemplo, o campo IDADE com valor “64” será transcrito na planilha como “Idoso”. Em seguida, a parte não-estruturada, que neste contexto representa o histórico da ocorrência, é analisada. O analista lê o histórico buscando identificar os conceitos presentes. Por exemplo, para um histórico que contém termos como “tiro” e “bala”, o analista entende que naquele crime foi utilizada uma arma de fogo. Esses conceitos também são inseridos na planilha.

Ao final da análise, todos os dados estruturados e não-estruturados estarão transcritos em uma planilha, que é a base para geração de estatísticas e análises criminais. Durante a criação da planilha, caso seja necessário, o analista consulta outras fontes de dados, como por exemplo informações sócio-econômicas, e as consolida nesta planilha, também de forma manual. A Figura 6.4 ilustra o fluxo completo deste trabalho.

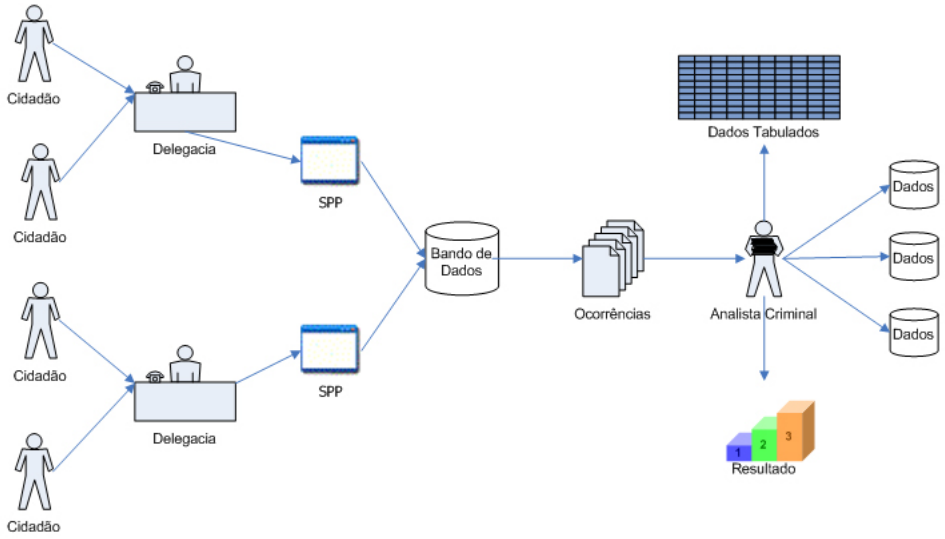


Figura 6.4: *Visão geral do processo de análise*

Para o escopo deste trabalho, serão utilizados apenas alguns dos atributos disponíveis no banco de dados. A escolha desses atributos foi baseada na experiência dos especialistas. A Tabela 6.1 mostra estes atributos e os valores de domínio. Além dos atributos disponíveis no banco de dados, foram utilizados também os conceitos possíveis de serem obtidos na parte textual, de acordo com a Ontologia. A coluna "Origem" indica se o atributo pertence ao banco de dados ou foi extraído do histórico.

Atributo	Domínio	Origem
Sexo Autor	MASCULINO, FEMININO	BD
Sexo Vítima	MASCULINO, FEMININO	BD
Raça Autor	PARDA, BRANCA, NEGRA	BD
Raça Vítima	PARDA, BRANCA, NEGRA	BD
Faixa Etária Autor	CRIANÇA, ADOLESCENTE, ADULTO, IDOSO	BD
Faixa Etária Vítima	CRIANÇA, ADOLESCENTE, ADULTO, IDOSO	BD
Estado Civil Autor	SOLTEIRO(A), UNIÃO ESTÁVEL, CASADO(A), DIVORCIADO(A), VIUVO(A), SEPARADO JUDICIALMENTE	BD
Estado Civil Vítima	SOLTEIRO(A), UNIÃO ESTÁVEL, CASADO(A), DIVORCIADO(A), VIUVO(A), SEPARADO JUDICIALMENTE	BD

Estado Instrução Autor	1º GRAU INCOMPLETO, PRIMARIO, 2º GRAU INCOMPLETO, 1º GRAU COMPLETO, 2º GRAU COMPLETO, 3º GRAU INCOMPLETO, ANALFABETO, 3º GRAU COMPLETO	BD
Estado Instrução Vítima	1º GRAU INCOMPLETO, PRIMARIO, 2º GRAU INCOMPLETO, 1º GRAU COMPLETO, 2º GRAU COMPLETO, 3º GRAU INCOMPLETO, ANALFABETO, 3º GRAU COMPLETO	BD
Período	MANHÃ, TARDE, NOITE, MADRUGADA	BD
Dia da Semana	SEGUNDA-FEIRA, TERÇA-FEIRA, QUARTA-FEIRA, QUINTA-FEIRA, SEXTA-FEIRA, SÁBADO, DOMINGO	BD
Local	ESTABELECIMENTO COMERCIAL, VIA PÚBLICA, RESIDÊNCIA, ZONA RURAL, ÓRGÃO PÚBLICO, ESTAB. SAÚDE, FORA PERÍMETRO URBANO, VEÍCULO PARTICULAR, GARAGEM/ESTACIONAMENTO, TEMPLO RELIGIOSO (IGREJA, SINAGOGA, ETC), LEITO/MARGEM DE RIO/CÓRREGO	BD
Comportamento	EVADIU-SE, SOCORRIDO	Texto
Motivação	ALCOOL, BRIGAS, FÚTIL, PASSIONAL, ROUBO, VINGANÇA	Texto
Arma de Fogo	ESPINGARDA, PISTOLA, REVÓLVER	Texto
Arma Branca	CANIVETE, FACA, PAU, PEDRA	Texto
Acidente de Trânsito	ATROPELAMENTO, BATIDA	Texto
Acidente	CAIR, DERRAME, QUEDA	Texto
Meio	ACIDENTE, ACIDENTE TRÂNSITO, ARMA BRANCA, ARMA DE FOGO	Texto
Local	BAR, CASA, ESTACIONAMENTO, HOSPITAL, LOTE, RESIDÊNCIA, RUA	Texto

Tabela 6.1: *Atributos utilizados no processo de geração das regras*

6.2 Ferramentas Utilizadas

Foram utilizadas as seguintes ferramentas para auxiliar a implementação da metodologia proposta neste trabalho.

6.2.1 WEKA

O WEKA (*Waikato Environment for Knowledge Analysis*) [108] foi desenvolvido na Universidade de Waikato e é amplamente utilizado na realização de testes com algoritmos de Mineração de Dados. É distribuído de forma gratuita e implementa uma série de algoritmos para as tarefas de mineração. A ferramenta permite que os algoritmos possam ser aplicados diretamente através de uma interface ou embutidos no código Java através de sua API. Uma referência completa em relação a isso pode ser obtida em [10]. Em [112] a ferramenta é apresentada em detalhes.

Figura 6.5: *Ferramenta WEKA*

Para a comunicação com a ferramenta, foi escolhido o padrão de arquivo ARFF (*Attribute-Relation File Format*) [79]. Este arquivo é utilizado como entrada para os algoritmos de mineração.

O padrão ARFF foi desenvolvido especificamente para a ferramenta WEKA e é composto basicamente por duas seções: Cabeçalho (onde os atributos são descritos) e Dados (que contém o conjunto de instâncias). Pode-se ver na Tabela 6.2 um exemplo simplificado de um arquivo ARFF.

Neste trabalho, a ferramenta WEKA foi utilizada para execução do algoritmo *Apriori* a fim de obter as regras de associação. Um arquivo no formato ARFF foi gerado e utilizado como entrada para o algoritmo.

6.2.2 Protégé

O Protégé é uma ferramenta *open source* para edição de Ontologias [52]. Através dela é possível construir Ontologias, seguindo o padrão OWL de forma visual, incluindo

```
@RELATION reeducando

@ATTRIBUTE idade NUMERIC
@ATTRIBUTE estado_civil STRING
@ATTRIBUTE class {fuga, não fuga}

@DATA
23,Solteiro,fuga
42,Casado,não fuga
```

Tabela 6.2: *Estrutura do arquivo ARFF*

a criação de classes, instâncias e relacionamentos. Além de ser uma *interface* gráfica, ela fornece uma API que possibilita a integração da ferramenta com outros aplicativos.

A Figura 6.6 mostra a visualização de uma Ontologia criada pela ferramenta.

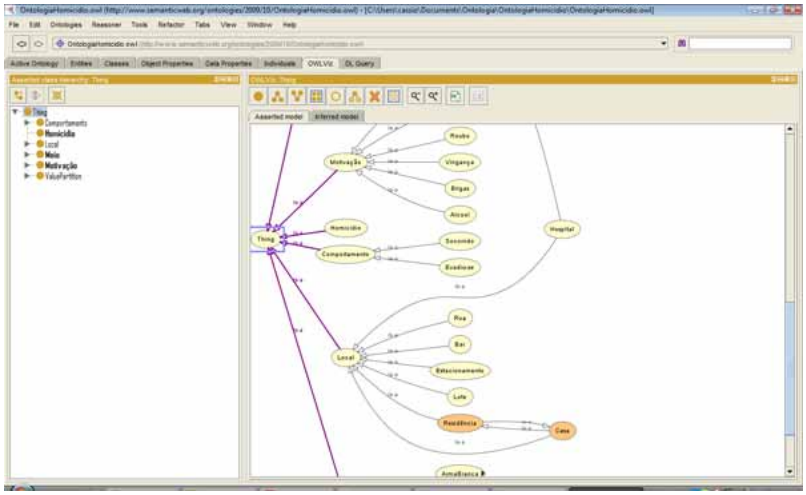


Figura 6.6: *Ferramenta Protégé mostrando a hierarquia de uma Ontologia*

Neste trabalho, o Protégé foi utilizado por um especialista para edição da Ontologia necessária. Foram mapeados os conceitos, relacionamentos e termos. Ao final, um arquivo OWL foi gerado, contendo a descrição da Ontologia.

6.2.3 *Framework* Jena

O Jena é um *framework open source* escrito em Java, para construção de aplicações baseadas em Web Semântica desenvolvida pelo *HP Labs Semantic Web Programme*, que fornece um ambiente de programação para RDF, RDFS, OWL, SPARQL e inclui um motor de inferências baseado em regras [49]. Possui os seguintes componentes:

- API RDF;
- Suporte para leitura e escrita RDF
  XML, N3 e N-Triplas;

- API OWL;
- Armazenamento em memória e persistência;
- Motor de consulta SPARQL.

O *framework* Jena permite a realização de operações, integradas a outros aplicativos, incluindo a realização de inferências sobre arquivos RDF e OWL. Através da inferência é possível interagir com a Ontologia e extrair informações que estejam indiretamente mapeadas.

A API do Jena permite interagir com o arquivo OWL criado pela ferramenta Protégé e realizar operações para recuperar os conceitos e seus termos.

6.2.4 Ferramenta de ETL *Kettle*

O *Kettle* [51] é uma ferramenta da *suite Pentaho* [80] utilizada para realizar as operações de extração, transformação e carga de dados. Através da ferramenta é possível desenhar um fluxograma que contém os passos que serão executados, como, por exemplo, ler os dados de diversas fontes, consolidá-los, tratá-los e armazená-los em um novo repositório. Além disso, a ferramenta possui um agendador que permite que o fluxo seja executado em horário e data programados.

A Figura 6.7 apresenta a tela inicial da ferramenta. Neste trabalho a ferramenta foi utilizada para implementar a etapa "Preparação do Ambiente".

Figura 6.7: *Ferramenta Kettle para Integração de Dados*

6.3 Funcionamento do Sistema

Para implementação da metodologia proposta, foram necessárias abordagens distintas para cada uma das suas etapas: na etapa "Preparação do Ambiente" adotou-se uma

solução de agendamento, e na etapa “Processamento” desenvolveu-se uma ferramenta específica para filtragem das regras de associação produzidas.

Esta separação foi necessária uma vez que a etapa “Preparação do Ambiente” contém atividades que demandam um tempo de processamento considerável, não sendo viável executá-la sempre que se deseje fazer a análise dos dados. Já a etapa “Processamento” visa investigar os dados em busca de padrões, o que, em geral, é feito de forma interativa e iterativa, exigindo refinamentos como, por exemplo, a mudança no valor de uma medida objetiva para eliminar regras irrelevantes.

Um vez definido o fluxo desejado para a preparação do ambiente, e desde que a Ontologia tenha sido criada, todo o processamento ocorre de forma automatizada: na etapa “Preparação do Ambiente”, as atividades necessárias são executadas de forma programada, e na etapa “Processamento”, após definidos os valores das medidas de interesse objetivas e subjetivas desejados para filtragem das regras, não há necessidade de intervenção do usuário.

6.3.1 Etapa: Preparação do Ambiente

Para implementar a etapa "Preparação do Ambiente"utilizou-se a ferramenta de ETL *Kettle*. Além de integrar as bases de dados, foram realizadas tarefas para limpeza, transformação e redução dos dados. A Figura 6.8 apresenta o fluxo criado na ferramenta e utilizado neste trabalho. A execução do fluxo foi agendada para todos os dias durante a madrugada.

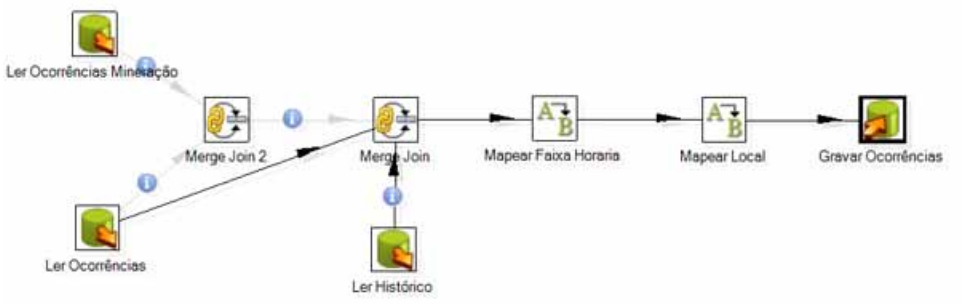


Figura 6.8: *Fluxo criado utilizando a ferramenta Kettle*

O fluxo começa com a combinação das tarefas “Ler Ocorrências Mineração” e “Ler Ocorrências”. Na primeira, as ocorrências são lidas de um *data warehouse* da instituição utilizando a linguagem SQL. Os registros contendo valores nulos e duplicados são eliminados pela consulta. Na segunda tarefa, alguns atributos não disponíveis no *data warehouse*, como por exemplo os dados relativos às partes envolvidas (autor e vítima), são recuperados da base de dados transacional, também utilizando a linguagem SQL. Seguindo o fluxo, aos atributos obtidos nas tarefas anteriores é adicionado o atributo não-estruturado que contém o histórico da ocorrência, também recuperado da base de dados

transacional através da linguagem SQL. Em seguida, os valores numéricos do atributo que representa a hora de ocorrência do evento foram convertidos (se for referente a valores) para um valor qualitativo, representando a faixa horária. O mesmo foi feito com o campo que representa o local do ocorrido. Por fim, os registros foram gravados em um repositório central.

6.3.2 Etapa: Processamento

Para a etapa "Processamento", a solução adotada foi o desenvolvimento de uma ferramenta específica. A ferramenta proposta foi desenvolvida utilizando a linguagem Java [48], e as APIs (*Application Programming Interface*) do Jena [49], para integração com a Ontologia, e do WEKA [108], para execução do algoritmo de mineração de regras de associação. A Figura 6.9 apresenta a tela inicial da ferramenta MARFUSS (*Mining Association Rules From Unstructured and Structured Sources*).

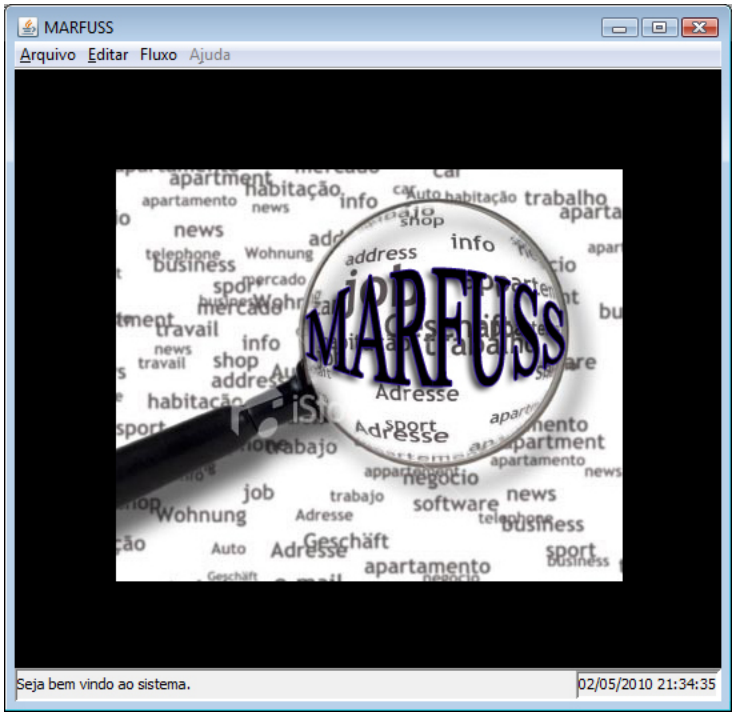


Figura 6.9: *Tela inicial do sistema proposto*

Para possibilitar a extração dos conceitos presentes na parte textual das ocorrências, conforme propõe a metodologia, foi criada uma Ontologia para mapear o conhecimento dos especialistas sobre um domínio, neste caso, o crime "Homicídio Doloso". Foi utilizado o auxílio de especialistas da área criminal para criação da Ontologia e a ferramenta Protégé [52] para editá-la. A Figura 6.10 mostra uma visão geral da Ontologia criada. O arquivo OWL completo da Ontologia pode ser visto no Apêndice A.

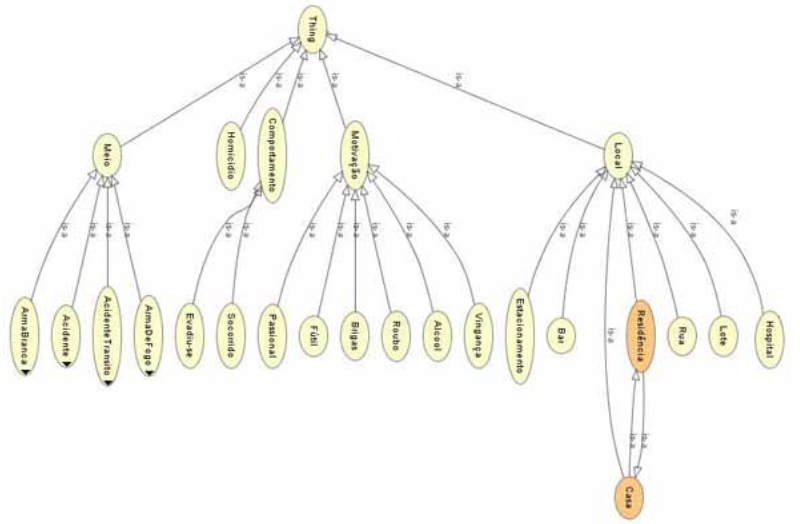


Figura 6.10: *Visão Geral da Ontologia "Homicídio Doloso" criada na ferramenta Protégé*

Nota-se que a criação da Ontologia foi feita de forma manual, com o auxílio de especialistas que mapearam o conhecimento sobre o domínio específico, através dos conceitos, na Ontologia. O processo de identificação e definição dos conceitos também ocorreu de forma manual. O fluxo se deu da seguinte forma: inicialmente, os conceitos foram identificados e representados; em seguida, os termos relativos a cada conceito foram identificados e o número mínimo de termos necessários definido; depois, estes conceitos e seus termos foram mapeados na ferramenta Protégé para criar a Ontologia; e, por fim, os relacionamentos entre os conceitos foram definidos.

A Figura 6.11 ilustra um conceito representado na Ontologia através da ferramenta Protégé. Observa-se o conceito "ArmaDeFogo", que pode ser de três tipos, "Espingarda", "Pistola" e "Revólver", e possui a propriedade "compostoPor" e "minimoDe". Em destaque, temos a classe "ArmaDeFogoValuePartition" que contém os termos do conceito que o representa. A propriedade "minimoDe" possui o valor "2" indicando que são necessários no mínimo dois termos para que o conceito seja considerado presente. A linguagem OWL utiliza a classe "ValuePartition" para representar termos.

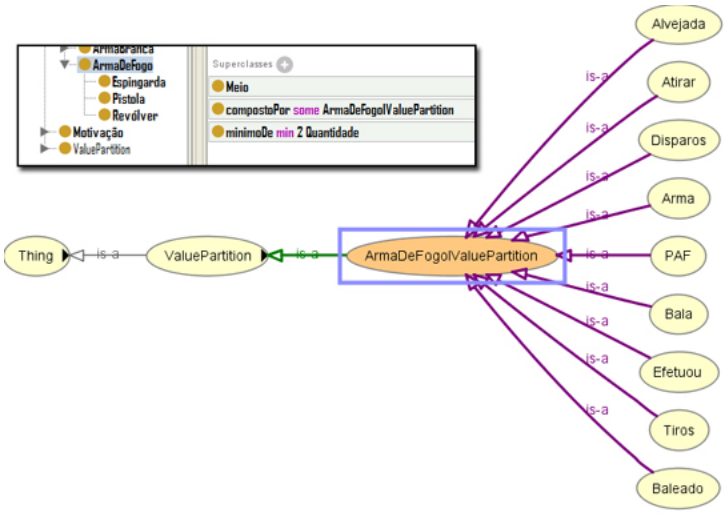


Figura 6.11: *Conceito "ArmadeFogo" e seus termos*

A Ontologia possui cerca de 120 classes, incluindo os descritores dos conceitos e seus termos. Para possibilitar a extração automática dos conceitos mapeados na Ontologia, foram criadas três propriedades específicas:

compostoPor Esta propriedade indica o relacionamento do descritor do conceito com os termos. A Figura 6.12 mostra o conceito "Faca" e seus termos. Esta propriedade é utilizada para montar o vetor de termos do conceito necessário no cálculo da similaridade.

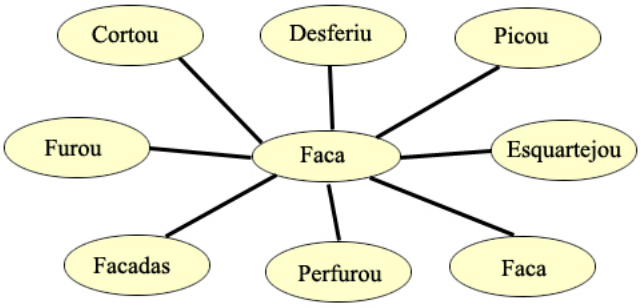


Figura 6.12: *Exemplo do conceito "Faca" e seus termos*

temUm Esta propriedade é utilizada para indicar as características que devem ser identificadas nos textos. Na Figura 6.13 tem-se que o conceito "Homcídio" possui as características de "Comportamento", "Local", "Meio" e "Motivação".

minimoDe Esta propriedade visa identificar dentro de um conjunto de propriedades a quantidade mínima de relacionamentos que devem existir. Por exemplo, para o conceito "Faca" temos a propriedade indicando que deve haver no mínimo dois termos para o conceito ser considerado presente.

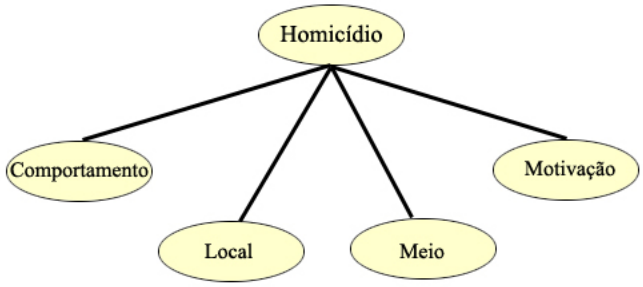


Figura 6.13: *Exemplo do relacionamento temUm do conceito "Homicídio"*

Através destas propriedades, foi possível interagir com a Ontologia criada e automatizar o processo de interpretação dos dados não-estruturados, para então combiná-los com os demais dados estruturados obtidos do banco de dados. O resultado desta combinação é um arquivo do tipo ARFF, utilizado como entrada pela ferramenta WEKA. Um trecho do arquivo utilizado pode ser visto abaixo:

```
1
2  @relation ocorrencias
3
4  @attribute sexo_autor {MASCULINO,FEMININO}
5  @attribute sexo_vitima {MASCULINO,FEMININO}
6  @attribute cor_raca_autor {PARDA,BRANCA,NEGRA}
7  @attribute cor_raca_vitima {PARDA,BRANCA,NEGRA}
8  @attribute faixa_etaria_autor {Adulto,Idoso,Adolescente,Criança}
9  @attribute faixa_etaria_vitima {Adulto,Idoso,Adolescente,Criança}
10 @attribute estado_civil_autor {SOLTEIRO(A),'UNIÃO␣ESTÁVEL',CASADO(A),
       DIVORCIADO(A),VIUVO(A),'SEPARADO(A)␣JUDICIALMENTE'}
11 @attribute estado_civil_vitima {SOLTEIRO(A),'UNIÃO␣ESTÁVEL',CASADO(A),
       DIVORCIADO(A),VIUVO(A),'SEPARADO(A)␣JUDICIALMENTE'}
12 @attribute grau_instrucao_autor {'1º␣GRAU␣INCOMPLETO',PRIMARIO,'2º␣GRAU
       ␣INCOMPLETO','1º␣GRAU␣COMPLETO','2º␣GRAU␣COMPLETO','3º␣GRAU␣
       INCOMPLETO',ANALFABETO,'3º␣GRAU␣COMPLETO'}
13 @attribute grau_instrucao_vitima {'1º␣GRAU␣INCOMPLETO',PRIMARIO,'2º␣
       GRAU␣INCOMPLETO','1º␣GRAU␣COMPLETO','2º␣GRAU␣COMPLETO','3º␣GRAU␣
       INCOMPLETO',ANALFABETO,'3º␣GRAU␣COMPLETO'}
14 @attribute periodo {Manhã,Tarde,Noite,Madrugada}
15 @attribute dia_da_semana {Quinta-feira,Domingo,Terça-feira,Sexta-feira,
       Sábado,Quarta-feira,Segunda-feira}
16 @attribute local {'ESTABELECIMENTO␣COMERCIAL','VIA␣PÚBLICA',RESIDÊNCIA,
       'ZONA␣RURAL','ÓRGÃO␣PÚBLICO','ESTAB.␣SAÚDE','FORA␣PERÍMETRO␣URBANO'
       ,'VEÍCULO␣PARTICULAR',GARAGEM/ESTACIONAMENTO,'TEMPLO␣RELIGIOSO␣(
       IGREJA,␣SINAGOGA,␣ETC)','LEITO/MARGEM␣DE␣RIO/CÓRREGO'}
17 @attribute Comportamento {Socorrido,Evadiu-se}
```

```
@attribute Motivação {Vingança,Roubo,Passional,Fútil,Brigas,Alcool}
@attribute ArmaDeFogo {Revólver,Pistola,Espingarda}
@attribute ArmaBranca {Pedra,Pau,Faca,Canivete}
@attribute AcidenteTransito {Batida,Atropelamento}
@attribute Acidente {Cair,Queda,Derrame}
@attribute Meio {ArmaDeFogo,ArmaBranca,AcidenteTransito,Acidente}
@attribute Local {Rua,Casa,Residência,Lote,Hospital,Estacionamento,Bar}

@data
MASCULINO,MASCULINO,PARDA,BRANCA,Adulto,Adulto,SOLTEIRO(A),SOLTEIRO(A),
    ANALFABETO,'3º GRAU COMPLETO',Manhã,Quinta-feira,'ESTABELECIMENTO
    COMERCIAL',Socorrido,Brigas,?,?,?,?,ArmaDeFogo,Hospital
MASCULINO,FEMININO,?,BRANCA,Adulto,Adulto,SOLTEIRO(A),SOLTEIRO(A),
    ANALFABETO,'3º GRAU COMPLETO',Manhã,Quinta-feira,'VIA PÚBLICA',?,
    Alcool,?,Faca,?,?,ArmaDeFogo,Bar
MASCULINO,FEMININO,PARDA,?,Adulto,Adulto,SOLTEIRO(A),SOLTEIRO(A),
    ANALFABETO,'3º GRAU COMPLETO',Manhã,Domingo,RESIDÊNCIA,Evadiu-se,
    Passional,?,?,?,?,ArmaDeFogo,Residência
```

A Figura 6.14 apresenta o diagrama de classes da ferramenta proposta. O sistema é basicamente composto por uma classe que desenha a tela principal, onde são configuradas as medidas de interesse, uma classe responsável por executar a extração dos conceitos e a filtragem das regras, uma classe responsável pela integração com a API do WEKA e outra para integração com a API do Jena.

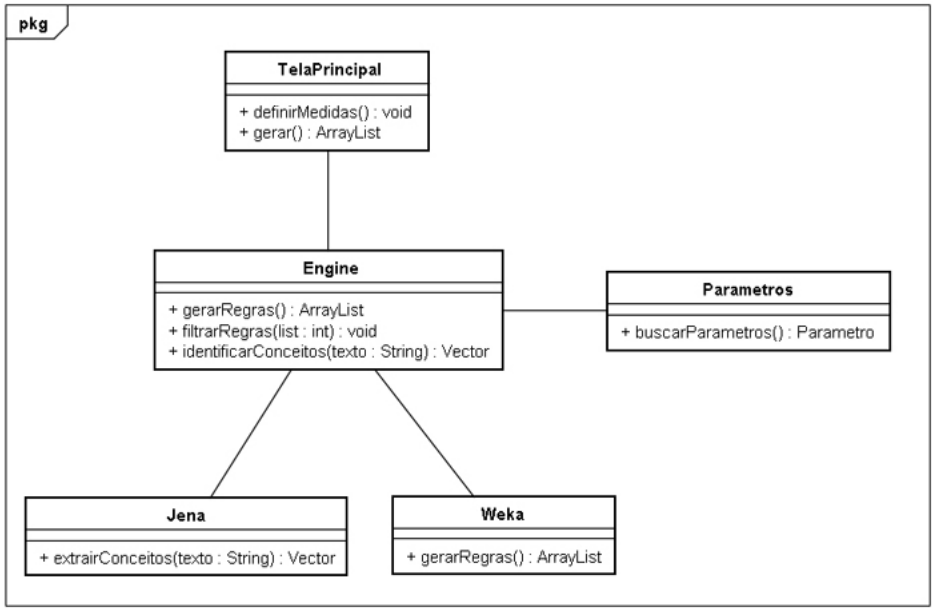


Figura 6.14: *Diagrama de Classes do sistema proposto*

O diagrama de sequência para a geração das regras de associação é mostrado na Figura 6.15. Observa-se que é necessário que o usuário defina os valores para as

medidas de interesse objetivas e subjetivas. Após definir os valores, a geração das regras é solicitada. O sistema recupera os parâmetros necessários, tais como o local onde o arquivo OWL está armazenado e a *string* de conexão com o banco de dados. Em seguida, os conceitos presentes na parte não-estruturada dos dados são extraídos. Estes valores são agrupados com os demais dados estruturados, através do arquivo ARFF, e as regras de associação são então geradas e filtradas.

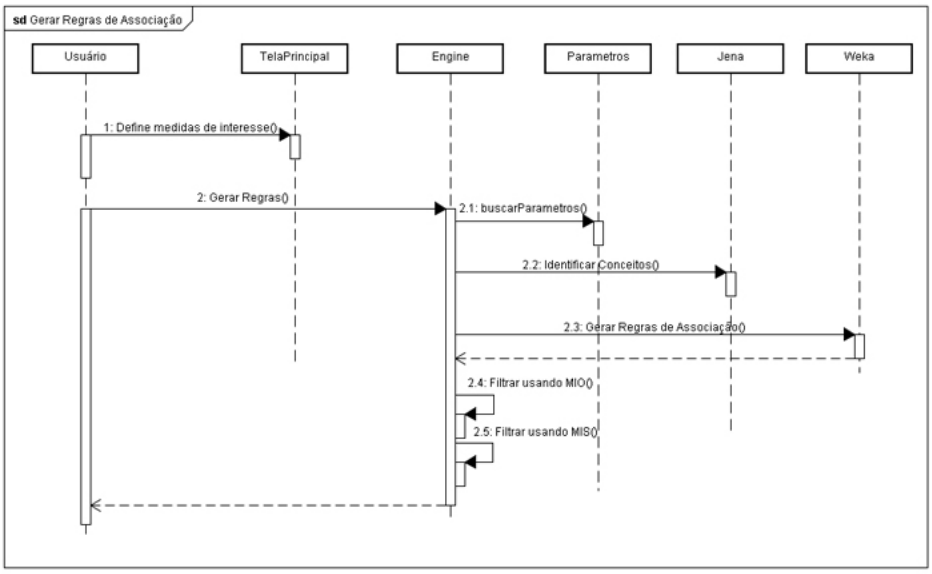


Figura 6.15: *Diagrama de sequência para funcionalidade "Gerar Regras"*

É durante a etapa "Processamento" que os dois pontos chave da metodologia são executados. A seguir, cada um destes pontos são detalhados.

6.3.2.1 Extração de conceitos

Durante a extração de conceitos, é feita a leitura dos conceitos mapeados na Ontologia e eles são confrontados com o texto, a fim de identificar os que estão presentes. Basicamente são executados os seguintes passos:

1. Para cada histórico, retirar *stopwords* e realizar o *stemming*;
2. Criar o vetor de termos para o histórico;
3. Ler a ontologia, buscando os conceitos chaves através da propriedade "composto-Por";
4. Para cada conceito chave, recuperar os termos através da propriedade "temUm";
5. Tratar o conceitos, retirando as *stopwords* e realizando o *stemming*;
6. Criar o vetor de termos para os conceitos;
7. Calcular a similaridade entre o vetor de histórico e os vetores de conceitos;

8. Calcular o valor mínimo para que o conceito possa ser considerado presente, através da propriedade “minimoDe”;
9. Recuperar todos os conceitos que possuem valores de similaridades maiores que o mínimo estabelecido;
10. Criar o arquivo ARFF.

Para o processo de remoção das *stopwords* foi utilizado um arquivo que contém uma lista de palavras consideradas irrelevantes. Caso o termo em questão esteja presente nesta lista, o mesmo é eliminado. Para a aplicação do processo de *stemming* foi utilizada a implementação *opensource* do algoritmo de Porter para o idioma Português, denominada PTStemmer [85]. No Apêndice B são mostrados os códigos utilizados para implementar os passos descritos.

Para exemplificar o comportamento da ferramenta, toma-se o seguinte histórico extraído de uma ocorrência policial (a fim de garantir o sigilo das partes, alguns elementos do texto foram retirados, entretanto, o conteúdo do texto foi mantido conforme cadastrado):

“Às 12:05 de hoje fomos comunicados da presente ocorrência. Nos deslocamos ao local, onde encontramos o corpo da vítima(1) caído em frente ao endereço já mencionado. Também na qd.x, em frente à qd.x, lt.x, n°x, na mesma rua, estava a motocicleta HONDA/CG 150 TITAN KS; cor vermelha; placa x, de propriedade de S., caída na calçada, com um buraco de tiro atravessando o tanque. As vítimas vinham do velório de A.M.B., vítima de homicídio, quando foram surpreendidas pelo(s) atirador(es). No local foram encontrados estojos de munição cal.40 e 380, além de projéteis que foram colhidos pela perita de local. Emitiu-se Requisição de Exame Cadavérico registrou-se para os devidos fins.”

Após o processo de remoção das *stopwords* e a realização do *stemming*, tem-se o seguinte texto:

“fom comunic pres ocorr desloc encontr corp vitim caid frent enderec mencion frent rua motociclet hondacg titan cor vermelh plac nlm propriedad caid calc burac tir atravess tanqu vitim vinh velori vitim homicidi conform registr delegac surpreend atir encontr estoj mun cal projetel colh perit iracild emitius requis exam cadaver registrous”

Um vetor é gerado para este texto, em que cada elemento representa uma palavra. O mesmo procedimento ocorre com os conceitos extraídos da Ontologia. Com os vetores gerados, a similaridade é calculada. Para o texto acima, os conceitos foram obtidos, com o valor da similaridade, e estão apresentados na Tabela 6.3.

Conceito	Valor	Similaridade
Meio	ArmaDeFogo	0.2
Local	Rua	1.0

Tabela 6.3: *Conceitos e Similaridades obtidas com o processamento do histórico da ocorrência*

6.3.2.2 Filtragem das regras

O processo de filtragem das regras compreende a eliminação das regras fracas utilizando inicialmente as medidas de interesse objetivas e posteriormente as medidas de interesse subjetivas. O processo ocorre em três etapas: geração das regras, filtragem utilizando as medidas de interesse objetivas e, por fim, filtragem utilizando as medidas de interesse subjetivas.

Basicamente são realizados os seguintes passos:

1. Geração das regras de associação utilizando o algoritmo Apriori, eliminando as regras com suporte e confiança menores que os estabelecidos pelo usuário;
2. Cálculo das medidas *lift*, novidade e convicção;
3. Eliminação das regras com valores de *lift*, novidade e convicção menores, ou maiores, que os estabelecidos pelo usuário;
4. Cálculo das medidas subjetivas Conformidade, Antecedente Inesperado, Consequente Inesperado e Antecedente e Consequente Inesperados;
5. Eliminação das regras com valores das medidas subjetivas menores que os estabelecidos pelo usuário.

Na tela mostrada na Figura 6.16 são definidas as medidas de interesse objetivas. A figura também informa os valores de suporte e confiança, além dos valores mínimos e máximos para as medidas lift, leverage e convicção. As regras geradas que possuírem valores fora da faixa definida serão eliminadas.

Na tela mostrada na Figura 6.17 são definidos os parâmetros das medidas de interesse subjetivas. Para que as medidas de interesse subjetivas possam ser calculadas é necessário que o usuário crie regras de associação que representem o conhecimento prévio sobre o assunto em questão, permitindo o confronto entre as regras geradas e as definidas.

Para definir as regras que deseja, o usuário deve selecionar o tipo de conhecimento (Impressão Geral ou Conhecimento Impreciso), o tipo de regra (Conformidade, Antecedente Inesperado, Consequente Inesperado ou Antecedente e Consequente Inesperado), cadastrar as regras e informar os valores mínimos de cada medida subjetiva. O formato da regra segue o padrão $X => Y$. Através do botão “Ver campos” os atributos possíveis e seus valores de domínio são mostrados para que o usuário possa utilizá-los.

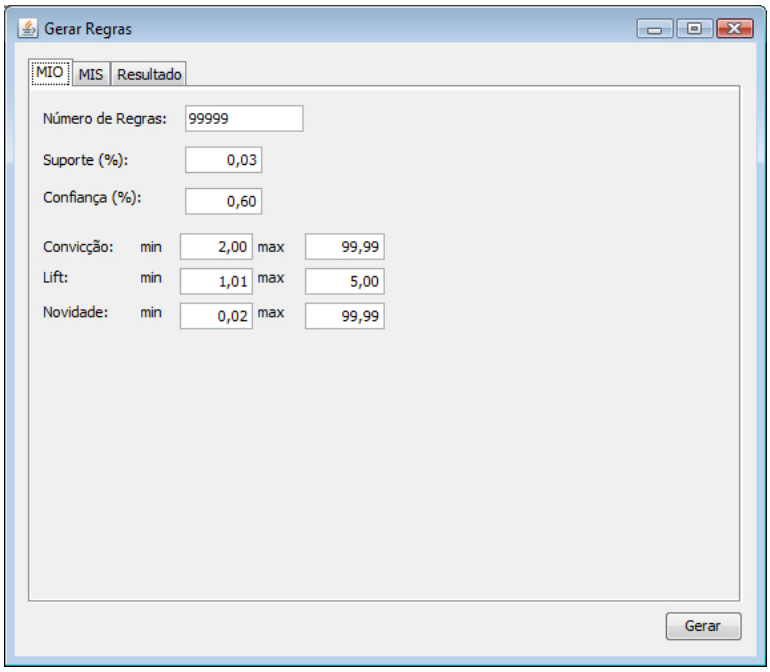


Figura 6.16: *Tela utilizada para definição das medidas objetivas*

No campo "Regras" o usuário monta de forma manual as regras que deseja. Ao clicar no botão "Adicionar" a regra é validada e em caso positivo inserida.

Conforme propõe a metodologia, para cada regra gerada são calculados os valores das medidas objetivas e subjetivas. Esses valores são confrontados com os valores mínimos definidos pelo usuário e as regras com valores menores são descartadas. A Figura 6.18 apresenta o resultado da execução. No Apêndice B são mostrados os códigos utilizados para realizar a geração e filtragem das regras.

A fim de exemplificar o processo de filtragem, tome-se como exemplo a seguinte regra gerada:

```
sexo_autor=MASCULINO estado_civil_vitima=SOLTEIRO(A) ==> Meio=ArmaDeFogo
```

Considerando os valores encontrados e os definidos pelo usuário para as medidas de interesse objetivas apresentadas na Tabela 6.4, tem-se que a regra está dentro dos valores definidos e, portanto, não será descartada.

Medidas	Valor Calculado	Valor Mínimo	Valor Máximo
Suporte	0.5	0.3	-
Confiança	0.8	0.6	-
Lift	2.67	1.01	5.00
Novidade	0.4	0.2	0.9
Convicção	4.0	2.0	10.0

Tabela 6.4: *Valores das medidas de interesse objetivas*

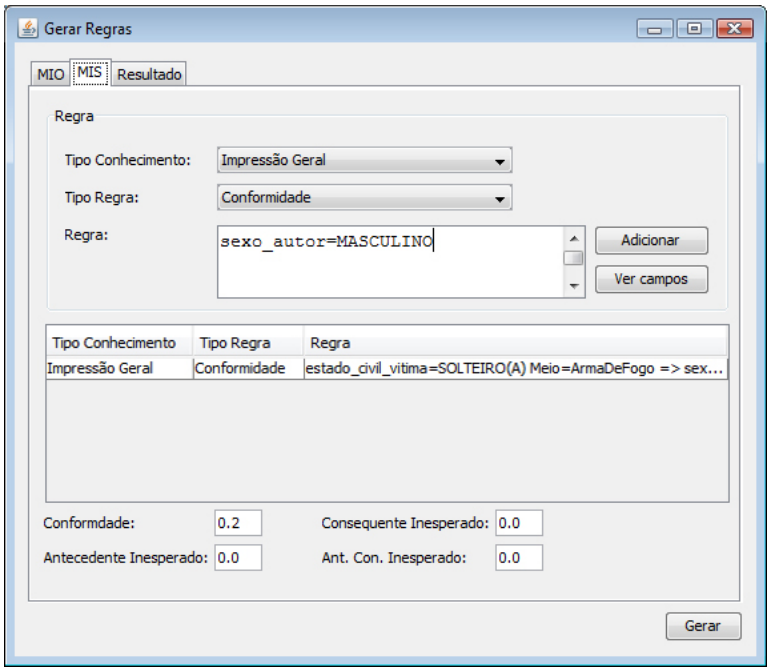


Figura 6.17: *Tela utilizada para definição das medidas subjetivas*

Em seguida, é necessário analisar as medidas de interesse subjetivas. Considerando que o usuário definiu a seguinte regra para a medida de interesse subjetiva, do tipo "Impressão Geral" e "Conformidade":

```
sexo_autor=MASCULINO ==> Meio=ArmaDeFogo
```

Conforme descrito na Seção 4.4.2, nesta situação temos a aplicação da Fórmula $conf_{ij} = A_{ij} \cdot C_{ij}$. Como $\frac{contAnt}{nAnt} < \frac{contCon}{nCon}$, onde $contAnt = 1$, $nAnt = 2$, $contCon = 1$ e $nCon = 1$, temos que $C_{ij} = \min(\frac{contCon}{nCon}, \frac{nContItens}{nItens})$ e $A_{ij} = \frac{contAnt}{nAnt}$. Para $nContItens = 2$ e $nItens = 2$, temos que $C_{ij} = 1$ e $A_{ij} = 0.5$, portanto $conf_{ij} = 0.5$.

Este valor deve ser confrontado com o valor mínimo informado pelo usuário e, caso esteja abaixo, a regra é descartada. O processo se repete para as demais regras definidas pelo usuário. Caso exista mais de uma regra do mesmo tipo definida pelo usuário, por exemplo conformidade, os valores resultantes serão somados.

Comparando os valores mínimos para as medidas subjetivas informados na Tabela 6.5 e o valor calculado para a regra acima, percebe-se que esta regra será mantida.

Medidas	Valor Calculado	Valor Mínimo
Conformidade	0.5	0.2

Tabela 6.5: *Valores das medidas de interesse subjetivas*

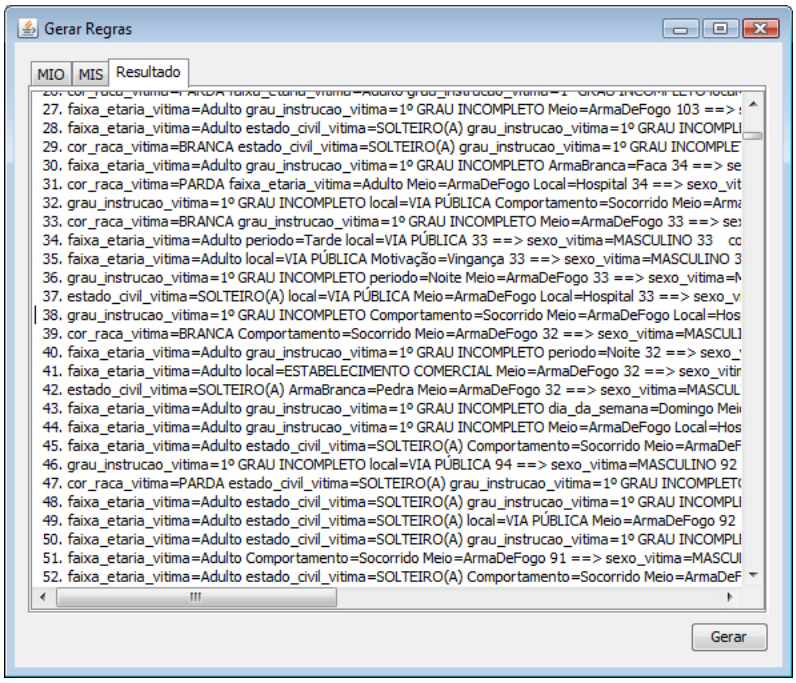


Figura 6.18: *Regras produzidas*

CAPÍTULO 7

Resultados

Neste capítulo são apresentados os resultados obtidos com a execução da metodologia proposta, através da aplicação da ferramenta criada sobre os dados da instituição utilizada no estudo de caso. Na Seção 7.1, é apresentado o contexto para validação dos resultados. A Seção 7.2 mostra os resultados obtidos durante a etapa "Preparação do Ambiente" e a Seção 7.3 apresenta os resultados obtidos na etapa "Processamento", destacando a extração dos conceitos e a filtragem das regras.

7.1 Contextualização

Devido à falta de um processo automatizado na instituição escolhida para o estudo de caso, a verificação dos resultados obtidos se deu da seguinte maneira:

- Para a etapa "Preparação do Ambiente" foi verificada a quantidade de registros inicialmente lidos e comparada com a quantidade final do repositório gerado.
- Para a etapa "Processamento" os dois pontos chave foram analisados de forma que os resultados obtidos com a extração de conceitos relevantes nos textos foram confrontados com os resultados da análise manual feita pelos especialistas, e para validação das regras produzidas foi utilizada a opinião dos especialistas, uma vez que, devido ao grande volume de dados, foi inviável a geração manual das regras de associação.

Para execução da metodologia proposta, foram selecionados 1.020 eventos criminais ocorridos entre 1 de janeiro de 2009 e 31 de dezembro de 2009. A parte estruturada da aplicação contém 13 atributos que armazenam informações relativas ao fato (data, hora e local) e as pessoas envolvidas (sexo, idade, raça, etc). A parte não-estruturada possui apenas um atributo que contém a descrição do evento. Porém, com a extração dos conceitos relevantes presentes na Ontologia, ao total são extraídos 8 conceitos. Assim, ao total foram utilizados 21 campos na execução do algoritmo de mineração de regras de associação.

7.2 Etapa: Preparação do Ambiente

Nesta etapa, foram acessadas as seguintes bases de dados da Secretaria de Segurança Pública:

- *Data warehouse*: visando buscar os dados das ocorrências policiais, para crimes de homicídio doloso, do ano de 2009;
- Base de dados transacional: visando buscar os dados das partes envolvidas nas ocorrências, bem como o histórico do evento.

A base de dados transacional, ao todo, possui na tabela de partes envolvidas cerca de 1.250.000 registros e na tabela de histórico cerca de 800.000 registros. O *data warehouse* possui cerca de 600.000 registros. O processamento completo do fluxo levou em média 30 minutos.

Ao todo, existem na base de dados, para o ano de 2009 e tipo de crime "homicídio doloso", cerca de 1.150 registros. Porém, devido ao processo de limpeza e eliminação de registros duplicados, ao final foram selecionados 1.020 registros.

Conforme análise manual das consultas executadas, verificou-se que deveriam ser eliminados 130 registros. O mesmo resultado foi obtido com a execução da ferramenta proposta. Constatou-se, assim, uma taxa de acerto de 100% para esta etapa. Este valor para a taxa de acerto já era esperado, uma vez que não existem fatores subjetivos nesta etapa.

7.3 Etapa: Processamento

Apesar de não existir a necessidade da intervenção humana durante a execução desta etapa, a não ser pela definição dos valores mínimos das medidas de interesse, para a validação dos resultados obtidos foi necessário analisar de forma isolada dois pontos: extração dos conceitos e filtragem das regras. No primeiro verificou-se a viabilidade da utilização da Ontologia na extração dos conceitos presentes nos textos e no segundo verificou-se a performance na realização da filtragem das regras.

Essa divisão foi necessária, pois, somente com as regras de associação produzidas, não seria possível validar se os conceitos extraídos dos textos foram, de fato, os corretos.

7.3.1 Extração de conceitos nos textos

No processo de validação dos conceitos relevantes extraídos dos textos, o resultado obtido com a execução da metodologia foi comparado com o resultado da análise manual feita pelos especialistas. Para a análise manual foi necessária a leitura de todos

os eventos pelos especialistas. A fim de facilitar a comparação dos resultados, os eventos foram divididos em três grupos de acordo com o tamanho do texto: textos com menos de 50 palavras (grupo 1), textos contendo entre 50 e 150 palavras (grupo 2) e textos com mais de 150 palavras (grupo 3).

A divisão proposta visa agrupar textos que possuam o mesmo nível de detalhes. Assim, os textos do grupo 1 possuem uma descrição fraca do evento. No grupo 2 os textos possuem a descrição básica. Os textos do grupo 3 apresentam detalhes sobre os acontecimentos.

A seguir, descreve-se o comportamento da metodologia, no tocante à extração de conceitos, e a comparação com os resultados obtidos pelos especialistas. Para exemplificar foi escolhido um texto de cada grupo.

No texto do grupo 1, a metodologia identificou apenas se o conteúdo digitado de fato referenciava ao evento ocorrido. O mesmo resultado foi obtido pelo especialista. A similaridade calculada entre o vetor de termos do conceito e o vetor de termos do texto foi de 22%. A seguir um exemplo de texto do grupo 1 (o texto foi mantido em sua forma original):

"O Grupo de Investigação de Homicídios, na data de 27/02/2008, por volta das 12:30h, foi acionado para comparecer na Reserva Ambiental situada no setor Madre Germana I, onde fora encontrado um cadáver enterrado, em avançado estado de decomposição (ossadas). Não foi possível a identificação da vitima, nem mesmo seu sexo ou idade. A Policia Militar, Técnico-Científica e IML também estiveram presentes no local onde foram tomadas as providencia cabíveis. No local, foram encontrados três estojos, devidamente apreendidos pelo Perito para posterior exame pericial."

Para o texto do grupo 2, a metodologia identificou os conceitos mostrados na Tabela 7.1. Os conceitos identificados pelos especialistas são mostrados na Tabela 7.2. É possível verificar que o conceito "rua" encontrado pela metodologia não está presente na análise manual. Isto ocorreu pela forma de representação do conceito na Ontologia. A seguir é apresentado um exemplo para o texto do grupo 2 (o texto foi mantido em sua forma original):

"De acordo com a testemunha André Tavares de Lima, a vítima chegou na porta do Bar TROPICAL CHOP, vindo caminhando do rumo da praça, quando dois indivíduos não identificado em uma Moto Honda 150cc, cor preta, parou na rua e o da garupa desceu sem o capacete e de arma em punho, sem nada dizer, começou a disparar por várias vezes no rumo da vítima, sendo que esta foi alvejada e caiu na calçada, em seguida os mesmo empreenderam fuga não dando tempo para ver as suas características. Foi chamado o socorro do "SAMU", onde compareceu a viatura USA n° 03 e o Dr. Laurence constatou o óbito. A vítima era foragido do sistema prisional, onde cumpria pena no Regime Semi-aberto e tinha envolvimento com traficantes de Drogas. Compareceu no local do fato a

viatura n° 3555 da 29ª CIPM, comandada pelo Sgt. Rogério e também compareceu a Perita Edmaria. Expediu-se a guia de exame cadavérico e registrou-se."

Conceito	Valor	Similaridade
Homicídio	Homicídio	0.222
Meio	Arma de Fogo	0.3
Local	Rua	1
Local	Bar	1
Motivação	Vingança	0.25

Tabela 7.1: *Grupo 2: conceitos identificados pela metodologia*

Característica	Valor
Foi homicídio?	Sim
Como matou?	Arma de Fogo
Onde?	Bar
Porque?	Vingança

Tabela 7.2: *Grupo 2: conceitos identificados pela análise do especialista*

Para o texto do grupo 3, a metodologia identificou os conceitos mostrados na Tabela 7.3. O mesmo resultado foi obtido pela análise manual do especialista (Tabela 7.4). Um exemplo do texto do grupo 3 pode ser visto abaixo (o texto foi mantido em sua forma original):

"O comunicante informa que hoje, quinta-feira, 01 de janeiro por volta das 18h15min foi informado via telefone pela Polícia Militar sobre a ocorrência de um homicídio doloso no Município do Girassol; Que acompanhado dos agentes Gílson e Durães foi ao local do fato e constatou a veracidade da informação, tratava-se de um homicídio doloso tendo como vítima a pessoa conhecida por "F. DE TAL", morador da cidade de Cocalzinho; Que em diligências próximo ao local do fato foi informado por populares de que "F." estava bebendo no "Bar da Galega"; Que a equipe foi ao Bar da Galega, mas este estava fechado, com correntes e cadeados pela parte externa; Que foi verificado manchas de sangue em frente ao Bar, mas alguém havia pego uma enxada, raspado e colocado terra para escondê-lo; Que em conversa com uma moradora, obteve a informação de que no Bar estavam duas mulheres e uns rapazes, iniciando em seguida uma discussão entre eles; Que uma mulher morena foi vista segurando uma faca, logo após "F. de Tal" ter sido esfaqueado e um outro rapaz esfaqueado sair correndo; Que a equipe encontrou o outro rapaz, trata-se de A.C.V.S., residente, no bairro Itamar Nobre; Que A.C. disse que estava no Bar da Galega e que um indivíduo conhecido por Salomão teria um caso com a Galega dona do Bar; Que ele teria matado o F. e depois corrido em sua direção para esfaqueá-lo; Que A. foi conduzido a esta delegacia para prestar declarações, mas como estava em estado de embriaguez não foi possível realizar tal oitiva; Que a

equipe teve informações por populares de que F. morava em Cocalzinho e já possuía várias passagens pela polícia, inclusive cumprido pena; Que foi acionado o IML para remoção do corpo de F.; Que após correr do Bar da Galega caiu morto em frente a Rua, a morte foi provocada por uma facada na região abdominal, que perfurou algum órgão. Nada mais."

Conceito	Valor	Similaridade
Homicídio	Homicídio	0.444
Arma Branca	Faca	0.25
Local	Bar	1
Motivação	Brigas	0.111
Motivação	Álcool	0.4
Comportamento	Evadiu-se	0.2

Tabela 7.3: *Grupo 3: conceitos identificados pela metodologia*

Característica	Valor
Foi Homicídio	Sim
Como matou?	Arma Branca
Tipo de arma?	Faca
Onde?	Bar
Porque?	Álcool e Brigas
Atitude autor?	Evadiu-se

Tabela 7.4: *Grupo 3: conceitos identificados pela análise do especialista*

Por fim, após a análise completa dos 1.020 eventos, obteve-se o gráfico da Figura 7.1. É possível observar que, para os textos do grupo 1, a taxa de acerto da metodologia, comparado com o resultado da análise manual, é de 98,5% (349 acertos e 5 erros). Para os textos do grupo 2, a taxa de acerto é de 94% (402 acertos e 23 erros) e para os textos do grupo 3 a taxa de acerto é de 95% (230 acertos e 11 erros).

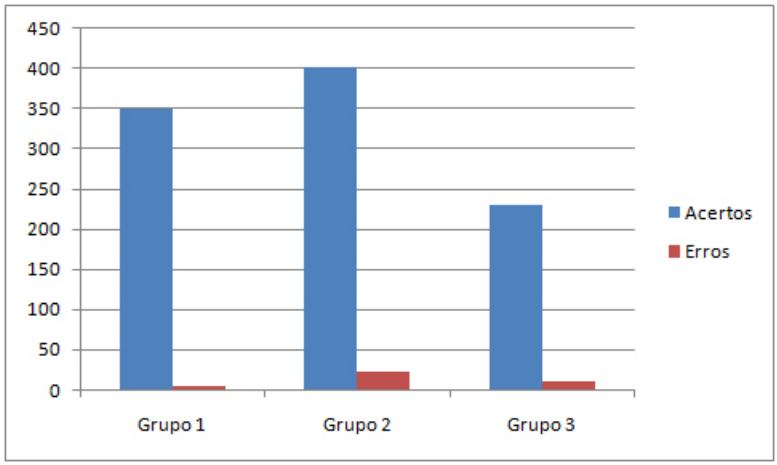


Figura 7.1: *Resultado geral da extração de informações*

Devido aos altos valores obtidos para as taxas de acerto no processo de extração de conceitos, foi realizada uma análise detalhada a fim de identificar possíveis falhas. Inicialmente, através de uma análise dos dados pode-se perceber que:

- A escolha de um determinado tipo de ocorrência, previamente selecionada para um domínio específico, auxiliou na eliminação de possíveis erros, uma vez que apenas textos relacionados ao conhecimento contido na Ontologia foram analisados;
- A criação manual da Ontologia, feita por especialistas criminais para um domínio específico e com conhecimento sobre as ocorrências selecionadas, permitiu que os conceitos fossem mais precisos.

Em seguida, optou-se por aplicar o mesmo procedimento de extração de conceitos, utilizando inclusive a mesma Ontologia, sobre um novo conjunto de ocorrências. Porém, desta vez, foram escolhidos outros tipos de ocorrências. Assim, esperava-se obter um resultado inverso, ou seja, os conceitos presentes na Ontologia não deveriam ser encontrados.

Em virtude da falta de uma análise já existente para outros tipos de ocorrências e de recursos humanos disponíveis, foi necessário selecionar um conjunto restrito de ocorrências, uma vez que o processo de validação foi feito de forma manual. Assim, foram selecionadas de forma aleatória, dentro do período do ano de 2009, outras 745 ocorrências, sendo divididas em dois grandes grupos: o grupo 1 composto por ocorrências relativas a homicídio (exceto os dolosos), com um total de 356, e o grupo 2 composto por ocorrências relativas a outros crimes, com um total de 389.

Para cada um dos grupos, fez-se a subdivisão em três subgrupos, sendo que o subgrupo 1 contém os textos com menos de 50 palavras, o subgrupo 2 contém os textos contendo entre 50 e 150 palavras e o subgrupo 3 contém os textos com mais de 150 palavras. A Tabela 7.5 contém o quantitativo das ocorrências para cada subgrupo.

Quantidade de ocorrências		
Grupo	Subgrupo	Quantidade
Relativo Homicídio	Subgrupo 1	116
	Subgrupo 2	203
	Subgrupo 3	63
Outros Crimes	Subgrupo 1	171
	Subgrupo 2	145
	Subgrupo 3	73

Tabela 7.5: *Quantidade de ocorrências*

Após a execução da função de extração de conceitos para cada uma das ocorrências e a comparação com a análise do especialista, obteve-se para as ocorrências do grupo 1, relativas a homicídio, uma taxa de acerto de 77% (Subgrupo 1: 87%; Subgrupo

2: 75%; Subgrupo 3: 70%). Verificou-se que os conceitos descritos na Ontologia não foram tão precisos para possibilitar a diferenciação entre o homicídio culposo (aquele em que não há intenção de matar) e a tentativa de homicídio (quando a vítima não vem a óbito). Obsevou-se que este resultado se deve não pela execução da metodologia em si, mas pelo nível de detalhamento da Ontologia. Quanto mais precisos forem o mapeamento e a descrição da Ontologia, melhor será a identificação dos conceitos. Para as ocorrências do grupo 2, relativas a outros crimes, foi obtida uma taxa de 85% (Subgrupo 1: 83%; Subgrupo 2: 85%; Subgrupo 3: 87%). Verificou-se que apesar de terem sido encontrados alguns dos conceitos descritos na Ontologia, como local do evento, outros como o meio utilizado e a motivação não foram identificados. A Figura 7.2 ilustra os resultados obtidos.

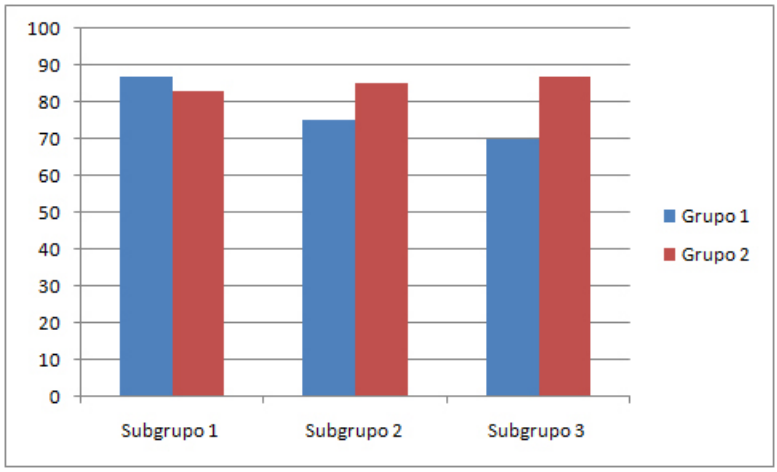


Figura 7.2: *Resultado da segunda validação*

7.3.2 Filtragem das regras

Após a análise dos resultados da extração de conceitos nos textos, há a verificação das regras geradas. Para filtragem das regras foram usadas as equações 4-1, 4-2 e 4-3 no cálculo das medidas de interesse objetivas e as equações 4-4, 4-5, 4-6 e 4-7 no cálculo das medidas de interesse subjetivas. Os valores mínimos aceitáveis por cada medida foram definidos pelos especialistas. Para definição destes valores foram realizadas diversas tentativas até que se chegasse a um conjunto final de regras julgado satisfatório.

Para validar a metodologia proposta, no tocante à filtragem das regras de associação utilizando uma combinação de medidas de interesse objetiva e subjetiva, foi analisada inicialmente a utilização das medidas objetivas para depois combiná-las com as medidas subjetivas. A Tabela 7.6 mostra algumas medidas subjetivas definidas pelo usuário.

Os valores das medidas de interesse objetivas foram definidos pelos especialistas. Foi possível observar que quanto menores foram os valores das medidas de interesse mais regras eram geradas, e com isso, surgiram muitas regras inúteis. Observou-se também

Tipo de Medida	Antecedente	Consequente
Conformidade	faixa_etaria_vitima=Adulto	Meio=ArmaDeFogo
Conformidade	cor_raca_vitima=PARDA faixa_etaria_vitima=Adulto estado_civil_vitima=SOLTEIRO(A) Meio=ArmaDeFogo	sexo_vitima=MASCULINO
Conformidade	grau_instrucao_vitima=1º GRAU INCOMPLETO	sexo_vitima=MASCULINO faixa_etaria_vitima=Adulto
Ant. Inesp.	sexo_autor=FEMININO	Meio=ArmaDeFogo
Ant. Inesp.	faixa_etaria_autor=Crianca Meio=ArmaDeFogo	faixa_etaria_vitima=Adulto
Ant. Inesp.	estado_civil_vitima=CASADO(A) grau_instrucao_autor=SUPERIOR COMPLETO sexo_autor=FEMININO	sexo_vitima=FEMININO
Cons. Inesperado	sexo_autor=MASCULINO Meio=ArmaDeFogo estado_civil_autor=CASADO(A)	sexo_vitima=FEMININO estado_civil_vitima= CASADO(A)
Cons. Inesp.	Meio=ArmaBranca grau_instrucao_autor=SUPERIOR COMPLETO	cor_raca_vitima=BRANCA
Cons. Inesp.	Meio=Acidente Trânsito	ArmaBranca=PEDRA
Ant. Cons. Inesp.	ArmaBranca=PAU	Local=RESIDÊNCIA
Ant. Cons. Inesp.	grau_instrucao_autor=SUPERIOR COMPLETO	grau_instrucao_vitima= SUPERIOR COMPLETO
Ant. Cons. Inesp.	Motivacao=VINGANÇA	ArmaFogo=ESPINGARDA

Tabela 7.6: *Configuração das medidas de interesse subjetivas*

que com valores maiores o número de regras diminuía, porém, regras muito óbvias eram geradas. Após diversas combinações de valores das medidas de interesse feitas pelos especialistas de maneira empírica, chegou-se a uma configuração ideal mostrada da Tabela 7.7. Esta combinação foi feita de forma manual, definindo os valores, executando a ferramenta, redefinindo os valores e assim sucessivamente. Após definição dos valores pelos especialistas, foi obtido um total de 275 regras.

Sobre o conjunto de 275 regras filtradas através das medidas objetivas, uma nova filtragem foi realizada, porém, utilizando-se das medidas subjetivas. Após serem calculados os valores das medidas subjetivas para cada regra e depois da eliminação daquelas com valor menor do que o definido, obteve-se um total de 17 regras. Conforme será visto a seguir, a taxa de regras consideradas úteis utilizando-se a combinação das medidas objetivas e subjetivas para filtragem ficou acima de 80%.

Com o intuito de verificar a efetividade das medidas de interesse subjetivas combinada com as medidas objetivas, foram realizados diversos testes ajustando apenas os valores das medidas de interesse objetivas visando obter as mesmas regras produzidas

Suporte	Confiança	*Lift*	Convicção	Novidade
0,05	0,80	1,05	2,00	0,03

Tabela 7.7: *Valores das medidas de interesse objetivas*

com a combinação das medidas. Porém, não se obteve o mesmo resultado, uma vez que utilizando apenas as medidas objetivas não foi possível eliminar as regras que não representavam conhecimentos novos e úteis ao domínio em questão. Foi possível constatar que a utilização das medidas subjetivas agrega ao processo de filtragem das regras o conhecimento do especialista, possibilitando que regras com baixa objetividade sejam consideradas úteis, bem como regras com alta objetividade sejam descartadas.

O gráfico da Figura 7.3 representa o comparativo entre o total de regras filtradas utilizando apenas as medidas de interesse objetivas e o total de regras consideradas úteis pelos especialistas. Na primeira coluna, do total de 1620 regras, 243 foram consideradas úteis, representando um percentual de 15%. Na segunda coluna, tivemos 1400 regras geradas e destas, 182 foram úteis (13%). Na terceira coluna, de um total de 700 regras, 210 foram consideradas úteis (30%). Na quarta coluna, das 250 regras, 80 foram consideradas úteis (32%). Na quinta coluna, com 100 regras, tivemos um percentual de 32%, com 32 regras úteis. Na sexta coluna, foi obtida uma taxa de 34% (50 regras geradas e 17 úteis). Na última coluna, das 17 regras geradas, 7 foram consideradas úteis, respresentando 41%.

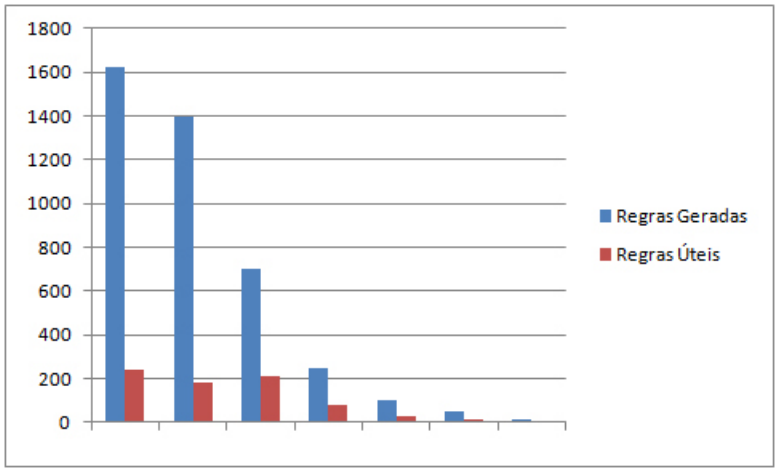


Figura 7.3: *Comparativo entre as regras geradas e as realmente úteis utilizando apenas as medidas de interesse objetivas*

No gráfico da Figura 7.4, os valores apresentados são da combinação entre as medidas objetivas e subjetivas. Observa-se que a combinação das medidas permite a geração de um número menor de regras, porém com mais expressividade. Na primeira coluna, foram geradas 235 regras e destas, 198 foram consideradas úteis, um taxa de 84%. Na segunda coluna, das 100 regras geradas, 87 foram consideradas úteis (87%). Na terceira coluna, foram geradas 50 regras, sendo 41 úteis, com taxa de 82%. Na última

coluna, das 17 regras geradas, 15 foram consideradas úteis, o que resultou em uma taxa de 88%.

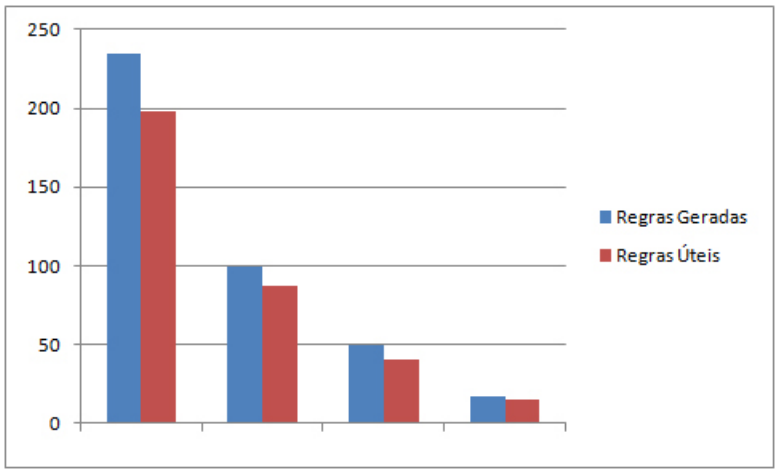


Figura 7.4: *Comparativo entre as regras geradas e as realmente úteis utilizando a combinação de medidas de interesse objetivas e subjetivas*

Após uma análise dos resultados obtidos verificou-se que os valores para as taxas de acertos se devem à utilização do conhecimento do especialista, através das medidas subjetivas, para filtragem das regras.

A Tabela 7.8 apresenta o conjunto final das regras obtidas. Com base nessas regras, o analista criminal pode sugerir aos gestores públicos ações que visem diminuir os índices criminais, ou até mesmo obter novas linhas de investigação. Por exemplo, a regra "*estado_civil_vitima=SOLTEIRO(A) Meio=ArmaDeFogo ==> sexo_vitima=MASCULINO*" indica que os crimes envolvendo arma de fogo e vítimas solteiras em geral são do sexo masculino. Essa regra indica uma linha de pesquisa para verificar por que os homens solteiros estão sendo vítimas de arma de fogo. Com base nessa resposta é possível criar campanhas preventivas para tais situações.

O conjunto final de regras produzidas foi utilizado pelos especialistas para auxiliar no processo de análise criminal do período utilizado (o ano de 2009) para o tipo de crime em questão. Segundo comentários, as regras foram consideradas úteis por terem fornecido informações que possibilitaram identificar a linha de análise criminal a ser seguida.

Regra	Antecedente	Consequente
1	estado_civil_vitima=SOLTEIRO(A) Meio=ArmaDeFogo	sexo_vitima=MASCULINO
2	sexo_autor=MASCULINO local=RESIDÊNCIA	sexo_vitima=FEMININO
3	sexo_autor=MASCULINO	faixa_etaria_vitima=Adulto sexo_vitima=MASCULINO
4	estado_civil_vitima=SOLTEIRO(A)	faixa_etaria_vitima=Adulto
5	cor_raca_vitima=PARDA	sexo_vitima=MASCULINO faixa_etaria_vitima=Adulto
6	sexo_autor=FEMININO Meio=ArmaDeFogo	faixa_etaria_vitima=Adulto sexo_autor=MASCULINO
7	Meio=AcidenteTrânsito sexo_autor=MASCULINO grau_instrucao_autor=SUPERIOR COMPLETO	sexo_vitima=MASCULINO
8	faixa_etaria_vitima=Adulto	sexo_vitima=MASCULINO Meio=ArmaDeFogo
9	sexo_autor=MASCULINO estado_civil_autor=CASADO(A)	sexo_vitima=FEMININO estado_civil_vitima=CASADO(A)
10	cor_raca_vitima=PARDA	sexo_vitima=MASCULINO
11	sexo_vitima=MASCULINO estado_civil_vitima=SOLTEIRO(A)	faixa_etaria_vitima=Adulto Motivacao=ALCOOL
12	faixa_etaria_vitima=Adulto local=VIA PÚBLICA	sexo_vitima=MASCULINO Meio=ArmaDeFogo
13	cor_raca_autor=PARDA Meio=ArmaDeFogo	sexo_vitima=MASCULINO cor_raca_vitima=BRANCA
14	grau_instrucao_autor=1º GRAU INCOMPLETO	sexo_vitima=MASCULINO faixa_etaria_vitima=Adulto
15	periodo=MADRUGADA Meio=ArmaDeFogo	sexo_vitima=FEMININO
16	faixa_etaria_vitima=Adulto estado_civil_vitima=SOLTEIRO(A)	sexo_vitima=MASCULINO
17	cor_raca_vitima=PARDA faixa_etaria_vitima=Adulto	sexo_vitima=MASCULINO

Tabela 7.8: *Conjunto de Regras Geradas*

CAPÍTULO 8

Conclusões

O processo de análise de dados, disponíveis em diferentes formatos (estruturado e não-estruturado), tem exigido por parte dos especialistas um esforço considerável. Aliado a esta dificuldade, existe o grande volume de informações disponíveis atualmente para serem tratadas.

O tratamento dos dados não-estruturados por parte dos especialistas em geral depende da análise manual de cada documento visando identificar os conceitos presentes ou as informações relevantes.

Outro fator que dificulta a análise dos dados é a necessidade de combinação entre os dados que estão armazenadas em meios estruturados e aqueles descobertos com a análise manual dos dados não-estruturados.

Por fim, a descoberta de relacionamentos entre os dados torna-se um processo impraticável se feito de forma manual, devido ao grande volume de dados e à grande quantidade de variáveis envolvidas.

Assim, são cada vez mais necessários métodos e ferramentas que automatizem os processos de análises e interpretações de grandes volumes de dados.

Neste trabalho, é proposta uma metodologia para identificar relacionamentos entre os dados através da mineração de regras de associação. Além disto, a proposta utiliza uma Ontologia para viabilizar a integração entre os dados estruturados e não-estruturados. Uma ferramenta, para automatizar todo o processo, foi desenvolvida e aplicada em uma situação real.

Com o uso da Ontologia, pôde-se verificar a viabilidade na utilização da técnica para análise de dados não-estruturados. Optou-se por trabalhar com a metodologia de "conceitos" para mapear o conhecimento do usuário sobre um determinado domínio. Assim, foi possível extrair dos dados não-estruturados os conceitos presentes de forma automatizada com uma taxa de acerto satisfatória.

A extração dos conceitos dos dados não-estruturados permitiu representá-los em uma forma estruturada e, então, combiná-los com os demais dados permitindo que os relacionamentos implícitos fossem descobertos. Como resultado da execução

da ferramenta foram obtidas regras de associação a partir de dados estruturados e não-estruturados.

Através das regras de associação foi possível transformar os dados em informações, permitindo que ações práticas pudessem ser tomadas. Entretanto, devido ao grande volume de regras geradas, foi necessário implementar meios de filtrar aquelas realmente úteis.

As medidas de interesse foram utilizadas para filtragem das regras. A combinação entre as medidas objetivas e subjetivas mostrou-se extremamente eficaz nesta tarefa, apresentando resultados superiores à aplicação de apenas uma delas.

Cabe salientar que, apesar de ter sido utilizada apenas em um contexto, a metodologia proposta pode ser aplicada em outras situações para integrar informações armazenadas em diferentes formatos.

No contexto deste trabalho, apesar dos bons resultados obtidos, alguns pontos precisam ser observados. A qualidade da análise automática dos dados não-estruturados está relacionada diretamente com a qualidade da Ontologia criada. Além disto, a filtragem das regras depende de uma configuração de valores mínimos para as medidas de interesse objetivas e subjetivas.

Do ponto de vista científico, apesar deste trabalho não propor um novo algoritmo de mineração, ele combina diversas técnicas de forma a permitir a interação do usuário no processo de geração das regras de associação, utilizando inclusive o conhecimento dele para isso. Dentre essas técnicas, destacam-se: a mineração de dados e textos, utilizada para o tratamento dos dados e a geração das regras de associação; ontologias, utilizadas para mapear o conhecimento do usuário e possibilitar a integração entre os dados não-estruturados com os estruturados; e conceitos, utilizados como forma de representar o conhecimento na Ontologia e possibilitar a identificação de dados relevantes nos textos.

Além das técnicas mencionadas, foi utilizado o coeficiente *Overlap* para cálculo da similaridade, o algoritmo *Apriori* para geração das regras de associação, as medidas de interesse objetivas novidade, conviccção e interesse, e as medidas de interesse subjetivas conformidade, antecedente inesperado, consequente inesperado e antecedente/consequente inesperado para filtragem das regras.

A seguir, na Seção 8.1 são apresentadas as principais contribuições deste trabalho. Na Seção 8.2 são apresentados os artigo e relatórios técnicos produzidos durante o desenvolvimento desta dissertação. Por fim, na Seção 8.3 são apresentados os direcionamentos futuros.

8.1 Contribuições

As principais contribuições identificadas durante o desenvolvimento desta dissertação são apresentadas a seguir.

Combinação de tecnologias Apesar de não propor um novo algoritmo para mineração de regras de associação, ou uma nova medida de interesse, este trabalho propõe uma nova maneira de combinar diversas tecnologias existentes (mineração de dados, textos e ontologias) para mineração de regras de associação.

Criação de uma metodologia A definição de uma metodologia permitiu a geração de regras de associação, utilizando dados de fontes estruturadas e não-estruturadas, através de uma Ontologia para integração destes dados.

Desenvolvimento da ferramenta A ferramenta criada permitiu aos usuários, de forma direta, gerar as regras de associação e interagir com estas regras eliminando aquelas irrelevantes.

Interpretação semântica de textos Através das Ontologias foi possível automatizar a identificação dos conceitos presentes nos dados textuais. Além disso, utilizou-se uma estrutura semântica para isso, através do mapeamento do conhecimento do usuário.

Filtragem das regras de associação A utilização da combinação das medidas de interesse objetivas e subjetivas permitiu a filtragem das regras de forma que o conhecimento do usuário pudesse ser levado em consideração.

8.2 Produções Bibliográficas

A seguir, são apresentadas as produções realizadas durante o desenvolvimento desta dissertação.

8.2.1 Artigos Publicados

8.2.1.1 Recuperação Contextualizada de Documentos em Bibliotecas Digitais Integradas [74]

Este trabalho aborda a recuperação contextualizada de documentos em um conjunto de Bibliotecas Digitais integradas por meio do protocolo OAI-PMH. Neste caso, apenas os documentos que fazem parte de um domínio especificado devem ser recuperados. No processo de contextualização das consultas, são utilizadas ontologias e a análise do conteúdo dos artigos da Wikipédia (através da manipulação de sua API), a qual é utilizada como um repositório de conhecimento auxiliar. In: 8th International

Information and Telecommunication Technologies Symposium, Florianópolis, Brazil. I2TS 2009, December 09-11, 2009.

8.2.2 Relatórios Técnicos

8.2.2.1 Mineração de Dados: Conceitos, Tarefas, Métodos e Ferramentas

Neste trabalho são apresentados os conceitos fundamentais da Mineração de Dados, principais tarefas e métodos. Além dos métodos tradicionais, algumas variantes e novas abordagens são discutidas. Ao final é apresentada uma lista das principais ferramentas para se trabalhar com mineração.

8.2.2.2 Um estudo sobre a interação entre Mineração de Dados e Ontologias

Este trabalho apresenta os conceitos relativos ao uso de ontologias na mineração de dados, além de diversos estudos sobre como resolver certos problemas ligados a essas tecnologias, tais como melhorar as regras de associação mineradas, definir melhor a medida de similaridade entre agrupamentos, inserir o conhecimento adquirido nas fases da mineração, automatizar a escolha dos melhores algoritmos, entre outros. Apresenta-se como a interação entre a mineração e as Ontologias têm sido realizadas recentemente.

8.3 Trabalhos Futuros

A partir do levantamento bibliográfico e da ferramenta desenvolvida nesta dissertação, podem ser destacados os seguintes pontos que poderiam ser objetos de trabalhos futuros:

- Aplicação da metodologia e da ferramenta proposta em outros tipos de crimes no contexto do estudo de caso proposto;
- Utilização da Ontologia para geração de novas medidas de interesse subjetivas, de forma que seja possível extrair automaticamente da Ontologia algumas medidas de interesse subjetivas;
- Criação da Ontologia através do processamento dos dados textuais dos documentos analisados sem a necessidade de intervenção humana para mapear o conhecimento dos especialistas;
- Desenvolvimento de um Sistema de Apoio à Decisão para executar a análise das regras geradas, permitindo a interpretação automática das regras produzidas e minimizando o esforço do usuário;

- Permitir a recuperação de regras de associação baseando-se em consultas formuladas em linguagem natural, facilitando a obtenção de resultados de forma mais simples e objetiva;
- Executar a metodologia e a ferramenta proposta utilizando outras bases de dados a fim de validá-las contra outros cenários;
- Utilização de outros algoritmos de geração de regras de classificação.

Referências Bibliográficas

[1] AGRAWAL, R.; IMIELINSKI, T.; SWAMI, A. **Mining Association Rules Between Sets of Items in Large Databases**. *Proc. of the ACM SIGMOD*, p. 207–216, 1993.

[2] AGRAWAL, R.; SRIKANT, R. **Fast Algorithms for Mining Association Rules**. *20th International Conference on Very Large Data Bases*, p. 487–499, 1994.

[3] ALDENDERFER, M. S.; BLASHFIELD, R. K. **Cluster Analysis**. Beverly Hills, 1984.

[4] ALLEMANG, D.; HENDLER, J. **Semantic Web for the Working Ontologist – Effective Modeling in RDFS and OWL**. Morgan Kaufmann, 2008.

[5] ALMEIDA, M. B.; BAX, M. P. **Uma Visão Geral Sobre Ontologias: Pesquisa Sobre Definições, Tipos, Aplicações, Métodos de Avaliação e de Construção**. *Revista Ciência da Informação*, 32(3):7–20, set./dez. 2003.

[6] BARTH, F. J.; BELDERRAIN, M. C.; QUADROS, N. L. P.; FERREIRA, L. L.; TIMOSZCZUK, A. P. **Recuperação e Mineração de Informações para Área Criminal**. *VI Encontro Nacional de Inteligência Artificial - ENIA - XXVII SBC*, 2007.

[7] BARTH, F. J.; TIMOSZCZUK, A. P. **Expansão Automática de Consultas utilizando Ontologias**. *Seminário de Pesquisa em Ontologia no Brasil*, 2008.

[8] BERNARAS, A.; LARESGOITI, I.; CORERA, J. **Building and Reusing Ontologies for Electrical Network Applications**. *Proceedings of the European Conference on Artificial Intelligence, ECAI/96*, 1996.

[9] BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. **The Semantic Web**. *Scientific American Magazine*, 2001.

[10] BOUCKAERT, R. R.; FRANK, E.; HALL, M.; KIRKBY, R.; REUTEMANN, P.; SEEWALD, A.; SCUSE, D. **WEKA Manual for Version 3-6-1**. Disponível em http://www.cs.waikato.ac.nz/~ml/weka/, acessado em Maio de 2010, 2009.

[11] BRAMER, M. **Undergraduate Topics in Computer Science - Principles of Data Mining**. Springer, 2007.

[12] BRIN, S.; MOTWANI, R.; ULLMAN, J. D.; TSUR, S. **Dynamic Itemset Counting and Implication Rules for Market Basket Data**. In: *SIGMOD '97: Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, p. 255–264. ACM, 1997.

[13] CABENA, P.; HADJINIAN, P.; STADLER, R.; VERHEES, J.; ZANASI, A. **Discovering Data Mining: From Concept to Implementation**. Prentice Hall, 1998.

[14] CANADA, S. **Statistics: Power from Data!** http://www.statcan.gc.ca/edu/power-pouvoir/toc-tdm/5214718-eng.htm, acessado em abril de 2009.

[15] CEGLAR, A.; RODDICK, J. F. **Association Mining**. *ACM Comput. Surv.*, 2006.

[16] CERBAH, F. **Mining the Content of Relational Databases to Learn Ontologies with Deeper Taxonomies**. *International Conference on Web Intelligence and Intelligent Agent Technology*, 2008.

[17] CHEN, X.; ZHOU, X.; SCHERL, R.; GELLER, J. **Using an Interest Ontology for Improved Support in Rule Mining**. In: *In 5th International Conference on Data Warehousing and Knowledge Discovery (DaWaK)*, 2003.

[18] CIMIANO, P. **Ontology Learning and Population from Text – Algorithms, Evalution and Applications**. Springer, 2006.

[19] CIOS, K. J.; PEDRYCZ, W.; SWINIARSKI, R. W.; KURGAN, L. A. **Data Mining - A Knowledge Discovery Approach**. Springer, 2007.

[20] CORRÊA, A. C. G. **Recuperação de Documentos Baseada em Informação Semântica no Ambiente AMMO**. Dissertação de Mestrado, Universidade Federal de São Carlos, 2003.

[21] **DAML – DARPA Agent Markup Language**. Disponível em http://www.daml.org, acessado em janeiro de 2010.

[22] **DAML+OIL**. Disponível em http://www.daml.org, acessado em janeiro de 2010.

[23] DANTAS, G. F. L. **Análise Criminal**. Disponível em http://blogandoseguranca.blogspot.com/2007/11/anlise-criminal.html, acessado em abril de 2010.

[24] DANTAS, G. F. L.; SOUZA, N. G. **As Bases Introdutórias da Análise Criminal na Inteligência Policial**. *Revista do Núcleo de Estudo e Pesquisa em Segurança Pública e Defesa Social*, 2004.

[25] DIXON, M. **An Overview of Document Mining Technology**. *Computer Based Learning Unit, University of Leeds*, 1997.

[26] ELSAYED, A.; EL-BELTAGY, S. R.; RAFEA, M.; HEGAZY, O. **Applying Data Mining for Ontology Building**. *Conference On Statistics, Computer Science, and Operations Research*, 2007.

[27] FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. **From Data Mining to Knowledge Discovery in Databases**. *American Association for Artificial Intelligence*, 1996.

[28] FERNANDEZ, M.; GOMEZ-PEREZ, A.; JURISTO, N. **METHONTOLOGY: from Ontological Art Towards Ontological Engineering**. *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering*, 1997.

[29] FERREIRA, A. B. H. **Dicionário Aurélio da Língua Portuguesa**. Nova Fronteira, 2004.

[30] GE, J.; QIU, Y.; CHEN, Z. **Cooperative Recommendation Based on Ontology Construction**. *International Conference on Computer Science and Software Engineering*, 2008.

[31] GENG, L.; HAMILTON, H. J. **Interestingness Measures for Data Mining: A Survey**. *ACM Comput. Surv.*, 2006.

[32] GONÇALVES, E. C. **Regras de Associação e suas Medidas de Interesse Objetivas e Subjetivas**. *INFOCOMP Journal of Computer Science*, 2005.

[33] GOTTLIEB, S.; ARENBERG, S.; SINGH, R. **Crime Analysis: From First Report To Final Arrest**. The FBI Law Enforcement Bulletin, 1994.

[34] GRUBER, T. R. **What is an Ontology?** Disponível em http://www-ksl.stanford.edu/kst/what-is-an-ontology.html, acessado em Maio de 2009, 1992.

[35] GRUBER, T. R. **Toward Principles for the Design of Ontologies Used for Knowledge Sharing**. *International Journal Human-Computer Studies*, p. 907–928, 1993.

[36] GRUBER, T. R. **Ontology**. In: Liu, L.; Özsu, M. T., editors, *Encyclopedia of Database Systems*. Springer-Verlag, 2008.

[37] GRÜNINGER, M.; FOX, M. **Methodology for the Design and Evaluation of Ontologies**. *IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995.

[38] GUARINO, N. **The Ontological Level**. In: *Philosophy and the Cognitive Sciences*, p. 443–456. Holder-Pichler-Tempsky, 1994.

[39] GUARINO, N. **Understanding, Building and Using Ontologies**. *Int. J. Hum.-Comput. Stud.*, 46(2-3), 1997.

[40] GUARINO, N. **Formal Ontology and Information Systems**. *Formal Ontology in Information Systems*, 1998.

[41] HAHSLER, M. **A Comparison of Commonly Used Interest Measures for Association Rules**. http://michael.hahsler.net/research/association_rules/measures.html, acessado em Abril de 2009, 2009.

[42] HAMMERGREN, T. C.; SIMON, A. R. **Data Warehousing For Dummies**. Wiley Publishing, 2009.

[43] HAN, J.; KAMBER, M. **Data Mining: Concepts and Techniques**. Elsevier, 2006.

[44] HAND, D.; MANNILA, H.; SMYTH, P. **Principles of Data Mining**. MIT Press, 2001.

[45] HAROLD, E. R. **XML 1.1 Bible**. 3rd Edition, Wiley Publishing, 2004.

[46] HEIJS, G. V.; SCHREIBER, A. T.; WIELINGA, B. J. **Using Explicit Ontologies in KBS Development**. *International Journal of Human and Computer Studies*, 1996.

[47] HOLZINGER, A.; GEIERHOFER, R.; MÖDRITSCHER, F.; TATZL, R. **Semantic Information in Medical Information Systems: Utilization of Text Mining Techniques to Analyze Medical Diagnoses**. *Journal of Universal Computer Science*, 2008.

[48] **Java**. Disponível em http://java.sun.com/, acessado em maio de 2010.

[49] **Jena Official Site**. Disponível em http://jena.sourceforge.net, acessado em abril de 2010.

[50] KEIM, D. A. **Information Visualization and Visual Data Mining**. *IEEE Transactions on Visualization and Computer Graphics*, p. 1–8, 2002.

[51] **Kettle Pentaho Data Integration**. Disponível em http://kettle.pentaho.org/, acessado em maio de 2010.

[52] KNAUBLOCK, H. **Protégé-OWL**. Disponível em http://protege.stanford.edu/overview/protege-owl.html, acessado em Abril de 2010, 2003.

[53] LAROSE, D. T. **Discovering Knowledge in Data: An Introduction to Data Mining**. John Wiley and Sons, Inc, 2005.

[54] LENCA, P.; VAILLANT, B.; MEYER, P.; LALLICH, S. **Association Rule Interestingness Measures: Experimental and Theoretical Studies**. *Springer Berlin / Heidelberg*, 2007.

[55] LI, G.; SHENG, H.; FAN, X. **Incorporating Metadata Into Data Mining with Ontology**. *Institute of Electronics, Information and Communication Engineers*, 2007.

[56] LIU, B.; HSU, W.; CHEN, S.; MA, Y. **Analyzing the Subjective Interestingness of Association Rules**. *IEEE Intelligent Systems*, 15(5):47–55, 2000.

[57] LOH, S. **Abordagem Baseada em Conceitos para Descoberta de Conhecimento em Textos**. Tese de Doutorado, Universidade Federal do Rio Grande do Sul, 2001.

[58] LOPES, M. C. S. **Mineração de Dados Textuais Utilizando Técnicas de Clustering para o Idioma Português**. Tese de Doutorado, Universidade Federal do Rio de Janeiro, 2004.

[59] MAGALHÃES, L. C. **Análise Criminal e Mapeamento da Criminalidade – GIS**. *Revista Âmbito Jurídico*, 2010.

[60] MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. **An Introduction to Information Retrieval**. Cambridge University Press, 2009.

[61] MARINICA, C.; GUILLET, F.; BRIAND, H. **Post-Processing of Discovered Association Rules Using Ontologies**. *International Conference on Data Mining Workshops*, 2008.

[62] MCCUE, C. **Data Mining and Predictive Analysis - Intelligence Gathering and Crime Analysis**. Elsevier, 2007.

[63] MILLER, G. **WordNet. An Electronic Lexical Database**. *MIT Press*, 1998.

[64] MORAIS, E. A. M. **Contextualização de Documentos em Domínios Representados por Ontologias Utilizando Mineração de Textos**. Dissertação de Mestrado, Universidade Federal de Goiás, 2007.

[65] MOURA, M. F. **Proposta de Utilização de Mineração de Textos para Seleção, Classificação e Qualificação de Documentos**. *Revista Embrapa Informática Agropecuária*, 2004.

[66] MYATT, G. J. **Making Sense of Data - A Practical Guide to Exploratory Data Analysis and Data Mining**. John Wiley and Sons, Inc, 2007.

[67] MYATT, G. J.; JOHNSON, W. P. **Making Sense of Data II - A Practical Guide to Data Visualization, Advanced Data Mining Methods, and Applications**. John Wiley and Sons, Inc, 2009.

[68] NIGRO, H. O.; CÍSARO, S. E. G.; XODO, D. H. **Data Mining with Ontologies: Implementations, Findings, and Frameworks**. Information Science Reference, 2007.

[69] NIST/SEMATECH. **NIST/SEMATECH e-Handbook of Statistical Methods**. http://www.itl.nist.gov/div898/handbook/, acessado em abril de 2009.

[70] NOY, N.; MCGUINNESS, D. **Ontology Development 101: A Guide to Creating Your First Ontology**. *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05*, 2001.

[71] **OIL – Ontology Inference Layer**. Disponível em http://www.ontoknowledge.org/oil, acessado em janeiro de 2010.

[72] OLIVEIRA, D.; BAIÃO, F.; MATTOSO, M. **MF-Ontology, uma Ontologia para o Processo de Mineração de Textos**. *Seminário de Pesquisa em Ontologia no Brasil*, 2008.

[73] OLIVEIRA, L. C. **Meta-Modelo Funcional para Recuperação de Informação**. Dissertação de Mestrado, Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Uberlândia, 2004.

[74] OLIVEIRA, R. R.; CAMILO, C. O.; CARVALHO, C. L.; SILVA, J. C. **Recuperação Contextualizada de Documentos em Bibliotecas Digitais Integradas**. *8th International Information and Telecommunication Technologies Symposium*, 2009.

[75] OLSON, D. L.; DELEN, D. **Advanced Data Mining Techniques**. Springer, 2008.

[76] **OWL - OWL Web Ontology Language**. Disponível em http://www.w3.org/TR/owl-ref, acessado em Abril de 2010.

[77] PANOV, P.; DZEROSKI, S.; SOLDATOVA, L. N. **OntoDM: An Ontology of Data Mining**. *International Conference on Data Mining Workshops*, 2008.

[78] PATRICK, J. J. **SQL Fundamentals**. Prentice Hall PTR, 2002.

[79] PAYNTER, G.; TRIGG, L.; FRANK, E.; KIRKBY, R. **Attribute-Relation File Format (ARFF)**. http://www.cs.waikato.ac.nz/~ml/weka/arff.html, acessado em Novembro de 2009, 2008.

[80] **Pentaho Open Source Business Intelligence**. Disponível em http://www.pentaho.com/, acessado em maio de 2010.

[81] PEREIRA, F. C.; GROSZ, B. J. **Natural Language Processing**. MIT Press, 1994.

[82] PIATETSKY-SHAPIRO, G. **Discovery, Analysis, and Presentation of Strong Rules**. In: *Knowledge Discovery in Databases*. American Association for Artificial Intelligence, 1991.

[83] PM, M.; DW, A. **UCI Repository of Machine Learning Databases**. http://www.ics.uci.edu/, acessado em abril de 2009.

[84] PONNIAH, P. **Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals**. John Wiley & Sons, 2001.

[85] **PTStemmer – A Java Stemming Toolkit for the Portuguese Language**. Disponível em http://code.google.com/p/ptstemmer, acessado em abril de 2010.

[86] QUEIROZ, C. A. M. **Manual de Polícia Judiciária - Doutrina, Modelos, Legislação**. Polícia Civil do Estado de São Paulo, 2007.

[87] REED, S.; LENAT, D. **Mapping Ontologies into Cyc**. *Cycorp, Inc*, 2002.

[88] REZENDE, S. O. **Mineração de Dados**. *XXV Congresso da Sociedade Brasileira de Computação*, 2005.

[89] SALTON, G.; MCGILL, M. J. **Introduction to Modern Information Retrieval**. Computer Science Series, 1983.

[90] SALTON, G.; BUCKLEY, C. **Term-Weighting Approaches in Automatic Retrieval**. Information Processing & Management, 1988.

[91] SANTOS, M. A. M. R. **Extraindo Regras de Associação a partir de Textos**. Dissertação de Mestrado, Pontifícia Universidade Católica do Paraná, 2002.

[92] SELIYA, N.; KHOSHGOFTAAR, T. M. **Software Quality Modeling With Limited Apriori Defect Data**, chapter Chapter 1, p. 1–16. Idea Group Publishing, 2007.

[93] **Overview of SGML Resources**. Disponível em http://www.w3.org/MarkUp/SGML/, acessado em maio de 2010.

[94] SHLENS, J. **A Tutorial on Principal Component Analysis**. Salk Insitute for Biological Studies and University of California, 2 edition, December 2005.

[95] **SHOE – Simple HTML Ontology Extensions**. Disponível em http://www.cs.umd.edu/projects/plus/SHOE, acessado em janeiro de 2010.

[96] SILBERSCHATZ, A.; TUZHILIN, A. **What Makes Patterns Interesting in Knowledge Discovery Systems**. *IEEE Transactions on Knowledge and Data Engineering*, 1996.

[97] SILVEIRA, M. L. **Recuperação Vertical de Informação: Um Estudo de Caso na Área Jurídica**. Tese de Doutorado, Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Minas Gerais, 2003.

[98] SIMOFF, S. J.; BÖHLEN, M. H.; MAZEIKA, A. **Visual Data Mining - Theory, Techniques and Tools for Visual Analytics**. Springer, 2008.

[99] SINOARA, R. A. **Identificação de Regras de Associação Interessantes por Meio de Análises com Medidas Objetivas e Subjetivas**. Dissertação de Mestrado, USP - São Carlos, 2006.

[100] STAAB, S.; SCHNURR, H.; STUDER, R.; SURE, Y. **Knowledge Processes and Ontologies**. *IEEE Intelligent Systems*, 2000.

[101] SWARTOUT, B.; PATIL, R.; KNIGHT, K.; RUSS, T. **Toward Distributed Use of Large-Scale Ontologies**. *Banff Knowledge Aquisition Workshop*, 1997.

[102] TAN, A. **Text Mining: The State of the Art and the Challenges**. *Workshop on Knowlegde Discovery from Advanced Databases*, 1999.

[103] TAN, P.; KUMAR, V.; SRIVASTAVA, J. **Selecting the Right Interestingness Measure for Association Patterns**. In: *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 32–41. ACM, 2002.

[104] USCHOLD, M.; KING, M. **Towards a Methodology for Building Ontologies**. *Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95*, 1995.

[105] VARGAS, G. **Código Penal Brasileiro**. Disponível em http://www.planalto.gov.br/ccivil/decreto-lei/Del2848compilado.htm, acessado em Maio de 2010, 1940.

[106] VIVACQUA, A. S.; GARCIA, A. C. B. **Mineração de Dados Baseada em Ontologia**. *Seminário de Pesquisa em Ontologia no Brasil*, 2008.

[107] **W3C – World Wide Web Consortium**. Disponível em http://www.w3.org, acessado em janeiro de 2010.

[108] WAIKATO, U. O. **WEKA**. http://www.cs.waikato.ac.nz/ml/weka/, acessado em Maio de 2009.

[109] Wang, J., editor. **Encyclopedia of Data Warehousing and Mining**. Idea Group Reference, 2005.

[110] WANG, J.; HU, X.; ZHU, D. **Minimizing the Minus Sides of Mining Data**. In: Taniar, D., editor, *Data Mining and Knowledge Discovery Technologies*, p. 254–279. IGI Publishing, 2008.

[111] WEB, W. S. **W3C Semantic Web Activity**. http://www.w3.org/2001/sw/, acessado em Maio de 2009, 2003.

[112] WITTEN, I. H.; FRANK, E. **Data Mining - Practical Machine Learning Tools and Techniques**. Elsevier, 2005.

[113] WIVES, L. K. **Um Estudo Sobre Agrupamento de Documentos Textuais em Processamento de Informações não Estruturadas Usando Técnicas de Clustering**. Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, 1999.

[114] WIVES, L. K. **Utilizando Conceitos como Descritores de Textos para o Processo de Identificação de Conglomerados (Clustering) de Documentos**. Tese de Doutorado, Universidade Federal Do Rio Grande Do Sul, 2004.

[115] WONG, T.; CHOW, K.; WANG, F. L. **An Unsupervised Learning Framework for Discovering the Site-Specific Ontology from Multiple Web Pages**. *International Conference on Machine Learning and Cybernetics*, 2008.

[116] **XOL – Ontology Exchange Language**. Disponível em http://www.ai.sri.com/pkarp/xol, acessado em janeiro de 2010.

[117] ZUFFO, J. A. **A Sociedade e a Economia no Novo Milênio**. *Editora Manole*, 2002.

APÊNDICE A

Arquivo OWL da Ontologia de domínio criada

Neste Apêndice, é apresentado o arquivo OWL (*Ontology Web Language*) para a Ontologia criada.

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
    <!ENTITY OntologiaHomicidio "http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#" >
]>

<rdf:RDF xmlns="http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#"
     xml:base="http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl"
     xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
     xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:owl="http://www.w3.org/2002/07/owl#"
     xmlns:OntologiaHomicidio="http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#">
    <owl:Ontology rdf:about=""/>

    <!--
    ///////////////////////////////////////////////////////////////////////////////////////
    //
    // Object Properties
    //
    ///////////////////////////////////////////////////////////////////////////////////////
     -->

    <!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#compostoPor -->
    <owl:ObjectProperty rdf:about="#compostoPor"/>

    <!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#minimoDe -->
    <owl:ObjectProperty rdf:about="#minimoDe"/>

    <!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#temUm -->
    <owl:ObjectProperty rdf:about="#temUm"/>

    <!--
    ///////////////////////////////////////////////////////////////////////////////////////
```

```
//
// Classes
//
///////////////////////////////////////////////////////////////////////////////
 -->

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Abalroamento -->
<owl:Class rdf:about="#Abalroamento">
    <rdfs:subClassOf rdf:resource="#BatidaValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Acidente -->
<owl:Class rdf:about="#Acidente">
    <rdfs:subClassOf rdf:resource="#Meio"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#AcidenteTransito -->
<owl:Class rdf:about="#AcidenteTransito">
    <rdfs:subClassOf rdf:resource="#Meio"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Afanar -->
<owl:Class rdf:about="#Afanar">
    <rdfs:subClassOf rdf:resource="#RoubolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Agrediu -->
<owl:Class rdf:about="#Agrediu">
    <rdfs:subClassOf rdf:resource="#BrigasValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Alcool -->
<owl:Class rdf:about="#Alcool">
    <rdfs:subClassOf rdf:resource="#Motiva&#231;&#227;o"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#compostoPor"/>
            <owl:someValuesFrom rdf:resource="#AlcoolValuePartition"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#AlcoolValuePartition -->
<owl:Class rdf:about="#AlcoolValuePartition">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Bebendo"/>
                <rdf:Description rdf:about="#Bebida"/>
                <rdf:Description rdf:about="#Buteco"/>
                <rdf:Description rdf:about="#Embriagues"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#ValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Alvejada -->
```

```
<owl:Class rdf:about="#Alvejada">
    <rdfs:subClassOf rdf:resource="#ArmaDeFogolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Amante -->
<owl:Class rdf:about="#Amante">
    <rdfs:subClassOf rdf:resource="#PassionallValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Amasiado -->
<owl:Class rdf:about="#Amasiado">
    <rdfs:subClassOf rdf:resource="#PassionallValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Arma -->
<owl:Class rdf:about="#Arma">
    <rdfs:subClassOf rdf:resource="#ArmaDeFogolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#ArmaBranca -->
<owl:Class rdf:about="#ArmaBranca">
    <rdfs:subClassOf rdf:resource="#Meio"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#ArmaDeFogo -->
<owl:Class rdf:about="#ArmaDeFogo">
    <rdfs:subClassOf rdf:resource="#Meio"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#minimoDe"/>
            <owl:onClass rdf:resource="#Quantidade"/>
            <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">2</owl:minCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#compostoPor"/>
            <owl:someValuesFrom rdf:resource="#ArmaDeFogolValuePartition"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#ArmaDeFogolValuePartition -->
<owl:Class rdf:about="#ArmaDeFogolValuePartition">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Alvejada"/>
                <rdf:Description rdf:about="#Atirar"/>
                <rdf:Description rdf:about="#Bala"/>
                <rdf:Description rdf:about="#Baleado"/>
                <rdf:Description rdf:about="#Disparos"/>
                <rdf:Description rdf:about="#Efetuou"/>
                <rdf:Description rdf:about="#PAF"/>
                <rdf:Description rdf:about="#Tiros"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
```

```
    <rdfs:subClassOf rdf:resource="#ValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Assassinado -->
<owl:Class rdf:about="#Assassinado">
    <rdfs:subClassOf rdf:resource="#HomicidiolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Atacado -->
<owl:Class rdf:about="#Atacado">
    <rdfs:subClassOf rdf:resource="#PedraValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Atingido -->
<owl:Class rdf:about="#Atingido">
    <rdfs:subClassOf rdf:resource="#PedraValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Atirar -->
<owl:Class rdf:about="#Atirar">
    <rdfs:subClassOf rdf:resource="#ArmaDeFogolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Atropelamento -->
<owl:Class rdf:about="#Atropelamento">
    <rdfs:subClassOf rdf:resource="#AcidenteTransito"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#compostoPor"/>
            <owl:someValuesFrom rdf:resource="#AtropelamentoValuePartition"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#minimoDe"/>
            <owl:onClass rdf:resource="#Quantidade"/>
            <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">2</owl:minCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#AtropelamentoValuePartition -->
<owl:Class rdf:about="#AtropelamentoValuePartition">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Ciclista"/>
                <rdf:Description rdf:about="#Encostamento"/>
                <rdf:Description rdf:about="#Pedestre"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#ValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Autor -->
<owl:Class rdf:about="#Autor">
    <rdfs:subClassOf rdf:resource="#HomicidiolValuePartition"/>
```

```
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Bala -->
<owl:Class rdf:about="#Bala">
    <rdfs:subClassOf rdf:resource="#ArmaDeFogolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Baldio -->
<owl:Class rdf:about="#Baldio">
    <rdfs:subClassOf rdf:resource="#LoteValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Baleado -->
<owl:Class rdf:about="#Baleado">
    <rdfs:subClassOf rdf:resource="#ArmaDeFogolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Bar -->
<owl:Class rdf:about="#Bar">
    <rdfs:subClassOf rdf:resource="#Local"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Batida -->
<owl:Class rdf:about="#Batida">
    <rdfs:subClassOf rdf:resource="#AcidenteTransito"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#compostoPor"/>
            <owl:someValuesFrom rdf:resource="#BatidaValuePartition"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#minimoDe"/>
            <owl:onClass rdf:resource="#Quantidade"/>
            <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">2</owl:minCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#BatidaValuePartition -->
<owl:Class rdf:about="#BatidaValuePartition">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Abalroamento"/>
                <rdf:Description rdf:about="#Colis&#227;o"/>
                <rdf:Description rdf:about="#Condutor"/>
                <rdf:Description rdf:about="#Ve&#237;culo"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#ValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Bebendo -->
<owl:Class rdf:about="#Bebendo">
    <rdfs:subClassOf rdf:resource="#AlcoolValuePartition"/>
```

```
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Bebida -->
<owl:Class rdf:about="#Bebida">
    <rdfs:subClassOf rdf:resource="#AlcoolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Brigas -->
<owl:Class rdf:about="#Brigas">
    <rdfs:subClassOf rdf:resource="#Motiva&#231;&#227;o"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#compostoPor"/>
            <owl:someValuesFrom rdf:resource="#BrigasValuePartition"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#minimoDe"/>
            <owl:onClass rdf:resource="#Quantidade"/>
            <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">2</owl:minCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#BrigasValuePartition -->
<owl:Class rdf:about="#BrigasValuePartition">
    <rdfs:subClassOf rdf:resource="#ValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Buteco -->
<owl:Class rdf:about="#Buteco">
    <rdfs:subClassOf rdf:resource="#AlcoolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Cadaver -->
<owl:Class rdf:about="#Cadaver">
    <rdfs:subClassOf rdf:resource="#HomicidiolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Cair -->
<owl:Class rdf:about="#Cair">
    <owl:equivalentClass rdf:resource="#Queda"/>
    <rdfs:subClassOf rdf:resource="#Acidente"/>
    <owl:disjointWith rdf:resource="#Derrame"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Canivete -->
<owl:Class rdf:about="#Canivete">
    <rdfs:subClassOf rdf:resource="#ArmaBranca"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Casa -->
<owl:Class rdf:about="#Casa">
    <owl:equivalentClass rdf:resource="#Resid&#234;ncia"/>
    <rdfs:subClassOf rdf:resource="#Local"/>
</owl:Class>
```

```
<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Chutar -->
<owl:Class rdf:about="#Chutar">
    <rdfs:subClassOf rdf:resource="#BrigasValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Ciclista -->
<owl:Class rdf:about="#Ciclista">
    <rdfs:subClassOf rdf:resource="#AtropelamentoValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Ci&#250;me -->
<owl:Class rdf:about="#Ci&#250;me">
    <rdfs:subClassOf rdf:resource="#PassionallValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Colis&#227;o -->
<owl:Class rdf:about="#Colis&#227;o">
    <rdfs:subClassOf rdf:resource="#BatidaValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Comportamento -->
<owl:Class rdf:about="#Comportamento"/>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Condutor -->
<owl:Class rdf:about="#Condutor">
    <rdfs:subClassOf rdf:resource="#BatidaValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Corpo -->
<owl:Class rdf:about="#Corpo">
    <rdfs:subClassOf rdf:resource="#HomicidiolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Correu -->
<owl:Class rdf:about="#Correu">
    <rdfs:subClassOf rdf:resource="#EvadiulValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Cortou -->
<owl:Class rdf:about="#Cortou">
    <rdfs:subClassOf rdf:resource="#FacaValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Derrame -->
<owl:Class rdf:about="#Derrame">
    <rdfs:subClassOf rdf:resource="#Acidente"/>
    <owl:disjointWith rdf:resource="#Queda"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Desentendimento -->
<owl:Class rdf:about="#Desentendimento">
    <rdfs:subClassOf rdf:resource="#BrigasValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Desferiu -->
<owl:Class rdf:about="#Desferiu">
    <rdfs:subClassOf rdf:resource="#FacaValuePartition"/>
</owl:Class>
```

```
<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Discuss&#227;o -->
<owl:Class rdf:about="#Discuss&#227;o">
    <rdfs:subClassOf rdf:resource="#BrigasValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Disparos -->
<owl:Class rdf:about="#Disparos">
    <rdfs:subClassOf rdf:resource="#ArmaDeFogolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Drogas -->
<owl:Class rdf:about="#Drogas">
    <rdfs:subClassOf rdf:resource="#VingancalValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Efetuou -->
<owl:Class rdf:about="#Efetuou">
    <rdfs:subClassOf rdf:resource="#ArmaDeFogolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Embriagues -->
<owl:Class rdf:about="#Embriagues">
    <rdfs:subClassOf rdf:resource="#AlcoolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Encaminhado -->
<owl:Class rdf:about="#Encaminhado">
    <rdfs:subClassOf rdf:resource="#SocorridoValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Encostamento -->
<owl:Class rdf:about="#Encostamento">
    <rdfs:subClassOf rdf:resource="#AtropelamentoValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Entorpecentes -->
<owl:Class rdf:about="#Entorpecentes">
    <rdfs:subClassOf rdf:resource="#VingancalValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Esmurrar -->
<owl:Class rdf:about="#Esmurrar">
    <rdfs:subClassOf rdf:resource="#BrigasValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Espancando -->
<owl:Class rdf:about="#Espancando">
    <rdfs:subClassOf rdf:resource="#BrigasValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Espingarda -->
<owl:Class rdf:about="#Espingarda">
    <rdfs:subClassOf rdf:resource="#ArmaDeFogo"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Esquartejou -->
<owl:Class rdf:about="#Esquartejou">
```

```
    <rdfs:subClassOf rdf:resource="#FacaValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Estacionamento -->
<owl:Class rdf:about="#Estacionamento">
    <rdfs:subClassOf rdf:resource="#Local"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Evadiu -->
<owl:Class rdf:about="#Evadiu">
    <rdfs:subClassOf rdf:resource="#EvadiulValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Evadiu-se -->
<owl:Class rdf:about="#Evadiu-se">
    <rdfs:subClassOf rdf:resource="#Comportamento"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#compostoPor"/>
            <owl:someValuesFrom rdf:resource="#EvadiulValuePartition"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="#Socorrido"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#EvadiulValuePartition -->
<owl:Class rdf:about="#EvadiulValuePartition">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Correu"/>
                <rdf:Description rdf:about="#Evadiu"/>
                <rdf:Description rdf:about="#Foragido"/>
                <rdf:Description rdf:about="#Fugiu"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#ValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Faca -->
<owl:Class rdf:about="#Faca">
    <rdfs:subClassOf rdf:resource="#ArmaBranca"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#compostoPor"/>
            <owl:someValuesFrom rdf:resource="#FacaValuePartition"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#minimoDe"/>
            <owl:onClass rdf:resource="#Quantidade"/>
            <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">2</owl:minCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

```
<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#FacaValuePartition -->
<owl:Class rdf:about="#FacaValuePartition">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Cortou"/>
                <rdf:Description rdf:about="#Desferiu"/>
                <rdf:Description rdf:about="#Esquartejou"/>
                <rdf:Description rdf:about="#Facadas"/>
                <rdf:Description rdf:about="#Furou"/>
                <rdf:Description rdf:about="#Perfurou"/>
                <rdf:Description rdf:about="#Picou"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#ValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Facadas -->
<owl:Class rdf:about="#Facadas">
    <rdfs:subClassOf rdf:resource="#FacaValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Foragido -->
<owl:Class rdf:about="#Foragido">
    <rdfs:subClassOf rdf:resource="#EvadiulValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Fugiu -->
<owl:Class rdf:about="#Fugiu">
    <rdfs:subClassOf rdf:resource="#EvadiulValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Furou -->
<owl:Class rdf:about="#Furou">
    <rdfs:subClassOf rdf:resource="#FacaValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#F&#250;til -->
<owl:Class rdf:about="#F&#250;til">
    <rdfs:subClassOf rdf:resource="#Motiva&#231;&#227;o"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Golpeado -->
<owl:Class rdf:about="#Golpeado">
    <rdfs:subClassOf rdf:resource="#PedraValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Gritar -->
<owl:Class rdf:about="#Gritar">
    <rdfs:subClassOf rdf:resource="#BrigasValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#HUGO -->
<owl:Class rdf:about="#HUGO">
    <rdfs:subClassOf rdf:resource="#SocorridoValuePartition"/>
</owl:Class>
```

```
<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Homicidio -->
<owl:Class rdf:about="#Homicidio">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#temUm"/>
            <owl:someValuesFrom rdf:resource="#Local"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#minimoDe"/>
            <owl:onClass rdf:resource="#Quantidade"/>
            <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">3</owl:minCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#temUm"/>
            <owl:someValuesFrom rdf:resource="#Meio"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#compostoPor"/>
            <owl:someValuesFrom rdf:resource="#HomicidiolValuePartition"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#temUm"/>
            <owl:someValuesFrom rdf:resource="#Motiva&#231;&#227;o"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#temUm"/>
            <owl:someValuesFrom rdf:resource="#Comportamento"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#HomicidiolValuePartition -->
<owl:Class rdf:about="#HomicidiolValuePartition">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Assassinado"/>
                <rdf:Description rdf:about="#Cadaver"/>
                <rdf:Description rdf:about="#Corpo"/>
                <rdf:Description rdf:about="#Morrer"/>
                <rdf:Description rdf:about="#Morte"/>
                <rdf:Description rdf:about="#Obito"/>
                <rdf:Description rdf:about="#V&#237;tima"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#ValuePartition"/>
</owl:Class>
```

```
<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Hospital -->
<owl:Class rdf:about="#Hospital">
    <rdfs:subClassOf rdf:resource="#Local"/>
    <rdfs:subClassOf rdf:resource="#SocorridoValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Ladr&#227;o -->
<owl:Class rdf:about="#Ladr&#227;o">
    <rdfs:subClassOf rdf:resource="#RoubolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Levado -->
<owl:Class rdf:about="#Levado">
    <rdfs:subClassOf rdf:resource="#SocorridoValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Local -->
<owl:Class rdf:about="#Local"/>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Lote -->
<owl:Class rdf:about="#Lote">
    <rdfs:subClassOf rdf:resource="#Local"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#minimoDe"/>
            <owl:onClass rdf:resource="#Quantidade"/>
            <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">2</owl:minCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#compostoPor"/>
            <owl:someValuesFrom rdf:resource="#LoteValuePartition"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#LoteValuePartition -->
<owl:Class rdf:about="#LoteValuePartition">
    <rdfs:subClassOf rdf:resource="#ValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Meio -->
<owl:Class rdf:about="#Meio">
    <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Morrer -->
<owl:Class rdf:about="#Morrer">
    <rdfs:subClassOf rdf:resource="#HomicidiolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Morte -->
<owl:Class rdf:about="#Morte">
    <rdfs:subClassOf rdf:resource="#HomicidiolValuePartition"/>
</owl:Class>
```

```
<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Motiva&#231;&#227;o -->
<owl:Class rdf:about="#Motiva&#231;&#227;o">
    <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Namorada -->
<owl:Class rdf:about="#Namorada">
    <rdfs:subClassOf rdf:resource="#PassionallValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Obito -->
<owl:Class rdf:about="#Obito">
    <rdfs:subClassOf rdf:resource="#HomicidiolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#PAF -->
<owl:Class rdf:about="#PAF">
    <rdfs:subClassOf rdf:resource="#ArmaDeFogolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Passional -->
<owl:Class rdf:about="#Passional">
    <rdfs:subClassOf rdf:resource="#Motiva&#231;&#227;o"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#compostoPor"/>
            <owl:someValuesFrom rdf:resource="#PassionallValuePartition"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#PassionallValuePartition -->
<owl:Class rdf:about="#PassionallValuePartition">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Amante"/>
                <rdf:Description rdf:about="#Amasiado"/>
                <rdf:Description rdf:about="#Ci&#250;me"/>
                <rdf:Description rdf:about="#Namorada"/>
                <rdf:Description rdf:about="#Trai&#231;&#227;o"/>
                <rdf:Description rdf:about="#Uni&#227;o_Est&#225;vel"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#ValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Pau -->
<owl:Class rdf:about="#Pau">
    <rdfs:subClassOf rdf:resource="#ArmaBranca"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Pedestre -->
<owl:Class rdf:about="#Pedestre">
    <rdfs:subClassOf rdf:resource="#AtropelamentoValuePartition"/>
</owl:Class>
```

```
<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Pedra -->
<owl:Class rdf:about="#Pedra">
    <rdfs:subClassOf rdf:resource="#ArmaBranca"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#compostoPor"/>
            <owl:someValuesFrom rdf:resource="#PedraValuePartition"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#minimoDe"/>
            <owl:onClass rdf:resource="#Quantidade"/>
            <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">2</owl:minCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#PedraValuePartition -->
<owl:Class rdf:about="#PedraValuePartition">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Atacado"/>
                <rdf:Description rdf:about="#Atingido"/>
                <rdf:Description rdf:about="#Golpeado"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#ValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Perfurou -->
<owl:Class rdf:about="#Perfurou">
    <rdfs:subClassOf rdf:resource="#FacaValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Picou -->
<owl:Class rdf:about="#Picou">
    <rdfs:subClassOf rdf:resource="#FacaValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Pistola -->
<owl:Class rdf:about="#Pistola">
    <rdfs:subClassOf rdf:resource="#ArmaDeFogo"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Quantidade -->
<owl:Class rdf:about="#Quantidade">
    <rdfs:subClassOf rdf:resource="#ValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Queda -->
<owl:Class rdf:about="#Queda">
    <rdfs:subClassOf rdf:resource="#Acidente"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Resid&#234;ncia -->
```

```
<owl:Class rdf:about="#Resid&#234;ncia">
    <rdfs:subClassOf rdf:resource="#Local"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Rev&#243;lver -->
<owl:Class rdf:about="#Rev&#243;lver">
    <rdfs:subClassOf rdf:resource="#ArmaDeFogo"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Roubar -->
<owl:Class rdf:about="#Roubar">
    <rdfs:subClassOf rdf:resource="#RoubolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Roubo -->
<owl:Class rdf:about="#Roubo">
    <rdfs:subClassOf rdf:resource="#Motiva&#231;&#227;o"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#compostoPor"/>
            <owl:someValuesFrom rdf:resource="#RoubolValuePartition"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#RoubolValuePartition -->
<owl:Class rdf:about="#RoubolValuePartition">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Afanar"/>
                <rdf:Description rdf:about="#Ladr&#227;o"/>
                <rdf:Description rdf:about="#Roubar"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#ValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Rua -->
<owl:Class rdf:about="#Rua">
    <rdfs:subClassOf rdf:resource="#Local"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Socorrido -->
<owl:Class rdf:about="#Socorrido">
    <rdfs:subClassOf rdf:resource="#Comportamento"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#compostoPor"/>
            <owl:someValuesFrom rdf:resource="#SocorridoValuePartition"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#minimoDe"/>
            <owl:onClass rdf:resource="#Quantidade"/>
            <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">2</owl:minCardinality>
```

```
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#SocorridoValuePartition -->
<owl:Class rdf:about="#SocorridoValuePartition">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Encaminhado"/>
                <rdf:Description rdf:about="#HUGO"/>
                <rdf:Description rdf:about="#Hospital"/>
                <rdf:Description rdf:about="#Levado"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
    <rdfs:subClassOf rdf:resource="#ValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Tiros -->
<owl:Class rdf:about="#Tiros">
    <rdfs:subClassOf rdf:resource="#ArmaDeFogolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Trai&#231;&#227;o -->
<owl:Class rdf:about="#Trai&#231;&#227;o">
    <rdfs:subClassOf rdf:resource="#PassionallValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Tr&#225;fego -->
<owl:Class rdf:about="#Tr&#225;fego">
    <rdfs:subClassOf rdf:resource="#VingancalValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Uni&#227;o_Est&#225;vel -->
<owl:Class rdf:about="#Uni&#227;o_Est&#225;vel">
    <rdfs:subClassOf rdf:resource="#PassionallValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#ValuePartition -->
<owl:Class rdf:about="#ValuePartition"/>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Ve&#237;culo -->
<owl:Class rdf:about="#Ve&#237;culo">
    <rdfs:subClassOf rdf:resource="#BatidaValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#VingancalValuePartition -->
<owl:Class rdf:about="#VingancalValuePartition">
    <owl:equivalentClass>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <rdf:Description rdf:about="#Drogas"/>
                <rdf:Description rdf:about="#Entorpecentes"/>
                <rdf:Description rdf:about="#Tr&#225;fego"/>
            </owl:unionOf>
        </owl:Class>
    </owl:equivalentClass>
```

```
    <rdfs:subClassOf rdf:resource="#ValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Vingan&#231;a -->
<owl:Class rdf:about="#Vingan&#231;a">
    <rdfs:subClassOf rdf:resource="#Motiva&#231;&#227;o"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#compostoPor"/>
            <owl:someValuesFrom rdf:resource="#VingancalValuePartition"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#minimoDe"/>
            <owl:onClass rdf:resource="#Quantidade"/>
            <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">2</owl:minCardinality>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#V&#237;tima -->
<owl:Class rdf:about="#V&#237;tima">
    <rdfs:subClassOf rdf:resource="#HomicidiolValuePartition"/>
</owl:Class>

<!-- http://www.semanticweb.org/ontologies/2009/10/OntologiaHomicidio.owl#Xingar -->
<owl:Class rdf:about="#Xingar">
    <rdfs:subClassOf rdf:resource="#BrigasValuePartition"/>
</owl:Class>

<!-- http://www.w3.org/2002/07/owl#Thing -->
<owl:Class rdf:about="&owl;Thing"/>

<!--
///////////////////////////////////////////////////////////////////////////////////////
//
// General axioms
//
///////////////////////////////////////////////////////////////////////////////////////
 -->
<rdf:Description>
    <rdf:type rdf:resource="&owl;AllDisjointClasses"/>
    <owl:members rdf:parseType="Collection">
        <rdf:Description rdf:about="#Espingarda"/>
        <rdf:Description rdf:about="#Pistola"/>
        <rdf:Description rdf:about="#Rev&#243;lver"/>
    </owl:members>
</rdf:Description>
<rdf:Description>
    <rdf:type rdf:resource="&owl;AllDisjointClasses"/>
    <owl:members rdf:parseType="Collection">
        <rdf:Description rdf:about="#Bar"/>
        <rdf:Description rdf:about="#Casa"/>
        <rdf:Description rdf:about="#Estacionamento"/>
        <rdf:Description rdf:about="#Hospital"/>
        <rdf:Description rdf:about="#Lote"/>
        <rdf:Description rdf:about="#Resid&#234;ncia"/>
```

```
            <rdf:Description rdf:about="#Rua"/>
        </owl:members>
    </rdf:Description>
    <rdf:Description>
        <rdf:type rdf:resource="&owl;AllDisjointClasses"/>
        <owl:members rdf:parseType="Collection">
            <rdf:Description rdf:about="#Canivete"/>
            <rdf:Description rdf:about="#Faca"/>
            <rdf:Description rdf:about="#Pau"/>’
        </owl:members>
    </rdf:Description>
</rdf:RDF>
<!-- Generated by the OWL API (version 2.2.1.1138) http://owlapi.sourceforge.net -->
```

APÊNDICE B

Códigos Java utilizado

Neste Apêndice, são apresentados os principais trechos do código Java utilizado para realização do *stemming*, da extração de conceitos da Ontologia, do cálculo de similaridade e da filtragem das regras.

B.1 Método para realizar o *stemming*

Listing B.1: *Função utilizada para normalizar o texto*

```
1  public String tratarHistorico(String texto) {
2    texto = limparTexto.retirarPontuacao(texto);
3
4    String textoNovo = "";
5
6    String[] tokens = texto.split("␣");
7
8    String stPalavra;
9      for (int i = 0; i < tokens.length; i++) {
10
11       stPalavra = st.wordStemming(tokens[i].toUpperCase());
12
13         if ((stPalavra.length() > 2 &&
14               stPalavra.length() < 20) &&
15               stopWords.indexOf(stPalavra) == -1) {
16           textoNovo += stPalavra + "␣";
17         }
18     }
19
20
21   return textoNovo;
22
23 }
```

B.2 Método para recuperar os conceitos da Ontologia

Listing B.2: *Método utilizado para recuperar os conceitos da Ontologia*

```
1  public HashMap tratarOntologia() {
2    OntDocumentManager dm = modelo.getDocumentManager();
3    dm.addAltEntry(URI, "file:" + file);
4    modelo.read(URI);
5
6    //Busca
7    OntClass homicidio = modelo.getOntClass(URI + "#Homicidio");
8
9    int qtde = 0;
10   ArrayList al = new ArrayList();
11   al.add(st.wordStemming(limparTexto.retirarPontuacao(homicidio.
         getLocalName())));
12
13   for (Iterator i = homicidio.listSuperClasses(true); i.hasNext();) {
14     OntClass c = (OntClass) i.next();
15
16     listaTermos = new ArrayList();
17
18     if (c.isRestriction()) {
19       Restriction r = c.asRestriction();
20
21       if (r.isSomeValuesFromRestriction()) {
22         SomeValuesFromRestriction av = r.asSomeValuesFromRestriction();
23
24         if (av.getOnProperty().getLocalName().contentEquals("temUm")) {
25           this.trataTemUm(av.getSomeValuesFrom().getURI());
26           vetorTemUm.put(av.getSomeValuesFrom().getURI(), listaTermos);
27         } else if (av.getOnProperty().getLocalName().contentEquals("
              compostoPor")) {
28           OntClass cp = modelo.getOntClass(av.getSomeValuesFrom().
                getURI());
29
30           for (Iterator ii = cp.listSubClasses(true); ii.hasNext();) {
31             OntClass c2 = (OntClass) ii.next();
32             al.add(st.wordStemming(limparTexto.retirarPontuacao(c2.
                  getLocalName())));
33           }
34         }
35       } else if (r.isMinCardinalityRestriction()) {
36         MinCardinalityRestriction av = r.asMinCardinalityRestriction();
37
```

```
38          if (av.getOnProperty().getLocalName().contentEquals("minimoDe")
                ) {
39            qtde = av.getMinCardinality();
40          }
41        }
42       }
43    }
44
45    listaTermos = new ArrayList();
46    NoOntologia no = new NoOntologia();
47    no.conceito = homicidio.getLocalName();
48    if (homicidio.listDisjointWith().toList().size() > 0) {
49     no.distintas = homicidio.listDisjointWith().toList();
50    }
51    no.termos = al;
52    no.minimoTermos = qtde;
53    listaTermos.add(no);
54    vetorTemUm.put(homicidio.getLocalName(), listaTermos);
55
56    return vetorTemUm;
57 }
```

Listing B.3: *Método auxiliar do método B.2*

```
1  private String[] trataTemUm(String uri) {
2    OntClass cp = modelo.getOntClass(uri);
3
4    Iterator ii = cp.listSubClasses(true);
5    if (ii.hasNext()) {
6      do {
7        OntClass c2 = (OntClass) ii.next();
8
9        if (c2.listSubClasses(true).hasNext() == true) {
10         NoOntologia no = new NoOntologia();
11         no.conceito = c2.getLocalName();
12         if (c2.listDisjointWith().toList().size() > 0) {
13           no.distintas = c2.listDisjointWith().toList();
14         }
15
16         for (Iterator i = c2.listSuperClasses(true); i.hasNext();) {
17           OntClass c = (OntClass) i.next();
18
19           if (c.isRestriction()) {
20             Restriction r = c.asRestriction();
21
22             if (r.isSomeValuesFromRestriction()) {
```

```
23                SomeValuesFromRestriction av = r.
                      asSomeValuesFromRestriction();
24
25                if (av.getOnProperty().getLocalName().contentEquals("
                      compostoPor")) {
26                  ArrayList al = new ArrayList();
27                  al.add(st.wordStemming(limparTexto.retirarPontuacao(c2.
                        getLocalName())));
28
29                  OntClass c22 = modelo.getOntClass(av.getSomeValuesFrom
                        ().getURI());
30
31                  for (Iterator iii = c22.listSubClasses(true); iii.
                        hasNext();) {
32                    OntClass oc2 = (OntClass) iii.next();
33                    al.add(st.wordStemming(limparTexto.retirarPontuacao(
                          oc2.getLocalName())));
34                  }
35                  no.termos = al;
36                }
37              } else if (r.isMinCardinalityRestriction()) {
38                MinCardinalityRestriction av = r.
                      asMinCardinalityRestriction();
39
40                if (av.getOnProperty().getLocalName().contentEquals("
                      minimoDe")) {
41                  no.minimoTermos = av.getMinCardinality();
42                }
43              }
44            }
45          }
46
47          if (no.termos != null) {
48            listaTermos.add(no);
49          }
50        }
51
52        this.trataTemUm(c2.getURI());
53        } while (ii.hasNext());
54      } else {
55        NoOntologia no = new NoOntologia();
56
57        ArrayList al = new ArrayList();
58        al.add(st.wordStemming(limparTexto.retirarPontuacao(cp.
              getLocalName())));
59
```

```
60        int qtde = 0;
61
62        for (Iterator i = cp.listSuperClasses(true); i.hasNext();) {
63
64        OntClass c = (OntClass) i.next();
65
66        if (c.isRestriction()) {
67          Restriction r = c.asRestriction();
68
69          if (r.isSomeValuesFromRestriction()) {
70            SomeValuesFromRestriction av = r.asSomeValuesFromRestriction
                ();
71
72            if (av.getOnProperty().getLocalName().contentEquals("
                compostoPor")) {
73              OntClass c2 = modelo.getOntClass(av.getSomeValuesFrom().
                  getURI());
74
75              for (Iterator iii = c2.listSubClasses(true); iii.hasNext()
                  ;) {
76                OntClass oc2 = (OntClass) iii.next();
77
78                al.add(st.wordStemming(limparTexto.retirarPontuacao(oc2.
                    getLocalName())));
79              }
80            }
81          } else if (r.isMinCardinalityRestriction()) {
82            MinCardinalityRestriction av = r.asMinCardinalityRestriction
                ();
83
84            if (av.getOnProperty().getLocalName().contentEquals("minimoDe
                ")) {
85              qtde = av.getMinCardinality();
86            }
87          }
88        }
89      }
90
91      no.conceito = cp.getLocalName();
92      if (cp.listDisjointWith().toList().size() > 0) {
93        no.distintas = cp.listDisjointWith().toList();
94      }
95
96      no.termos = al;
97      no.minimoTermos = qtde;
98
```

```
99      listaTermos.add(no);
100   }
101   return null;
102 }
```

B.3 Método utilizado para cálculo da similaridade

Listing B.4: *Método utilizado para o cálculo da similaridade*

```
1  public ArrayList processarTexto(String texto) {
2    /**
3    * Processa o texto.
4    */
5    String s = preProcessamento.tratarHistorico(texto);
6    /**
7    * Recupera os conceitos.
8    */
9    if (hm == null) {
10     hm = tratarOntologia.tratarOntologia();
11   }
12
13   Iterator it = hm.entrySet().iterator();
14
15   ArrayList listaConceitos = new ArrayList();
16
17   /**
18   * Encontra os termos que estão presentes na ontologia e no texto.
19   */
20   while (it.hasNext()) {
21     Entry e = (Entry) it.next();
22     ArrayList hm2 = (ArrayList) e.getValue();
23
24     Iterator it2 = hm2.iterator();
25
26     //Percorre todos os conceitos
27     while (it2.hasNext()) {
28       NoOntologia al = (NoOntologia) it2.next();
29
30       String[] val = s.split("␣");
31
32       double qtde = 0;
33       HashMap hs = new HashMap();
34       for (int i3 = 0; i3 < val.length; i3++) {
35         if (al.termos.contains(val[i3])) {
36           if (hs.containsKey(val[i3])) {
```

```
37              Integer inte = (Integer) hs.get(val[i3]);
38              hs.remove(val[i3]);
39              hs.put(val[i3], inte + 1);
40            } else {
41            hs.put(val[i3], 1);
42            qtde = qtde + 1;
43            }
44          }
45        }
46
47        double simi = qtde / (al.termos.size() <= val.length ? al.termos.
              size() : val.length);
48
49        if (simi > 0 && simi >= ((al.minimoTermos * 1) / al.termos.size()
              )) {
50          ConceitoPresente cp = new ConceitoPresente();
51
52          al.pai = tratarOntologia.buscaSuperClasse(al.conceito);
53
54          cp.noOntologia = al;
55          cp.qtde = qtde;
56          cp.termosPresente = hs;
57          cp.similaridade = simi;
58          listaConceitos.add(cp);
59        }
60      }
61    }
62
63    /**
64    * Verifica se existe classes distintas e utiliza a de maior
          similaridade
65    */
66    ArrayList<ConceitoPresente> listaFinal = new ArrayList<
          ConceitoPresente>();
67    for (int i = 0; i < listaConceitos.size(); i++) {
68      ConceitoPresente no = (ConceitoPresente) listaConceitos.get(i);
69
70      if (no.noOntologia.distintas != null) {
71        String pai = no.noOntologia.pai;
72        ArrayList<ConceitoPresente> aux = new ArrayList<ConceitoPresente
              >();
73
74        for (int i2 = listaConceitos.size() - 1; i2 > i; i2--) {
75          ConceitoPresente no2 = (ConceitoPresente) listaConceitos.get(i2
                );
76
```

```
77            if (no2.noOntologia.pai.equals(pai)) {
78              aux.add(no2);
79            }
80          }
81
82          if (aux.size() > 0) {
83            aux.add(no);
84            listaConceitos.removeAll(aux);
85
86            double maiorSimi = 0;
87            ConceitoPresente maiorNo = null;
88            for (int i2 = 0; i2 < aux.size(); i2++) {
89              if (aux.get(i2).similaridade >= maiorSimi) {
90                maiorSimi = aux.get(i2).similaridade;
91                maiorNo = aux.get(i2);
92              }
93             }
94
95             listaFinal.add(maiorNo);
96           }
97        } else {
98          listaFinal.add(no);
99        }
100     }
101
102     return listaConceitos;
103 }
```

B.4 Método utilizado para geração e filtragem das regras

Listing B.5: *Método utilizado para gerar e filtrar as regras de associação*

```
1  private void jButtonGerarRegrasActionPerformed(java.awt.event.
       ActionEvent evt) {
2
3    try {
4    //Busca os registros
5    Instances inst = ca.buscarInstances();
6
7    Associator associator = new Apriori();
8    StringBuffer outBuff = new StringBuffer();
9
10   numRegras = Integer.parseInt(txtNumeroRegras.getText());
```

```
11    minSuporte = Double.parseDouble(texto.Substituir(txtSuporte.getText()
          , ",", "."));
12    minConfianca = Double.parseDouble(texto.Substituir(txtConfianca.
          getText(), ",", "."));
13
14    minLeverage = Double.parseDouble(texto.Substituir(txtLeverage.getText
          (), ",", "."));
15    minLift = Double.parseDouble(texto.Substituir(txtLift.getText(), ",",
           "."));
16    minConviccao = Double.parseDouble(texto.Substituir(txtConviccao.
          getText(), ",", "."));
17
18    maxLeverage = Double.parseDouble(texto.Substituir(txtLeverage1.
          getText(), ",", "."));
19    maxLift = Double.parseDouble(texto.Substituir(txtLift1.getText(), ","
          , "."));
20    maxConviccao = Double.parseDouble(texto.Substituir(txtValorMaximo.
          getText(), ",", "."));
21
22    minConformidade = Double.parseDouble(texto.Substituir(
          txtMinConformidade.getText(), ",", "."));
23    minAnteInesperado = Double.parseDouble(texto.Substituir(
          txtMinAnteInesperado.getText(), ",", "."));
24    minConsInesperado = Double.parseDouble(texto.Substituir(
          txtMinConsInesperado.getText(), ",", "."));
25    minConsAnteInesperado = Double.parseDouble(texto.Substituir(
          txtMinAnteConsInesperado.getText(), ",", "."));
26
27    String[] options = new String[20];
28    int current = 0;
29
30    if (mostrarItemSet) {
31      options[current++] = "-I";
32    }
33
34    options[current++] = "-N";
35    options[current++] = "" + numRegras; // num rules
36    options[current++] = "-T";
37    options[current++] = "" + tipoMetrica; // tipo de métrica
38    options[current++] = "-C";
39    options[current++] = Utils.doubleToString(minConfianca, 2); //
          m_minMetric;
40    options[current++] = "-D";
41    options[current++] = "0.01"; // m_delta;
42    options[current++] = "-U";
43    options[current++] = "1.0"; // m_upperBoundMinSupport;
```

```
44    options[current++] = "-M";
45    options[current++] = Utils.doubleToString(minSuporte, 2); //
          m_lowerBoundMinSupport;
46    options[current++] = "-S";
47    options[current++] = "-1.0"; // m_significanceLevel;
48
49    while (current < options.length) {
50      options[current++] = "";
51    }
52
53    ((Apriori) associator).setOptions(options);
54
55    // Imprime algumas informações do Algoritmo
56    outBuff.append("=== Run information ===\n\n");
57    outBuff.append("\n");
58    outBuff.append("Relation:     " + inst.relationName() + '\n');
59    outBuff.append("Instances:    " + inst.numInstances() + '\n');
60    outBuff.append("Attributes:   " + inst.numAttributes() + '\n');
61
62    for (int i = 0; i < inst.numAttributes(); i++) {
63      outBuff.append("                " + inst.attribute(i).name() + '\n');
64    }
65
66    // Gera o modelo
67    associator.buildAssociations(inst);
68    outBuff.append("=== Associator model (full training set) ===\n\n");
69
70    outBuff.append("Minimum support: " + Utils.doubleToString(minSuporte,
          2) + " (" + ((int) (minSuporte * (double) inst.numInstances() +
          0.5)) + " instances)\n");
71    outBuff.append("Minimum metric <");
72    switch (tipoMetrica) {
73      case CONFIDENCE:
74        outBuff.append("confidence >: ");
75        break;
76      case LIFT:
77        outBuff.append("lift >: ");
78        break;
79      case LEVERAGE:
80        outBuff.append("leverage >: ");
81        break;
82      case CONVICTION:
83        outBuff.append("conviction >: ");
84        break;
85      }
86
```

```
87  outBuff.append(Utils.doubleToString(minConfianca, 2) + "\n\n");
88
89  FastVector[] fv = ((Apriori) associator).getAllTheRules();
90
91  String saida = "";
92  for (int i = 0; i < fv[0].size(); i++) {
93    double confianca = ((Double) fv[2].elementAt(i));
94    double lift = 0.0;
95    double leverage = 0.0;
96    double conviction = 0.0;
97
98    if (tipoMetrica != CONFIDENCE) {
99      lift = (Double) fv[3].elementAt(i);
100     leverage = (Double) fv[4].elementAt(i);
101     conviction = (Double) fv[5].elementAt(i);
102
103     if ((lift >= minLift && lift <= maxLift) &&
104       (leverage >= minLeverage && leverage <= maxLeverage) &&
105       (conviction >= minConviccao && conviction <= maxConviccao) &&
106       confianca >= minConfianca) {
107
108       //Inserir o calculo da medida subjetiva
109       Iterator iRegras = regras.iterator();
110       double valorConformidade = 0.0;
111       double valorAnteInex = 0.0;
112       double valorConsInex = 0.0;
113       double valorAnteConsInex = 0.0;
114
115       int[] itensAnte = ((AprioriItemSet) fv[0].elementAt(i)).items()
              ;
116       int[] itensCons = ((AprioriItemSet) fv[1].elementAt(i)).items()
              ;
117
118       double itemAnte = itensAnte.length;
119       double itemCons = itensCons.length;
120       double itemTotalRegra = itemAnte + itemCons;
121
122       while (iRegras.hasNext()) {
123         MedidaSubjetiva ms = (MedidaSubjetiva) iRegras.next();
124
125         double itemAnteInformado = ms.antecedente.size();
126         double itemConsInformado = ms.consequente.size();
127         double itemTotalInformado = itemAnteInformado +
              itemConsInformado;
128
129         double numIgualAnte = 0;
```

```
130             double numIgualCons = 0;
131             double numIgualTotal = 0;
132             for (int i2 = 0; i2 < inst.numAttributes(); i2++) {
133               if (itensAnte[i2] != -1) {
134                 if (ms.antecedente.containsKey(inst.attribute(i2).name())
                        ) {
135                   numIgualAnte++;
136                 }
137               }
138             }
139
140             for (int i2 = 0; i2 < inst.numAttributes(); i2++) {
141               if (itensCons[i2] != -1) {
142                 if (ms.consequente.containsKey(inst.attribute(i2).name())
                        ) {
143                   numIgualCons++;
144                 }
145               }
146             }
147
148             numIgualTotal = numIgualAnte + numIgualCons;
149             double nItens = (itemTotalInformado == 0 ? 1 : (numIgualTotal
                    / itemTotalInformado));
150
151             double a = 0.0;
152             double c = 0.0;
153             if (ms.tipoConhecimento.equals("Impressão␣Geral")) {
154               if ((numIgualAnte / itemAnte) > (numIgualCons / itemCons))
                      {
155                 a = Math.min((numIgualAnte / itemAnte), (numIgualTotal /
                        nItens));
156                 c = numIgualCons / itemCons;
157               } else {
158                 c = Math.min((numIgualCons / itemCons), (numIgualTotal /
                        nItens));
159                 a = numIgualAnte / itemAnte;
160               }
161             } else if (ms.tipoConhecimento.equals("Conhecimento␣Impreciso
                    ")) {
162               a = Math.min((numIgualAnte / itemAnte), (itemAnteInformado
                      == 0 ? 1 : (numIgualAnte / itemAnteInformado)));
163               c = Math.min((numIgualCons / itemCons), (itemConsInformado
                      == 0 ? 1 : (numIgualCons / itemConsInformado)));
164             }
165
166             if (ms.tipoRegra.equals("Conformidade")) {
```

```
167             valorConformidade += a * c;
168           } else if (ms.tipoRegra.equals("Antecedente␣Inesperado")) {
169             valorAnteInex += (a - c <= 0 ? 0 : a - c);
170           } else if (ms.tipoRegra.equals("Consequente␣Inesperado")) {
171             valorConsInex += (c - a <= 0 ? 0 : c - a);
172           } else if (ms.tipoRegra.equals("Antecedente␣Consequente␣
                Inesperado")) {
173             valorAnteConsInex += Math.max(valorConformidade, Math.max(
                  valorConsInex, valorAnteInex));
174           }
175         }
176
177         if ((valorConformidade >= minConformidade) &&
178             (valorAnteInex >= minAnteInesperado) &&
179             (valorConsInex >= minConsInesperado) &&
180             (valorAnteConsInex >= minConsAnteInesperado)) {
181
182             for (int i2 = 0; i2 < inst.numAttributes(); i2++) {
183               if (itensAnte[i2] != -1) {
184                 outBuff.append(inst.attribute(i2).name() + '=');
185                 outBuff.append(inst.attribute(i2).value(itensAnte[i2])
                      + '␣');
186               }
187             }
188             outBuff.append("=>");
189             for (int i2 = 0; i2 < inst.numAttributes(); i2++) {
190               if (itensCons[i2] != -1) {
191                 outBuff.append(inst.attribute(i2).name() + '=');
192                 outBuff.append(inst.attribute(i2).value(itensCons[i2])
                      + '␣');
193               }
194             }
195
196             outBuff.append("\n␣␣Medidas=");
197             outBuff.append("\n␣␣␣␣␣Confiança:␣" + Utils.doubleToString
                  (((Double) fv[2].elementAt(i)).doubleValue(), 2));
198             outBuff.append("\n␣␣␣␣␣Interesse:␣" + Utils.doubleToString
                  (((Double) fv[3].elementAt(i)).doubleValue(), 2));
199             outBuff.append("\n␣␣␣␣␣Novidade:␣" + Utils.doubleToString
                  (((Double) fv[4].elementAt(i)).doubleValue(), 2));
200             outBuff.append("\n␣␣␣␣␣Convicção:␣" + Utils.doubleToString
                  (((Double) fv[5].elementAt(i)).doubleValue(), 2));
201             outBuff.append("\n␣␣␣␣␣Conformidade:␣" + Utils.
                  doubleToString(valorConformidade, 2));
202             outBuff.append("\n␣␣␣␣␣Ante.␣Inesp.:␣" + Utils.
                  doubleToString(valorAnteInex, 2));
```

```
203                 outBuff.append("\n     Conse. Inesp.: " + Utils.
                        doubleToString(valorConsInex, 2));
204                 outBuff.append("\n     Ante. Conse. Inesp.: " + Utils.
                        doubleToString(valorAnteConsInex, 2));
205                 outBuff.append("\n");
206
207               }
208             }
209           }
210       }
211       campoArea1.setText(outBuff.toString());
212     } catch (Exception ex) {
213         ex.printStackTrace();
214     }
215 }
```