EDUARDO SIQUEIRA MARTINS

APLICAÇÃO DO PROCESSO DE DESCOBERTA DE CONHECIMENTO EM BASE DE DADOS A METADADOS TEXTUAIS DE INFRAESTRUTURAS DE DADOS ESPACIAIS

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

VIÇOSA
MINAS GERAIS – BRASIL
2012

Ficha catalográfica preparada pela Seção de Catalogação e Classificação da Biblioteca Central da UFV

T
M386a
2012

Martins, Eduardo Siqueira, 1981-
Aplicação do processo de descoberta de conhecimento em base de dados a metadados textuais de infraestruturas de dados espaciais / Eduardo Siqueira Martins. – Viçosa, MG, 2012.
xii, 78f. : il. (algumas col.) ; 29cm.

Orientador: Jurguta Lisboa Filho.
Dissertação (mestrado) - Universidade Federal de Viçosa.
Referências bibliográficas: f. 73-78

1. Mineração de dados (Computação). 2. Metadados. 3. Sistemas de informação geográfica. I. Universidade Federal de Viçosa. II. Título.

CDD 22. ed. 006.312

EDUARDO SIQUEIRA MARTINS

APLICAÇÃO DO PROCESSO DE DESCOBERTA DE CONHECIMENTO EM BASE DE DADOS A METADADOS TEXTUAIS DE INFRAESTRUTURAS DE DADOS ESPACIAIS

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 27 de fevereiro de 2012.

Fábio Ribeiro Cerqueira

Leacir Nogueira Bastos

Jugurta Lisboa Filho
(Orientador)

Toda honra e toda glória aos mestres dos mestres: Jesus.

A ti minha gratidão eterna.

AGRADECIMENTOS

A Deus, pela força, saúde e determinação para realizar este trabalho e, também, por colocar as pessoas certas em meu caminho, que sempre me auxiliaram, não só neste trabalho como em toda a vida.

A minha amada esposa Damila, pelo amor, pela paciência, pela dedicação, pela compreensão e pelo incentivo constante. A você também dedico esta conquista.

Ao meu filho Carlos Eduardo, benção de Deus, pela energia positiva transmitida nos seus sorrisos, pelas madrugadas de companhia. Filho papai te ama!

Aos meus pais, Joel e Maria José, por todo amor, apoio e solidariedade, durante esta jornada e por sempre acreditarem em mim.

Ao professor e orientador Jugurta Lisboa Filho, pelo companheirismo, pelo diálogo, pela credibilidade, pelos desafios e por ser exemplo de pessoa e profissional.

Ao professor Marcos Ribeiro, pelos conselhos, pelasricas contribuições na identificação, correção de falhas e análise dos resultados, durante o desenvolvimento deste trabalho.

A todos do corpo docente do Departamento de Informática da UFV e ao Altino Souza Filho, secretário da Pós-Graduação, pela pronta ajuda perante as burocracias do mestrado.

Aos colegas de estudo que me incentivaram e ajudaram durante o mestrado, em especial ao Odilon e Thiago Miranda. Ao Victor, aluno do Curso de Ciência da Computação da UFV e ao Yuri aluno do Curso de Sistemas de Informação do UnilesteMG, pela parceria e pelas contribuições no desenvolvimento desta pesquisa.

BIOGRAFIA

EDUARDO SIQUEIRA MARTINS, filho de Joel de Ávila Martins e Maria José Siqueira Martins, brasileiro, nascido em 08 de outubro de 1981, em João Monlevade, Minas Gerais.

No ano de 2001, após concluir o ensino médio no Colégio Kennedy de João Monlevade-MG, ingressou no curso de graduação em Sistemas de Informação, no Centro Universitário do Leste de Minas Gerais, concluindo-o no ano de 2005.

Trabalhou como programador e analista de sistemas no Centro Universitário doLeste de Minas Gerais de maio de 2003 a agosto de 2006. É professor da Instituição nos cursos de graduação desde agosto de 2006.

Em 2009, iniciou o mestrado em Ciência da Computação, na Universidade Federal de Viçosa (UFV), defendendo a dissertação em fevereiro de 2012.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE ABREVIATURAS

ANZLIC - Australia New Zealand Land Information Council

CONCAR - Comissão Nacional de Cartografia

FGDC - Federal Geographic Data Committee

IBGE - Instituto Brasileiro de Geografia e Estatística

IDE - Infraestruturas de Dados Espaciais

INDE - Infraestrutura Nacional de Dados Espaciais

INPE - Instituto Nacional de Pesquisas Espaciais

INSPIRE - Infrastructure for Spatial Information in Europe

ISO International Organization of Standards

MD – Mineração de Dados

MGB - Metadados Geoespaciais do Brasil

NSDI - National Spatial Data Infrastructure

KDD - Knowledge Discovery in Databases

OGC - Open Geospatial Consortium

SAIF - Spatial Archive and Interchange Format

SDI - Spatial Data Infrastructure

SGBDR - Sistema Gerenciador de Banco de Dados Relacional

SIG - Sistemas de Informação Geográfica

VGI - Voluntered Geographic Information

XML - eXtensible Markup Language

RESUMO

MARTINS, Eduardo Siqueira, M.Sc., Universidade Federal de Viçosa, fevereirode 2012. **Aplicação do processo de descoberta de conhecimento em base de dados a metadados textuais de infraestruturas de dados espaciais**. Orientador: Jugurta Lisboa Filho.

O constante avanço na área de tecnologia da informação tem viabilizado o armazenamento de grandes quantidades de dados em Infraestrutura de Dados Espaciais (IDE) devido baixo custo de dispositivos de armazenando, fácil acesso à Internet, existência de sistemas de informação e de ferramentas de gerenciamento. Com o aumento na quantidade de dados, surge a necessidade de novas pesquisas para extrair conhecimento de forma eficaz e inteligentenestas estruturas. Apesar das IDEs possibilitarem a cooperação e o compartilhamento de dados geoespaciais através de metadados, o caráter aberto e distribuído dessas infraestruturas é um fator que aumenta a dificuldade de recuperação desses dados. Este cenário apresenta muitos desafios de pesquisa em ciência da computação, tanto no nível físico (diversidade de estruturas de armazenamento) quanto conceitual (diversidade de perspectivas e de domínios de conhecimento). Uma alternativa que vem sendo aplicada para extração do conhecimento em estruturas de dados é a Mineração de Dados (MD), dada sua versatilidade. Pesquisas baseadas em MD em IDE e que utilizam metadados para descoberta de conhecimento foram analisadas. Entretanto, não foi encontrado trabalhos que abordam a mineração semântica dos metadados sobre IDE. Portanto, objetivo do trabalho é aplicar a MD sobre bases de metadados em IDE para extrair relações e conhecimento. Para isso, as etapas do processo de descoberta do conhecimento, técnicas de regras de associação, clusterização, mineração de texto e algoritmos de mineração foram utilizados. Para comprovar a viabilidade do método proposto, um estudo de caso foi utilizado para verificar a aplicação da mineração semântica neste tipo de base de dados.

ABSTRACT

MARTINS, Eduardo Siqueira, M.Sc., Universidade Federal de Viçosa, February, 2012. **Application of the process of knowledge discovery in database to textual metadata of spatial data infrastructures**. Adviser: Jugurta Lisboa Filho.

The constant advances in information technology have made possible the storage of large amounts of data in Spatial Data Infrastructure (SDI) because of low cost of devices for storing, easy Internet access, availability of information systems and management tools. With the increased amount of data comes the need for further research to extract knowledge effectively and intelligently in these structures. Despite SDIs provides cooperation and sharing of geospatial data through metadata, the open and distributed character of theses infrastructures is a factor that increases the difficulty of recovering such data. This scenario presents many challenges for research in computer science at both the physical level (variety of storage structures) and the conceptual level (diversity of perspectives and knowledge domains). An alternative that has been applied for the extraction of knowledge in data structures is the Data Mining (DM) due to its versatility. Researches based on DM in SDI and that use metadata for knowledge discovery were analyzed. However, no studies were found that address metadata semantic mining about SDI. Therefore, the aim of this study is to apply the MD about metadata bases in SDI to extract relationships and knowledge. For this reason, the steps of the process of knowledge discovery, techniques of association rules, clustering, text mining and mining algorithms were used. To prove the viability of the proposed method, a case study was used to verify the application of the mining semantic in this type of database.

1. INTRODUÇÃO

O crescente volume de dados tem gerado uma urgente necessidade de desenvolvimento de novas técnicas e ferramentas capazes de extrair e transformar, computacionalmente, dados em informações úteispara representar oconhecimento. Essas informações estão, na verdade, escondidas sob grandes quantidades de dados e não podem ser descobertas ou facilmente identificadas por sistemas convencionais de gerenciamento de banco de dados(SFERRA; JORGE CORREA, 2003).

A capacidade de armazenamento dos sistemas atuais ultrapassou a capacidade humana de analisar os dados. O desafio, portanto, é analisar de forma eficiente e automática esta massa de informações, extraindo conhecimento útil e novo (RAVIKUMAR;GNANABASKARAN,2010).

Bases de Dados de grandes empresas contêm uma potencial mina de ouro de informações. Porém, de acordo com Mitra*et al.* (2002), estes dados raramente são obtidos de forma direta. Um grande volume de informações está armazenado em bases de dados de forma dispersa e muitas vezes redundante.Esta grande disponibilidade dos dados aliada a necessidade de transformá-los em conhecimento significativo, têm demandado investimentos em pesquisas da comunidade científica e da indústria de software. (HAN; KAMBER, 2001).

Mineração de Dados (MD) é uma área que vem a cada dia sendo mais pesquisada, dada sua utilidade em diversas áreas de aplicação. A quantidade de dados a serem processados é cada vez maior, assim como sua complexidade. Como consequência, é cada vez mais difícil extrair conhecimento a partir desses dados. Dessa forma, o objetivo deste trabalho é utilizar a mineração de dados sobre uma coleção de metadados e utilizar algoritmos de mineração para extrair relações entre estes dados.

1.1. O problema e sua importância

Segundo Alves (2005), a sociedade atual, denominada sociedade da informação, vem sendo caracterizada pela valorização da informação, pelo uso cada vez maior de tecnologias de informação e comunicação e pelo crescimento exponencial dos

recursos informacionais disponibilizados em diversos ambientes, principalmente na Web.Apesar disso, ainda tem-se problemas relacionados a busca, localização, acesso e recuperação de informações.

O crescimento de dados espaciais, dos Sistemas de Informação Geográfica (SIG) e utilização generalizada de bancos de dados espaciais enfatizam a necessidade da descoberta do conhecimento nestas estruturas (RAVIKUMAR; GNANABASKARAN, 2010). Em sua evolução, os SIG tornaram-se parte integrante da infraestrutura de sistemas de informação de muitas organizações. Como consequência ocorreu um aumento significativo no número e volume das fontes de dados espaciais disponíveis. Assim, o compartilhamento e a democratização dessa grande base de dados tornaram-se cada vez mais importantes para uso e extração de conhecimentos (SILVA, 2009).

A comunidade geoespacial está se movendo em direção a bases de dados distribuídas e serviços Web, seguindo a evolução geral de tecnologia da informação e comunicação. Dessa forma, o compartilhamento de recursos e dados entre múltiplas comunidades de informação eleva a necessidade de novas tecnologias e a descoberta dos recursos de apoio e recuperação de informação.

Os usuários têm agora um acesso mais fácil aos dados geoespaciais através das Infraestruturas de Dados Espaciais (IDE)onde podem realizar: a coleta, a gestão, o acesso, a distribuição e a utilização eficaz dos dados espaciaiscomo, por exemplo, através de catálogos de dados geográficos (NEBERT, 2004). Porém, normalmente têm menos conhecimentos no domínio da informação geográfica. Isso às vezes leva à falha de tomada de decisões e leva a consequências negativas (DEVILLERS *et al.*, 2011).

Nesse contexto, surge a seguinte questão: como transformar dados em informações úteis para representar o conhecimento nesses grandes volumes de dados e estruturas ?

Várias pesquisas têm sido feitas no processamento de dados para se atingir a informação e representá-la em conhecimento. Pesquisas demonstram que os metadados dentro de IDEs são elementos fundamentais. Além de realizar a descoberta, avaliação, acesso e uso de funções, eles também podem suportar a interoperabilidade dos dados e gerar conhecimento (MANSO-CALLEJO *et al.*, 2011). Aditya e Kraak (2007) discutem e propõem o desenvolvimento evolutivo de interfaces de pesquisa das IDEs e mecanismos de mineração de dados em catálogos de metadados pela utilização de

mapas e gráficos que apoiem a descoberta de recursos geoespaciais. Neste mesmo contexto, Silva (2009) apresenta um mecanismo de busca semântica através do entendimento do contexto nas buscas a metadados. Também Azevedo (2005) define um modelo capaz de explorar as características espaciais do padrão XML (eXtensible Markup Language) e avaliar o impacto em sistemas de recuperação de dados e sua capacidade de tratar diferentes tipos de documentos XML.

Entretanto, todas estas pesquisas não utilizam o processo de mineração de dados para a extração de relações semânticas em uma coleção de metadados em uma IDE.Dessa forma, o trabalho aplica a MD sobre bases de metadados em IDE para extrair relações, padrões e conhecimento. Para isso, as etapas do processo de descoberta do conhecimento, técnicas e algoritmos de mineração foram utilizadas.

1.2. Objetivos

O objetivo geral desta pesquisa é aplicar algoritmos de mineração de dados sobre bases de metadados em Infraestruturas de Dados Espaciais, de forma a identificar padrões em dados geográficos.

Especificamente, pretende:

- Constituir uma base de metadados, com coleções de dados importados de IDEs;
- Aplicar diferentes algoritmos de mineração de dados sobre a base de metadados construída;
- Extrair relações e conhecimentoatravés dos metadados contidos em IDEs, utilizando algoritmos de mineração de dados;

1.3. Organização da dissertação

Esta dissertação está organizada da seguinte forma:

- Capítulo 1 - Apresenta o problema, sua importância e os objetivos do trabalho;
- Capítulos 2 e 3 - Descreve o referencial teórico, resultado da pesquisa bibliográfica sobre as áreas e tecnologias relacionadas ao trabalho;

- Capítulo 4 - Apresenta o processo de experimentação da mineração de dados na base de metadados do IBGE;
- Capítulo 5 - Descreve os resultados alcançados no estudo de caso;
- Capítulo 6 - Apresenta as conclusões e possíveis desdobramentos desta dissertação.

2. INFRAESTRUTURAS DE DADOS ESPACIAIS

Dados geoespaciais estão disponíveis em organizações públicas e privadas, no nível local, regional, nacional ou global. Para que haja sucesso na troca de dados geoespaciais entre essas organizações, IDEs são utilizadas. Neste contexto, pode ser definida como um conjunto de políticas, tecnologias, normas, padrões e recursos humanos necessários para coleta, gestão, acesso, distribuição e utilização eficaz dos dados espaciais (NEBERT *et al.* ,2004).

É uma estrutura técnica e organizacional adequada a economizar recursos, tempo e esforços na tentativa de adquirir novos conjuntos de dados, evitando ações redundantes e redução no custo de produção (RAJABIFARD; WILLIAMSON, 2001). Também oferece serviços de acesso à informação geográfica, de modo que , o usuário possa consultar um serviço para determinar se os dados que procuram estão disponíveis, consultar outro para avaliar detalhes sobre a fonte e, caso as características dos dados atendam à necessidade, acionar serviços para recuperá-los (DAVIS JR.; ALVES, 2005).

Dentre as iniciativas de IDE existentes, pode-se citar também o *Infrastructure for Spatial Information in Europe* (INSPIRE[1]), uma associação que visa promover a disponibilização, implementação e avaliação de informação da natureza espacial para a União Européia. O seu objetivo é estabelecer um enquadramento legal para a criação gradual e harmonizada de uma infraestrutura européia de informação geográfica (INSPIRE, 2010).

No Brasil, a Comissão Nacional de Cartografia (CONCAR[2]) tem como missão coordenar e orientar a elaboração e implementação da Política Cartográfica Nacional, visando assegurar um sistema cartográfico que garanta a atualidade e integridade da Infraestrutura Nacional de Dados Espaciais (INDE[3]), englobando tecnologias, políticas, normas e recursos humanos(CONCAR, 2011).

Segundo USFD (2000) a importância dasIDE para a administração e para o desenvolvimento econômico e social tem conduzido a maioria dos países do mundo a envolverem-se no processo de desenvolvimento de tais infraestruturas. Os países estão

[1]INSPIRE - http://www.ec-gis.org/inspire
[2] CONCAR - http://www.concar.ibge.gov.br.
[3]INDE - http://www.inde.gov.br.

cada vez mais aderindo a idéia de catálogos com fonte de dados pesquisáveis na Web. Os que já possuem afirmam claramente que as IDE são um meio estratégico para o desenvolvimento, tanto de países desenvolvidos como de países em desenvolvimento.

As áreas de pesquisas sobre IDE estão voltadas para semântica de dados e serviços geográficos, padrões de troca de arquivo, compartilhamento de informações em Banco de dados espaciais distribuídos, *Data Warehouse* Espacial,*Data mining* Espacial, acesso a informação espacial pelos cidadãos, granularidade do processamento de informação geográfica, organização e implementação de IDE, VGI (*Volunteered Geographic Information*) e*Digital Earth* (BERNARD; CRAGLIA, 2005), (DAVIS JR.; ALVES, 2005), (DAVIS JR *et al*., 2009).

2.1. Metadados

Metadados é definido genericamente como dados sobre dados. Alves (2005) define como parte do conteúdo da IDE, armazenam o conteúdo informacional de um recurso que pode estar em meio eletrônico ou não.São estruturados com o apoio da linguagem XML na codificação de documentos eletrônicos e utiliza o princípio de markup para colocar os dados em formato de fácil compreensão.

No contexto de IDE, a forma de representação da informação pelos metadados possibilita a construção de uma rede de conhecimentos e facilita a localização, interoperabilidade e a recuperação das informações eletrônicas. Neste sentido diversas bases de dados de diferentes ambientes computacionais podem ser integradas e compartilhadas, tais como as de governos, institutos, órgãos, associações, prefeituras e outros.

Por fazer parte do trabalho, na Seção 2.1.1 são apresentadas características dos metadados geográficos e padrões utilizados para representá-los.

2.1.1. Padrões de metadados geográficos

Segundo Prado*et al.* (2010) padrão de metadados geográficos é um conjunto de normatizações que permitem a descrição textual do dado geográfico de forma previamente estabelecida. Os padrões de metadados surgiram com o desenvolvimento dos mapas digitais, produzidos pela cartografia digital, e pela manipulação da

informação geográfica através dos SIG (SILVA,2007).Segundo Concar (2011),são essenciais para:

- promover a sua documentação, integração e disponibilização (de dadosgeoespaciais), bem como possibilitar sua busca e exploração;
- evitar duplicidade de ações e o desperdício de recursos (na produção e divulgação de dados geoespaciais);
- o compartilhamento e disseminação (de dados), sendo obrigatório para órgãos e entidades do poder executivo federal e voluntário para os demais;
- órgãos e entidadesna produção, direta ou indireta, ou na aquisição de dados, obedecer aos padrões (de dados e de metadados) estabelecidos para a INDE.

Para alcançar a interoperabilidade dos dados entre IDEs, metadados foram implementados utilizando padrões para oferecer replicação as características, notações e formalismos, além de facilitar a disseminação de normas e regras que melhoram a comunicação entre usuários e sistemas. Dos vários padrões que podem representar os metadados geográficos existentes, Prado *et al*. (2010) destaca o padrão americano do *United States Federal Geographic Data Comitte* (FGDC), o padrão da *International Organization of Standards* (ISO) e o padrão *Dublin Core*, proposto pela *Dublin Core Metadata Initiative*. No Brasil, o órgão regulador de metadados geográficos é a Comissão Nacional de Cartografia (CONCAR). O padrão de metadados geoespaciais do Brasil (MGB), homologado em 2009, é um subconjunto dos metadados estendidos e previstos na norma internacional ISO19115:2003 e que descreve os ambientes de aquisição, edição e divulgação dos dados.

O padrão ISO19115 é caracterizado por suportar qualquer tipo de informação geográfica e abranger vários outros padrões.Por isso, possui mais de 350 elementos para representá-lo. O padrão do FGDC também é caracterizado por suportar um grande número de tipos de informação geográfica.Ele possui 334 elementos distintos, dentre os quais 119 são compostos, ou seja, existem apenas para agrupar outros. O padrão Dublin Coreé considerado o mais utilizado e difundido em diversas áreas do conhecimento,

devido a não limitar-se a dados geográficos e ter apenas 15 elementos para representá-lo.

Embora seja extensoe complexo, o padrão ISO19115 prevê o fato de que na maioria das situações o preenchimento de todos os elementos não é necessário. Apenas 8elementos são obrigatórios, sendo consideradosa base mínima ou o núcleo padrão a todos perfis de metadados (CONCAR, 2011).

Na Seção4.1 são descritos os detalhes de uso dos tipos de padrões utilizado no trabalho.

3. MINERAÇÃO DE DADOS

De forma clássica, pode-se definirMD como um processo não-trivial de identificar, em dados, padrões válidos, novos, potencialmente úteis e compreensíveis (FAYYAD *et al.*, 1996). Han e Kamber (2001)definema MD como processo de descoberta do conhecimento em grandes quantidades de dados armazenados em bases de dados, *datawarehouses* ou outros repositórios de informações. ParaElmasri *et al.* (2004), MD se refere à mineração ou descoberta de novas informações.

A MD envolve a integração de várias áreas, como, por exemplo, banco de dados, estatística, aprendizagem de máquina, computação, reconhecimento de padrões, redes neurais, visualização de dados, recuperação de imagem, análise de dados espaciais, dentre outras (HAN; KAMBER, 2001). Por isso, é considerada uma das áreas interdisciplinares mais promissoras em desenvolvimento na tecnologia de informação.

A MD pode ser aplicada de duas formas: como um processo de verificação e como um processo de descoberta. No processo de verificação o sistema é limitado a verificar as hipóteses formuladas pelo usuário. No processo de descoberta, o sistema autonomamente encontra novos padrões (FAYYAD *et al.* ,1996).

De acordo com os objetivos da aplicação e da natureza dos dados, uma determinada tarefa é utilizada. Na mineração existem dois tipos de tarefas básicas: descritiva (ou não-supervisionada) e preditiva (ou supervisionada). As descritivas se concentram em encontrar padrões que descrevam os dados de forma interpretável pelos seres humanos, como, por exemplo, identificar dentro de um conjunto de dados gruposde pessoas que possuem características parecidas. As preditivas realizam interferência nos dados para construir modelos que serão usados para predições do comportamento de novos dados, como, por exemplo, descobrir a previsão do tempo através de dados coletados por informações de satélite para predizer se em uma determinada região vai chover ou não (FAYYAD *et al.* ,1996), (HAN; KAMBER, 2001).

Existem várias tarefas de MD que seguem os princípios das tarefas básicas, algumas mais conhecidas e utilizadas são: associação, classificação e clusterização (agrupamento) (HAN; KAMBER, 2001).Estas tarefas estão detalhadas na Seção 3.2.

3.1. O Processo de descoberta de conhecimento em bases de dados (KDD)

O Processo de Descoberta de Conhecimento em Base de Dados, mais conhecido por sua sigla em inglês KDD (*Knowledge Discovery in Databases*) (FRAWLEY *et al.*, 1992) consiste, basicamente, em encontrar conhecimento útil que esteja encoberto em enormes quantidades de dados.

Fayyad*et al.* (1996) definem KDD como processo, não trivial, de extração de informações implícitas, previamente desconhecidas e potencialmente úteis, a partir dos dados armazenados em um banco de dados. Em outras palavras, ele diz que este processo procura novas informações para o sistema (e possivelmente para o usuário) que tragam algum benefício para o usuário.

Segundo Han e Kamber (2006), o processo de KDD pode ser visto como uma evolução natural da tecnologia de informação. Hoje é possível buscar e armazenar grandes quantidades de dados. Porém, essa quantidade de dados gerados superou a capacidadehumana de compreensão dos mesmos, o que criou a necessidade de ferramentas para realizar esta análise.

Praticamente todas as áreas de conhecimento podem usar as técnicas de MD. Frawley(1992) cita algumas destas áreas:

- **Marketing –**análise comportamento do cliente; identificar diferentes grupos de clientes ;
- **Medicina** -sintomas de doenças, análise de experimentos, efeitos colaterais de medicamentos;
- **Detecção de fraude–**monitoramento de crédito, chamadas clonadas de telefones celulares, identificar transações fraudulentas;
- **Agricultura–**tendências e classificação de pragas em legumes e frutas;
- **Área Social** - pesquisa de intenção de votos, resultados de eleições;
- **Militar–** análise de informações; perfil de usuários de drogas;
- **Ciência Espacial -** astronomia, análise de dados espaciais;

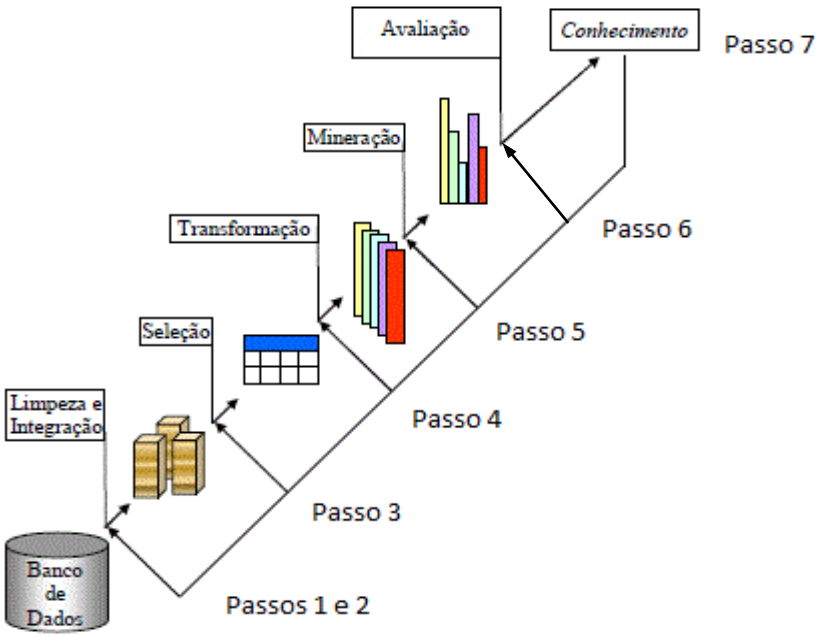
Segundo Curotto (2003) vários autores utilizam o termo Mineração de Dados em um sentido mais amplo, representando todo o processo de KDD, ao invés de apenas uma etapa deste processo. Assim, é preciso entender todas as etapas do processo de

descoberta do conhecimento e saber como aplicá-las para obter os melhores resultados possíveis.

O objetivo do processo de descoberta é fazer com que dados brutos virem conhecimento útil.Segundo Han e Kamber (2001), os passos para obter este conhecimento consistem em uma sequência iterativa (Figura 3.1) e podem ser vistos da seguinte forma:

1. Limpeza dos dados – identificação e tratamento dos dados inconsistentes, incompletos ou irrelevantes;
2. Integração dos dados – união de várias fontes de dados em uma única;
3. Seleção dos dados – seleção dos dados considerados relevantes para a análise;
4. Transformação dos dados – transformação dos dados em formatos apropriados para o processo de mineração de dados;
5. Mineração de dados – aplicação dastarefas, métodos e algoritmospara extrair padrões em um conjunto de dados;
6. Avaliação dos padrões –análise da utilidade dos padrões encontrados.
7. Apresentação e representação do conhecimento - uso de técnicas de visualização e representação do conhecimento para apresentar ao usuário o conhecimento obtido.

Os passos de 1 a 4 são chamados de etapas de pré-processamento dos dados, onde ocorre a preparação dos dados para a mineração. Opasso 5 etapa de mineração dos dados e é onde são descobertos os padrões - o conhecimento – de fato. Os passos 6 e 7 são chamados de etapas de pós-processamento, onde há a avaliação se o conhecimento descoberto é realmente novo ou relevante e se está apto para a apresentação ao usuário.Dessa forma, tem-se, então, que o únicopasso que realmente faz a descoberta de conhecimento é o 5, o que pode causar a impressão de que todos os outros são meramente suplementares a ela. Por este motivo muitas vezes o processo de KDD é visto como um sinônimo de Mineração de Dados.



Fonte: Adaptada de Han e Kamber (2001).

Figura 3-1 – Etapas do processo de KDD

Apresentado os passos para o processo de KDD a Seção 3.2 demonstra as tarefas utilizadas no trabalho.

3.2. Tarefas de mineração de dados

As tarefas são recursos utilizados na MD imprescindíveis para descoberta de padrões e conhecimento.Segundo Han Kamber (2001) não há uma tarefa que atenda a todos os requisitos, por isso,é importante conhecer suas características para direcionar os dados a serem minerados para uma tarefa específica ou para um conjunto de tarefas. Dessa forma, a aplicaçãodos dados em uma tarefa de maneira errada contribui para descobertas irrelevantes no processo de extração de conhecimento.

As tarefas de MD podem ser dividias em três categorias, conforme listadas por Han e Kamber (2001) e resumidas abaixo:

- **Regras de Associação** –esta tarefa faz a associação das transações e identifica as regras conforme suas ocorrências. Determina os itens que sempre ocorrem em conjunto ou com certa frequência. Como, por exemplo, as seguintes regras: “todos os clientes que compram os

produtos A e B compram o produto C” e “80% dos clientes que compram o produto A também compram o produto B”. Assim, as regras identificam associações que ajudam na tomada de decisão.Às vezes, apenas com mudança da disposição de um item na prateleira de um comércio, pode trazer melhoras nas vendas e satisfação dos clientes. Por ser uma parte importante deste trabalho, esta tarefa está detalhada na Seção 3.3.

- **Classificação e Predição–**a classificação utiliza o modelo de dados para estabelecerclasses de objetos que ainda não foram classificadas.Como, por exemplo, classificar o perfil do cliente através da faixa salarial,como bronze, prata ou ouro. Outro exemplo seria utilizar uma base de dados de veículos, em que cada registro contém os atributos de opcionais, preço, número de portas, cilindrada, tipo de câmbio e assim, classificar cada veículo em popular ou de luxo. Já a predição está associadaa prever valores desconhecidos ou futuros. Como, por exemplo, prever qual será o valor do combustível daqui um determinado período de tempo; o número de pessoas que sairá da classe D para classe B em dez anos; se o valor do dólar subirá ou cairá no dia seguinte, entre outros.
- **Clusterização (Agrupamento) -** Diferentemente da classificação e predição, a clusterização trabalha sobre dados onde tais classes não estão pré-definidas. A tarefa consiste em formar grupos de objetos que sejam semelhantes entre si. Nessa tarefa cabe ao especialista avaliar e classificar os resultados dos grupos. Alguns exemplos são: agrupar sintomas que identificam uma ou mais doenças; os clientes que possuem perfis parecidos e diferentes, entre outros. Por ser uma parte importante deste trabalho, esta tarefa está detalhada na Seção 3.4.

3.3. Regras de associação

Em muitas aplicações, há a necessidade de verificar-se quão frequentemente dois ou mais objetos se correlacionam. O caso mais frequente de aplicação é na análise

de cesta de produtos (*basket case analysis*) (HAN; KAMBER; 2001), onde, por exemplo, pode-se descobrir quais produtos são comprados juntos mais frequentemente.

Para extrair os conjuntos de itens frequentes e encontrar as regras de associação, as medidas de suportee confiança são fundamentais. Segundo Zhang e Zhang (2002) estas medidas são mais utilizadas em regras de associação. Elas vão determinar diretamente tanto a quantidade como a qualidade das regras geradas.

Segundo Goldschmidt e Passos (2005), o *suporte (sup)* elimina as associações consideradas irrelevantes e tem por objetivo indicar o percentualde ocorrências da associação em relação ao número total de transações.Já a*confiança (conf)*indica a força da implicação ($A \rightarrow B$), isto é, em pelo menos X% das vezes que o antecedente (A) ocorrer nas transações, o consequente (B) também deve ocorrer.

Para Cheng *et al.* (2008), o grande número de regras geradas facilmente, torna a análise das regras complexa e muitas vezes restringe o seu uso na prática. Sendo assim, para minimizar este número, constitui-se um limite para filtragem de associações descobertas. Com este intuito é determinado umvalor de suporte e confiança mínimapara encontrar somente as regras que estejam dentro dosvalorespreestabelecidos. Normalmente a maioria dos algoritmos de regras de associação utilizaestes valores como parâmetros de entrada.

Para maior compreensão do uso do suporte e confiança é apresentado um exemplo que demonstra o cálculo destas duas medidas para extração de regras de associação. A tabela 3.1 lista os itens por transação comprados em uma padaria.

Tabela 3.1 – Relação de itens comprados por transação

Transações	Itens comprados
1	Ovo, pão
2	Ovo, sorvete, manteiga
3	Detergente, pão
4	Ovo, manteiga, pão

Considerando o valor preestabelecido do suporte mínimo = 50%(2 transações) e a confiança mínima = 50%, pode-se obter as regras de associação somente que enquadram nestes parâmetros de medidas.

Os itens (ovo, pão) e (ovo, manteiga) da tabela 3.1 foram os que se enquadraram nos valores preestabelecidos. As regras de associação geradas eo cálculo do suporte e da confiança com estes itens estão demonstrado na tabela 3.2.

Para calcular o suporte dos itens, o cálculo é feito através do número de transações (*n*) que contém X e Y dividido pelo número total de transações (T) e pode ser representado pela fórmula $\frac{n}{T} \times 100$. Já a confiança é calculada pelo número de transações (*n*) que contém X e Y dividido pelo número de transações que contém X (m) e pode ser representada pela fórmula $\frac{n}{m} \times 100$.Também pode ser calculada através do suporte de X e Y dividido pelo suporte de X.

Tabela 3.2 – Regras de Associação Geradas, cálculo de suporte e confiança

Regras geradas	Suporte(%)	Confiança(%)
ovo → pão	$\frac{2}{4}$=0,5 (50%)	$\frac{2}{3}$=0,66 (66%)
pão → ovo	$\frac{2}{4}$=0,5 (50%)	$\frac{2}{3}$=0,66 (66%)
ovo → manteiga	$\frac{2}{4}$=0,5 (50%)	$\frac{2}{3}$=0,66 (66%)
manteiga → ovo	$\frac{2}{4}$=0,5 (50%)	$\frac{2}{2}$=1 (100%)

Pelas regrasgeradas na tabela 3.2 é possível comprovar que todos osvalores de suporte e confiançasão iguais ou superiores ao percentual mínimo preestabelecido (50%). Assim as associações que não atingem o preestabelecido são descartadas.

Importante dizer que as regras geradas possuem indicadores valiosos que ajudam no processo de tomada de decisão (HAN; KAMBER, 2006).Analisando o valor de confiança da regra manteiga → ovo, por exemplo, percebe-se que quando o cliente compra manteiga também compra ovo em 100% das vezes. Já a regra ovo → pãoindica que o cliente que compra ovo tem 66% de possibilidade de também comprar pão.Pelo exemplo, o gerente poderia tomar uma decisão imediata de colocar ovo e manteiga próximos para facilitar a compra do cliente devido ao índice apresentado e fazer que a segunda regra seja parâmetro de análise para futura tomada de decisão.

Para Melanda (2004), há a necessidade de considerar outras medidas para a avaliação das regras de associação. Os algoritmos de associação utilizam métricas tais

como: *confidence, lift, leverage, conviction*. Estas métricas ajudam na filtragem das regras podendo ser outros parâmetros útéis. Neste trabalho além do suporte e a confiança, a medida *lift*foi utilizadapara avaliar o quanto dois ou mais conjuntos de itens são independentes entre si e assim descobrir o grau de relação entre eles. Na seção 4.4.2 esta métrica está detalhada.

Após a apresentação das medidas de regras de associação, a seção 3.3.1 detalhao algoritmo de regras de associação utilizado no trabalho.

3.3.1. Algoritmo Apriori

Desenvolvido por Agrawal e Srikant (1994), o algoritmo *Apriori* está entre os 10 algoritmos de MD mais utilizados na comunidade acadêmica (WU *et al.*,2008).Satisfaz o suporte mínimo definido pelo usuário e é o mais utilizado para geração dos conjuntos de itens frequentes (LI ,2008).

O algoritmo se baseia no seguinte princípio:sejam X e Y dois conjuntos de itens quaisquer. Se X for frequente, então qualquer subconjunto Y que estiver contido em (ou for igual a) X também é frequente. Se X não for frequente, então qualquer subconjunto Y que contiver (ou for igual a) X também não é frequente. Segundo Pitoni(2002) esta regra é utilizada para reduzir o número de candidatos a serem comparados com cada transação no banco de dados. A Figura 3.2 demonstra o pseudocódigo do algoritmo.

Passo de Junção: C_k é gerado combinando L_{k-1} consigo

Passo de Poda: Qualquer (*k-1*)-*Conjunto de Itens* que não é freqüente não pode ser um subconjunto de um *k-Conjunto de Itens* freqüente

Pseudo-Código:

C_k: *Conjunto de Itens* candidatos de tamanho *k*

L_k: *Conjunto de Itens* freqüentes de tamanho *k*

L_1 = {itens freqüentes};

para (k = 1; L_k != Ø; k++) ***faça***

C_{k+1} = candidatos gerados de L_k;

para cada transação *t* na base de dados ***faça***

atualiza o contador de todos os candidatos em C_{k+1} que estão contidos em *t*

L_{k+1} = candidatos em C_{k+1} com *min_support*

end

return $\cup_k L_k$;

Figura 3-2 – Pseudocódigo do algoritmo *APriori*

O algoritmoé dividido em duas etapas. Na primeira, é feito o cálculo do suporte para cada item separadamente (conjunto-de-1-item) e depois selecionado os itens que satisfazem o suporte mínimo preestabelecido, formando os conjuntos-de-1-item frequentes. Na segunda, ocorre a geração dos conjuntos-de-2-itens pela junção dos conjuntos-de-1-item frequentes encontrados na etapa anterior. É feito o cálculo do suporte e comparado ao suporte mínimo e assim é formado o conjunto-de-2-itens frequentes. O algoritmo continua esse processo, até que o conjunto-de-k-itens frequentesseja um conjunto vazio (PITONI, 2002).

Para ilustrar as etapas do algoritmo, tomamos como referência a tabela 3.1. Para contemplar a primeira etapa do algoritmo,as tabelas 3.3 e 3.4 demonstram respectivamente, o cálculo do suporte de cada item (C1) e a geração do conjunto-de-1-item frequentes (L1), este último determinado pelo suporte mínimo de50%.

Tabela 3.3 – Cálculo de suporte para conjunto-de-1-item (C1)

ItemID	Itens	Suporte
1	detergente	1/4 = 0,25 (25%)
2	manteiga	2/4 = 0,5 (50%)
3	ovo	3/4 = 0,75 (75%)
4	pão	3/4 = 0,75 (75%)
5	sorvete	1/4 = 0,25 (25%)

Tabela 3.4 – Geração do conjunto-de-1-item frequentes (L1)

ItemID	Itens	Suporte
1	manteiga	2/4 = 0,5 (50%)
2	ovo	3/4 = 0,75 (75%)
3	pão	3/4 = 0,75 (75%)

Na segunda etapa, para descobrir o conjunto-de-2-itens (C2), o algoritmo usa somente os itens em (L1) para gerar o (C2). A tabela 3.5 demonstra o (C2) e a tabela 3.6 o conjunto de itens frequentes gerados (L2).

Tabela 3.5 - Cálculo de suporte para conjunto-de-2-itens (C2)

ItemID	Itens	Suporte
1,2	manteiga, ovo	2/4 = 0,5 (50%)
1,3	manteiga, pão	1/4 = 0,25 (25%)
2,3	ovo, pão	2/4 = 0,5 (50%)

Tabela 3.6 - Geração do conjunto-de-2-itens frequentes (L2)

ItemID	Itens	Suporte
1,2	manteiga, ovo	2/4 = 0,5 (50%)
2,3	ovo, pão	2/4 = 0,5 (50%)

Para descobrir o conjunto-de-3-itens, segue os mesmos passos já apresentados. Como o (C3) não atingiu o suporte mínimo (tabela 3.7), a (L3) é vazia e então o algoritmo termina tendo encontrado todos os itens frequentes.

Tabela 3.7 - Cálculo de suporte para conjunto-de-3-item (C3)

ItemID	Itens	Suporte
1,2,3	manteiga, ovo, pão	1/4 = 0,25 (25%)

Apesar de não apresentar no exemplo, os parâmetrosde confiança e tipo de métrica foram considerados no trabalho para melhorar a qualidade das regras geradas pelo algoritmo*APriori*.Na seção 4.4.2 está detalhado o uso destes parâmetros.

Embora seja um dos algoritmos mais utilizados para regras de associação, o algoritmo APriori não apresenta boa eficiência na mineração de grandes volume de dados e quando preestabelecido o valor de suporte mínimo baixo. Este fato acontece devido a geração de um número muito elevado de conjunto de itens frequentes a cada iteração. Sendo assim, o processamento se torna custoso computacionalmente e, muitas vezes, inviável (HAN; KAMBER, 2006).

Inicialmente, os pontos negativos apresentados não inviabilizaram o uso do algoritmo no trabalho. Esta justificativa deve-se pelo tamanho da base e os parâmetros

de confiança e tipo de métricas utilizados para minimizar a quantidade de regras geradas.

Apresentado os conceitos e características utilizadas na análise de regras de associação, na seção 3.4 é apresentado o método de clusterização (agrupamento) de dados.

3.4. Clusterização de dados

O termo "*cluster*ização" também significa"agrupamento".Segundo Carlantonio (2001), clusterizar é fazer com que os elementos que compõem cada grupo (*cluster*) sejam mais parecidos entre si do que parecidos com os elementos dos outros grupos. É colocar os semelhantes juntos num mesmo grupo e os desiguais em grupos distintos.

Han e Kamber (2001) dizem que,ao contrário de classificação e predição, que analisam classes rotuladas, ou seja, elementos categorizados, na clusterizaçãonão há classes pré-definidas, os elementos são agrupados de acordo com a semelhança.

Com os elementos semelhantes em cada grupo, é possível analisar e identificar as características de cada um deles e, desta forma, criar um nome que represente esse grupo, podendo assim virar uma classe.

Segundo Han e Kamber (2001), os elementos a serem agrupados podem estar descritos pelas seguintes variáveis:

- **escaladas em intervalos** - podem assumir intervalos diferentes dependendo da unidade de medida tais como peso, distância e altura que podem ser representados por diferentes unidades;
- **binárias** - utilizadas quando se quer demonstrar a ausência ou a presença da característica da variável, sendo representada respectivamente com 0 ou 1;
- **variáveis discretas –** podem ser divididas em ordinais ou nominais. Ordinais seguem uma ordem, como, porexemplo, dia da semana e dia do mês, já as nominais não seguem uma ordem, como, por exemplo, estado civil, cargo e cor do cabelo;
- ou ainda combinações desses tipos de variáveis.

3.4.1. Etapas do processo de clusterização

Jain *et al*. (1999) aponta quatro etapas para assegurar a eficiência da clusterização: (a) seleção das variáveis, (b) definição da medida de similaridade, (c) seleção do algoritmo de clusterização e (d) avaliação do resultado. Adicionalmente, Xu e Wunsch (2005) destacam a etapa (e), interpretação dos resultados como outra etapa importante ao processo de extração do conhecimento em clusterização, totalizando cinco etapas.

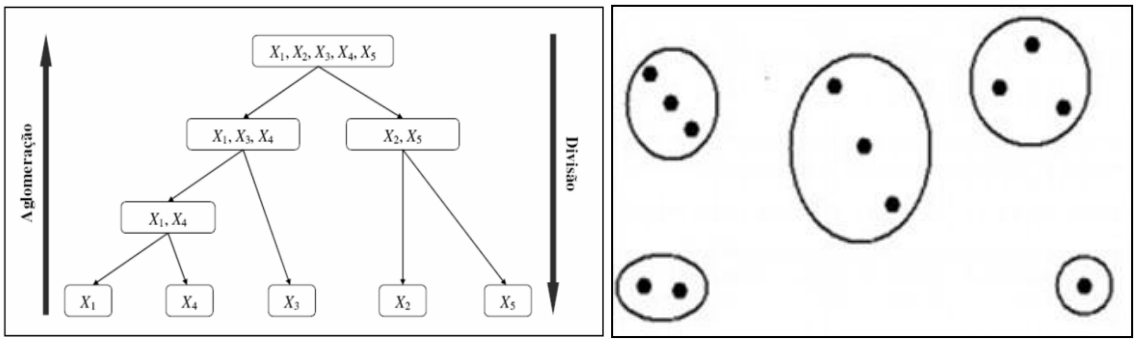
Na etapa de seleção das variáveis são identificadas as variáveis mais relevantes do conjunto de dados inicial. Em seguida, é feita a escolha dos formatosa serem utilizados: quantitativo ou qualitativo, contínuo ou binário, nominal ou ordinal. Após, é avaliada a necessidade de realizar a normalização entre o(s) formato(s) escolhido(s), ou seja, fazer com que todas as variáveis estejam dentro do mesmo padrão ou intervalo. Segundo Han e Kamber (2006), isso evita que as unidades de medida influenciem o agrupamento dos objetos, dando pesos diferentes às variáveis. Por fim, é criada a matriz de dados, responsável por mapear as variáveis, ondeas linhas representam o conteúdo das variáveis e a coluna o seu nome.

Amedida de similaridade tem como objetivomensurar o quanto dois objetos são semelhantes/distintos entre si (XU; WUNSCH, 2005). Na etapa de escolha dessa medida, éanalisado o formato da matriz da etapa anteriore definida a medida para o cálculo da similaridade/dissimilaridade entre osobjetos. Existem várias medidas para realizar este cálculo, como, por exemplo, *Manhattan,Minkowski, Pearson,*dentre outras, conforme apresentado por Xu e Wunsch (2005), mas, a Euclidiana é a mais utilizada. Neste método, calcula-se a distância em linha direta entre os dois pontos que representam os elementos e tende-se, neste caso, a encontrar grupos de objetos no formato esférico ou de árvore conforme o método de agrupamento escolhido(COLE,1998),(HAN ; KAMBER,2006).

Na etapa de seleção do algoritmo de clusterização primeiramente é avaliado o método de agrupamento que melhor represente a clusterização dos dados. Existem vários métodos, mas segundo Jain *et al.* (1999), os principais são: hierárquico(o conjunto de *clusters* está organizado e representado como uma árvore de forma aglomerativa ou divisiva - Figura 3.2a) e particionamento (divide os objetos em K

clusters distintos seguindo uma definição de similaridade entre eles – Figura 3.2b). Em seguida é selecionado o algoritmo que represente melhor as características do método escolhido e o objetivo de agrupamento pretendido.

Existem vários tipos de algoritmos, uns apropriados para grandes quantidades de dados e outros para pequenas quantidades; algoritmos em que o número de *clusters* tem que ser fornecido pelo usuário e outros em que não há essa exigência; dentre outras características (AGRAWAL *et al.*,1998). Por isso, a escolha do algoritmo deve ser feita observando bem suas características. Não existe um algoritmo que atenda a todos os requisitos.Dessa forma, às vezes é necessário utilizar até mais de um algoritmo para chegar ao resultado pretendido.O algoritmo escolhido no trabalho e o motivo de sua escolha estão detalhados *na* seção3.4.2.



(A) (B)

Figura 3-3 - Exemplo de representação do método hierárquico (A) e do método por particionamento (B)

Na etapa de avaliação dos resultados é avaliada a qualidade dos dados agrupados em cada cluster. A validação pode ser com base em índices estatísticos ou através da comparação com outros algoritmos. Mello (2008) destaca a importância de avaliar os resultados com diferentes parâmetros de entrada (número de clusters, tipos de métricas, dentre outros), pois a ordem e a forma de apresentação dos dados podem sofrer mudanças. Além disso, a análise dos resultados pode levar à redefinição dos atributos escolhidos e/ou da medida de similaridade, definidos nas etapas anteriores. (CARLANTONIO, 2001).

Por fim, na etapa de interpretação dos resultadossãodescobertos os possíveis rótulos ou categorização de cada grupo.Em muitas circunstâncias, uma série de testes e repetições para certeza da interpretação são necessárias. Identificar estes rótulos não é

uma tarefa trivial.O auxílio de um especialista no domínio do problema é essencial para garantir a confiabilidade do conhecimento extraído. (XU; WUNSCH, 2005).

Com as etapas do processo de clusterização apresentada, na seção 3.4.2 é detalhado o algoritmo utilizado no trabalho.

3.4.2. Algoritmo *K-means*

O algoritmo de particionamento *K-means*, também conhecido como K-médias, tem como princípio particionar em K clusters um conjunto de dados, onde K foi previamente definido (OLIVEIRA, 2008). Dessa maneira, cada *cluster* é representado por um centroide que é a base para o agrupamento dos dados. Este agrupamento é construído de modo que a similaridade dentro do *cluster* seja alta e entre *clusters* seja baixa.

K-means, é um dos algoritmos mais conhecidos e amplamente utilizado na área de clusterização. Trabalhos de Oliveira (2008) e Han e Kamber (2001) apresentam boa funcionalidade com K-means quando os clusters são compactos e separados uns dos outros. Também é relativamente escalável e eficiente para o processamento de grandes estruturas de dados e seu resultado pode ser utilizado como entrada para outras tarefas como, por exemplo, a tarefa de classificação e predição.

Para que o algoritmo consiga ajudar na extração do conhecimento é preciso ficar atento ao conjunto de dados a ser utilizado. O *K-Means* só pode ser aplicado a dados onde a média pode ser calculada. Além disto, a presença de ruídos ou valores extremos nos dados influencia na média, podendo gerar uma associação dos objetos ao centroide dos grupos de maneira errônea e assim prejudicar todo o processo de agrupamento dos dados (MELLO, 2008).

Segundo Fontana e Naldi (2011) os quatro passos para a construção do algoritmo são:

Passo1: sorteio inicial dos centroides conforme número de *clusters* definido como parâmetro pelo usuário.

Passo2: associa os objetos ao centroide mais próximo.

Passo3: recalcula o valor do centroide de cada *cluster* a partir da média dos objetos pertencentes ele.

Passo4:desvincula os objetos dos *clusters*. Os passos 2 e 3 são repetidos até que não haja mais mudança nos centroides, que ocorram poucas mudanças nos centroides ou que o número de iterações definida pelo usuário seja alcançada. Tipicamente, o critério do erro quadrático é usado para ajudar a identificar o melhor agrupamento dos objetos dentro dos centroides.

Segundo Han e Kamber (2001) a fórmula para calcular a soma dos erros quadráticos é dada por:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} \left| p - m_i \right|^2 ,$$

onde E é a soma dos erros quadráticos, m_i é a média ou centroide do grupo C_i e p é o vetor que representa um objeto no espaço. Dessa forma, o método pára de realocar os objetos quando encontra um mínimo local para o valor de E. Ou seja, o algoritmo pára quando os objetos não mudam mais de grupo.

A Figura 3.4 ilustra os passos apresentados do algoritmo*k-means*e a Figura 3.5 seu pseudocódigo.

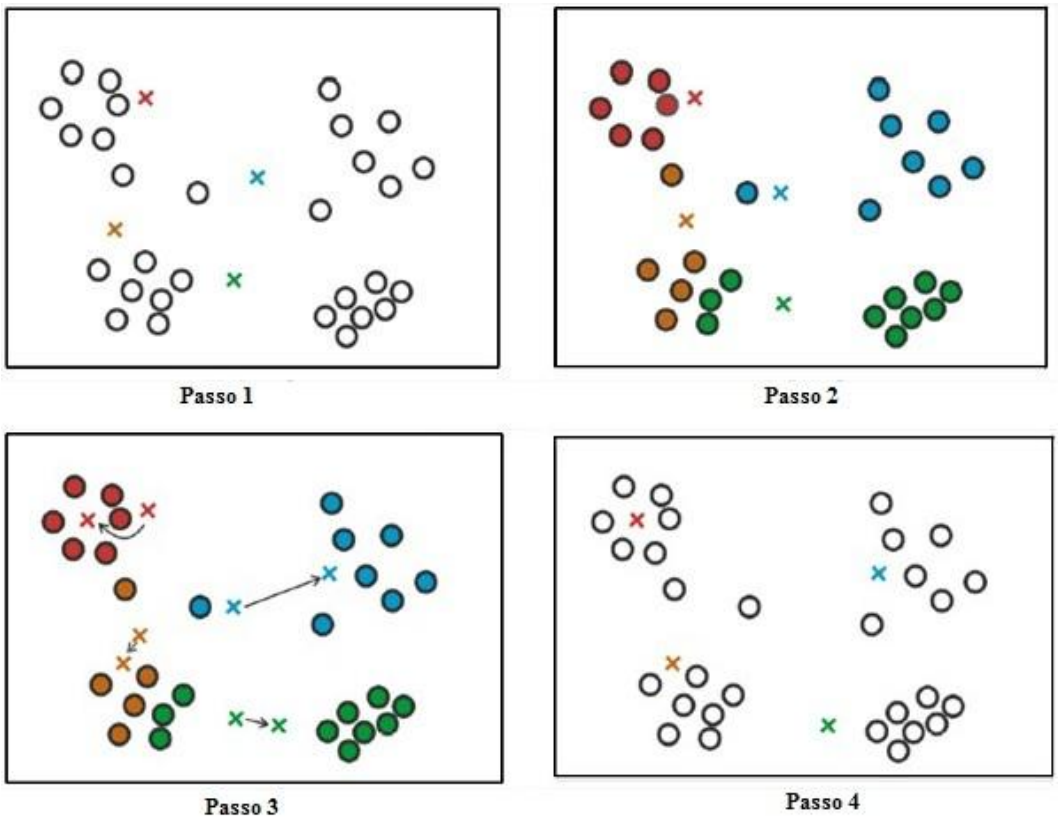


Figura 3-4 - Representação visual do algoritmo *K-means*

Pseudocódigo do algoritmo k-means

Entrada:
$X = \{x_1, x_2, ..., x_n\}$: conjunto de instâncias a serem agrupadas
k: número de grupos

Saída:
$P = \{G_1, G_2, ..., G_k\}$: partição com k grupos

```
1 selecionar aleatoriamente k documentos como centróides
  iniciais;
2 repita
3 |   para cada documento x ∈ X faça
4 |   |   computar a (dis)similaridade de x para cada
  |   |   centroide C ;
5 |   |   atribuir x ao centroide mais próximo ;
6 |   fim
7 |   recomputar o centroide de cada grupo;
8 até atingir um critério de parada;
```

Figura 3-5 - Pseudocódigo do algoritmo *K-Means*

Importante dizer que a determinaçãodo número K de *clusters*que serão aplicados ao algoritmo ajuda ou dificulta a análise e interpretação dos resultados. Existem conjuntos de dados em que o número baixo de cluster trará bons resultados, mas, com outros conjuntos o resultado pode ser diferente. Segundo Han Kamber (2006) é preciso executar o algoritmo diversas vezes com diferentes entradas e avaliar os resultados apresentados.

Demonstrado as principais tarefas utilizadas no trabalho, na Seção 3.5 é apresentadaa mineração em texto.

3.5. Mineração em texto

Tendo suas bases na mineração de dados, a mineração em textos é definida por Barion e Lago (2008) como um conjunto de métodos usados para organizar e descobrir informações em base de texto. A mineração em textos tornou-se amplamente difundida, acompanhando o crescimento da internet e da evolução na área de linguística computacional.

Para realizar a mineração de textos, é preciso fazer o pré-processamento dos dados para que fiquem em formatos e/ou padrões adequados para serem minerados.

Segundo Monteiro *et al.* (2006), a preparação dos dadossegue uma ordem onde, cada etapa contribui com a execução de outra. Essas etapas podem ser dividisdas seguinte forma: correção ortográfica (elimina possíveis erros ortográficos do texto), remoção de *Stopwords*(remove um conjunto de palavras com muita frequência e que não são representativas para o entendimento do texto, como, por exemplo, preposições, artigos, conjunções, pronomes, dentre outros) e *stemming* (remove todas as variações da palavra permanecendo somente a raiz ou s*tem*). O uso de cada etapa depende do que se pretende obter com o texto. No trabalho as técnicas *stopwords* e *stemming* foram utilizadas com o auxilio do algoritmo RSLP (Removedor de Sufixos da Língua Portuguesa).

Por fazer parte deste trabalho, a seção 3.5.1 detalha as características de *stemming.*

3.5.1. Processo *stemming*

Segundo Coelho (2007), a técnica de *stemming* é utilizada na área de Recuperação de Informação - RI com o intuito de melhorar a qualidade do resultado produzido pelo sistema. Tem por objetivo reduzir palavras a uma forma comum de representação, chamada de *stem* (ou radical.), como, por exemplo, as palavras "gelo" e "geleira". Aplicando a técnica de *stemming*, pode-se reduzir essas duas palavras a um único radical (*stem*) comum "gel"(FRAKES; BAEZA-YATES; 1992).

Muitas vezes, o usuário está interessado em documentos que contenham, além das *palavras-chave* que inseriu, as variantes destas palavras. Utilizando a técnica de *stemming* é possível fazer essa busca, melhorando assim o resultado do sistema. Também, com a redução de *palavras-chave* a um único *stem*, é possível reduzir o tamanho do índice utilizado pelo sistema de recuperação de informação (FRAKES; BAEZA-YATES; 1992).

Apesar das vantagens que esta técnica apresenta, algumas desvantagens devem ser consideradas. A primeira é a perda de precisão na recuperação de informação (já que agora se passa a pesquisar somente os radicais das palavras, e não as próprias palavras) e a perda do contexto de informação, pois se podem ter dois *stems* iguais para palavras com sentidos diferentes, como, por exemplo, o *stem* "corr" que pode ser a palavra "corrida" ou a palavra "corrimão".

Ainda assim, com uso de *stemming* foi possível recuperar, minerar e classificar os metadados através da aplicação de regras e exceções. A perda de precisão e contexto das palavras foi baixa e não compremeteu o entendimento destas na base de dados.

3.5.2. Algoritmos de *stemming*

A maioria dos algoritmos de *stemming*foi desenvolvido para a língua inglesa, não apresentando um bom desempenho quando aplicados a línguas derivadas do latim, como o português COELHO (2007). Além disso, a língua portuguesa apresenta certas particularidades que causam alterações profundas no radical das palavras. Logo, há de se esperar que um *stemming* para língua portuguesa encontre maiores dificuldades do que um para a língua inglesa, por exemplo. Algumas dificuldades da língua portuguesa são citadas por Orengo e Huyck (2001) e listadas abaixo:

- **Exceções** - Um exemplo de exceções é que normalmente o sufixo "ão" representa palavras que estão no aumentativo, mas são várias as exceções como, por exemplo, "verão", "estão", etc. É preciso então construir uma lista de exceções para evitar erros;
- **Homografia**- São palavras que possuem forma gráfica igual ou semelhante, mas têm pronúncia e significados diferentes. Um exemplo é a palavra "pregar", que pode significar tanto cravar pregos quanto dar um sermão;
- **Mudanças para a raiz morfológica -** Existem casos em que o processo gera mudanças na raiz da palavra, como, por exemplo a palavra “bombons”, no caso desubstituição de"ns" para "m";

Outra dificuldade que pode ser vista de forma geral são os nomes próprios. Não é recomendado utilizar stemming em nomes próprios, uma vez em que existiria uma grande dificuldade em reconhecê-los. Uma das razões é a grande quantidade de nomes próprios existentes e outra é que muitas vezes eles compartilham o mesmo significado de outras palavras, como, por exemplo, "Machado", que pode significar tanto uma ferramenta quanto um sobrenome.

Mesmo com essas dificuldades, alguns algoritmos foram desenvolvidos para língua portuguesa. Entre eles se destaca o algoritmo de Porter para língua portuguesa (PORTER, 2005), o algoritmo STEMBR (ALVARES; GARCIA; FERRAZ, 2005) e o

Removedor de Sufixos da Língua Portuguesa - RSLP, sugerido por Orengo eHuyck (2001) e melhorado por Coelho (2007). Este algoritmo foi escolhido para ser utilizado neste trabalho, e por isso está detalhado Seção3.5.2.1.

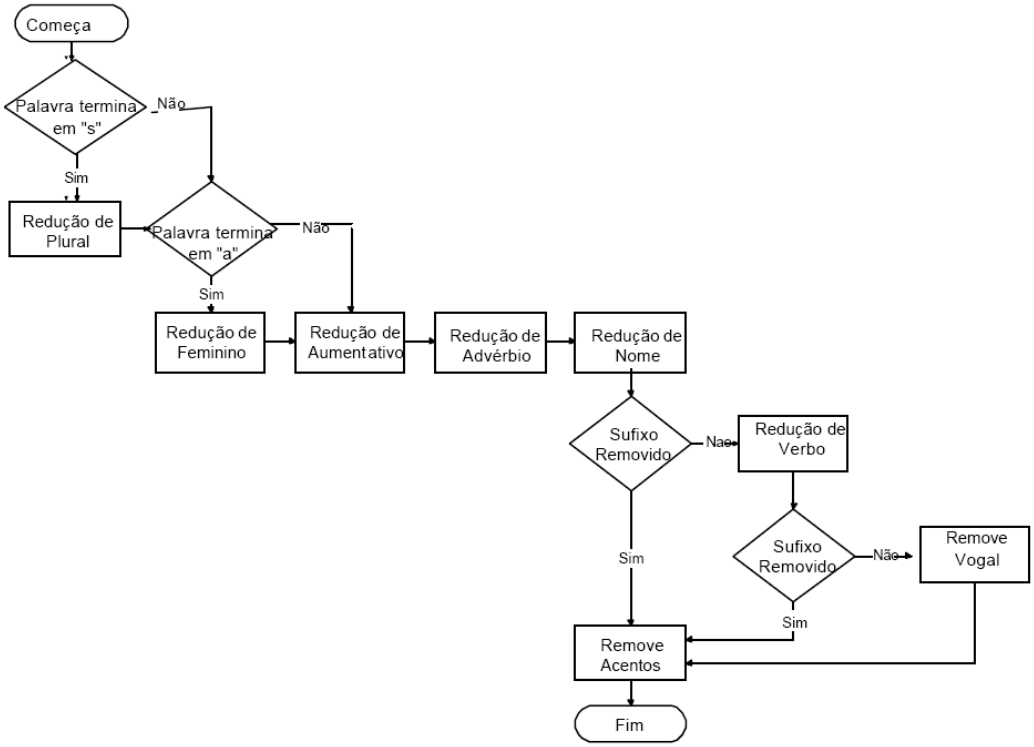
3.5.2.1. Removedor de sufixos da língua portuguesa (RSLP)

Segundo Coelho (2007), o RSLP trabalha aplicando sucessivos passos para a remoção de sufixos sobre uma palavra, definidos através de regras. Além de utilizar regras específicas para o português, ele também conta com um dicionário de exceções, evitando assim remover sufixos de palavras cuja terminação seja similar a um sufixo que se deseja remover (como o sufixo • \.inho. e a palavra • \linho.).

O processo de *stemming* do RSLP é dado por uma sequência de passos que são executadas em uma determinada ordem. O fluxograma mostrado na Figura 3.5 apresenta a sequência que estes passos devem obedecer (LOPES, 2004).

Em cada passo está contido um conjunto de regras. Porém, nem todas as regras são aplicadas. A cada passo as regras são examinadas e apenas uma regra é aplicada. O sufixo mais longo possível é sempre removido primeiro por causa da ordem das regras dentro do passo. Isso garante que, por exemplo, o sufixo “s”, pertencente à redução de plural, não seja removido incorretamente de uma palavra antes do sufixo “es” ser testado.

Uma palavra, quando colocada para passar pela técnica de *stemming*, é testada sucessivamente pelas regras de um passo, verificando a presença do sufixo na palavra processada. Se o sufixo da regra for encontrado, verifica-se se a palavra não consta na lista de exceções daquela regra. Caso não conste, verifica-se se o *stem* resultante possui o tamanho mínimo requerido pela regra. Em caso afirmativo, remove-se o sufixo, adiciona-se o sufixo de substituição (se houver) e encerra-se a execução deste passo, passando para o próximo. Caso a palavra conste na lista de exceções, passa-se para a próxima regra. Caso a lista de regras termine sem que a palavra seja modificada, continua-se no próximo passo.



Fonte: Orengo Huyck (2001)

Figura 3-6 – Trecho da sequência de passos executados pelo algoritmo RSLP.

O algoritmo original, proposto por ORENGO e HUYCK (2001), dava uma sequência imutável da ordem dos passos pela qual uma palavra passa. Com a melhoria proposta por COELHO (2007), esta sequência de passos passou a ser definida pelo usuário, através do carregamento de regras e exceções a partir de um arquivo de configuração. O arquivo de configuração padrão, incluso no algoritmo melhorado e utilizado neste trabalho, propõe a seguinte ordem para os passos de remoção de sufixo:

1. Redução de plural
2. Redução adverbial
3. Redução do feminino
4. Redução do aumentativo
5. Redução nominal
6. Redução verbal
7. Remoção de vogais temáticas
8. Remoção de acentuação

Destes passos, somente o último (remoção de acentuação) é implementado diretamente no programa, sendo sempre o último passo a ser executado.

Dentre as demais melhorias propostas e implementadas por COELHO (2007), estão a documentação do código-fonte, a reorganização da implementação, um dicionário de *stems*, um dicionário de nomes próprios, a configuração do fluxo de execução e a configuração das funcionalidades.

Apresentado os conceitos da mineração e seus desdobramentos, na Seção 4 é feito a experimentação em uma base de dados.

4. EXPERIMENTAÇÃO DA MINERAÇÃO NA BASE DE DADOS DO IBGE

Tendo em vista os conceitos apresentados nas seções anteriores, nesta seção são apresentados os experimentos e os procedimentos realizados para atender aos objetivos propostos neste trabalho.

Na Seção 4.1 demonstra-se o processo de extração e criação da base de metadados, na Seção 4.2 são feitas a preparação e transformação da base através do processo de *stemming*, na Seção 4.3 apresenta-se a transformação dos dados para tarefa a de regras de associação e clusterização e, por final, na Seção 4.4 são demonstrados o processo da mineração e as características para cada tarefa de mineração.

4.1. Criação da base de dados

O primeiro passo para a criação da base de metadados foi obtê-los a partir de repositórios públicos. A base de metadados geográficos do IBGE (Instituto Brasileiro de Geografia e Estatística)[4]foi escolhida por ter um volume de metadados considerável para mineração e pela confiabilidade e integridade de seu conteúdo. Para isso, foi preciso fazer a extração dos dados contidos na base e passar para uma base local, processo este auxiliado pela ferramenta *GeoNetwork*[5].

O *GeoNetwork* é um sistema de gerenciamento de informações geoespaciais aberto e baseado em padrões, projetado para permitir o acesso a bases de dados georeferenciadas e a produtos cartográficos disponíveis em diversos provedores, através de metadados descritivos, potencializando o compartilhamento de informação entre organizações e seus usuários, utilizando os recursos da Internet (IBGE, 2009).É muito utilizado para gestão de grandes volumes de dados geográficos, permitindo a organização destes através de padrões de metadados. Dentre os principais recursos e características, pode-se citar:

- Controle de acesso personalizado;

[4]www.ibge.gov.br
[5]www.metadados.geo.ibge.gov.br/geonetwork

- Permissão para o armazenamento de metadados em diversos sistemas gerenciadores de banco de dados relacional: *MySQL, SQLServer, PostgreSQL* e *Oracle*.
- Coleta programada e sincronização dos metadados distribuídos entre catálogos;
- Carregar (*uploading*) e baixar(*downloading*) dados, documentos e outros conteúdos; e
- Suporte aos padrões de metadados ISO 19115:2003, FGDC e DublinCore.

O *GeoNetwork*fornece a funcionalidade de colheita (*harvesting*) para fazer a coleta de dados em perfeito acordo com os direitos de acesso e posse dos mesmos. Através deste mecanismo é possível obter informações públicas de diferentes nós instalados ao redor do mundo, copiando e armazenando periodicamente esta informação localmente (IBGE, 2009). Também fornece, através de sua ferramenta denominada GAST uma forma para a configuração do banco de dados que armazenará os metadados. O *GeoNetwork* possui um banco padrão, mas optou-se por utilizar o PostgreSQL (POSTGRE, 2010), um sistema gerenciador de banco de dados objeto-relacional.

Para a realização da coleta dos dados, foi criado um banco de dados no SGBD PostgreSQL, depois as 20 tabelas padrão através do GAST (Figura 4.1) e, em seguida, feita a conexão e a carga dos metadados com a base do IBGE pelo*harvesting*.

A principal tabela dentre as geradas é a *metadata*, que contém os metadados. No momento em que foi criada a cópia da base de dados a tabela *metadata* continha 4343 registros. Nesta tabela destacam-se dois campos-chave: *data*, que contém os arquivos XML referentes ao conteúdo do metadado, e *schemaid*, que contém o tipo de padrão do metadado. Entre os padrões identificados, estão:

- ISO-19139 (ISO-19139, 2007) e ISO-19115 (ISO-1915, 2003);
- MGB-Completo e MGB-Sumarizado (CONCAR, 2009);
- Dublin-Core (DUBLIN CORE, 2011);
- FGDC (FGDC, 1998) .

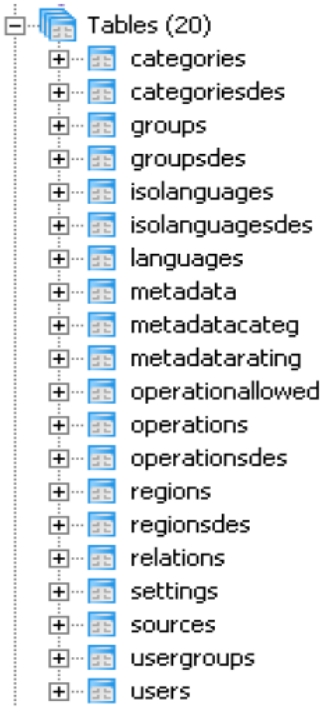


Figura 4-1 - Tabelas criadas pelo GAST

Foram identificados diferentes padrões entre os metadados.Dessa formafoi necessário definir qual informação era relevantepara criar uma base integrada, com padrão únicoe que poderia trazer resultados úteis quando minerada.

Assim, foi definido que os atributos que constariam na base de dados na qual fosse mineradaeram os campos *título*, *resumo* e *palavra chave* devido a suas características semânticas. Posteriormente, foi preciso conhecer a estrutura XML dos padrões existentes para verificar como cada padrão se referenciava a estes campos.

Partindo deste ponto, os desafios foram: extrair a informação referente ao XML do banco de dados, que estava no campo *data*, e separá-los em arquivos distintos; relacionar quais campos, em cada padrão, que correspondiam aos campos *título*, *resumo* e *palavra chave*; extrair os campos do arquivo XML em questão e colocá-los em uma nova base de dados.

Para fazer a extração dos arquivos XML, foi criado um programa em java onde, para cada registro da tabela *metadata*, um arquivo de nome <*padrão*>_<*id*>.*xml*era criado, sendo que <*padrão*> representa qual o padrão em que está o metadado e <*id*> qual o id do registro da tabela *metadata*. Esta nomenclatura foi

adotada para saber a qual padrão o XML pertencia e qual o registro correspondente na tabela original. A Figura 4.2 exemplifica nomes gerados para alguns dos arquivos.

Com os arquivos prontos, foram relacionados os atributos dos arquivos XML que correspondiam às informações que eram de interesse (*titulo, resumoe palavras chave*) e criado um novo programa em java que faria a extração dos elementos dos arquivos XML e os colocaria em uma nova tabela, criada somente com os campos referentes *a id, titulo, resumo* e *palavra chave.*

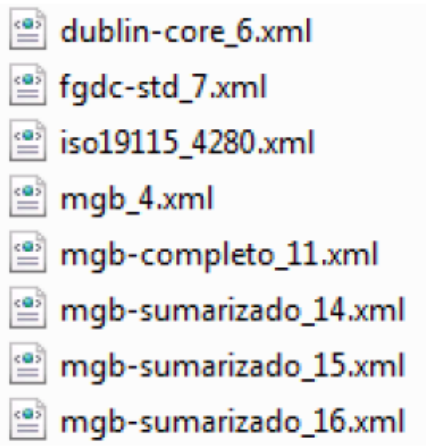


Figura 4-2 - Arquivos extraídos do banco de dados

Este programa utiliza a biblioteca JDOM (JDOM, 2009) para manipulação dos arquivos XML e lê um arquivo de configuração chamado *config.ini*. Neste arquivo, o usuário deve colocar o caminho da pasta onde será feita a leitura dos arquivos XML, a tabela na qual os dados serão inseridos e qual campo do arquivo XML se relaciona com qual campo da tabela. Com base nestas informações, o programa lê cada arquivo XML, obtém os dados contidos nos campos especificados pelo usuário e os adiciona no campo correspondente da tabela. Caso o mesmo campo apareça mais de uma vez no arquivo XML (por exemplo, *<kw>Teste</kw>* e *<kw>Teste1</kw>*), a informação que irá para o campo correspondente na tabela será da forma *<dado1>;<dado2>;..;<dadoN>* (no exemplo anterior, o dado a ser inserido no campo seria *Teste;Teste1*). Dessa forma, cada arquivo XML terá um registro na tabela, como mostra a Figura4.3.

Com a visualização dos metadados na base de dados, inicialmente percebeu-se que o campo título não seria relevante neste momento, devido ao alto grau de

semelhança das ocorrências neste atributo. Logo, nas etapas posteriores, começou-se a trabalhar somente com os campos *resumo* e *palavra chave*.

id [PK] serial	titulo text	resumo text	palavra_chave text
9	Carta Topográfica Matrici	Esta folha é parte integrante	Cartografia;SÃO JOSÉ DO BARREIRO / RJ_SP;Carta Topogr
10	Carta Topográfica Vetoria	Esta folha é parte integrante	Cartografia;VITÓRIA / ES;Carta Topográfica Vetorial
11	Carta Topográfica Matrici	Esta folha é parte integrante	Cartografia;TUPACIGUARA / MG;Carta Topográfica Matric
12	Carta Topográfica Vetoria	Esta folha é parte integrante	Cartografia;MARILAC / MG;Carta Topográfica Vetorial

Figura 4-3 - Exemplo dos metadados geradosa partir do XML

Com as informações necessárias em um banco de dados único e padronizado, a próxima etapa foi a fase de preparação dos dados, conforme detalhado na Seção 4.2.

4.2. Fase de preparação dos dados

No processo de descoberta de conhecimento, esta é a fase que mais consome tempo (HAN; KAMBER, 2006), e neste trabalho não foi diferente.

Nesta etapa, a fase de preparação foi feita através da aplicação da técnica de *stemming*(Seção 3.5.1) na base de dados criada. A base foi transformada ("*stemmizada*"), de forma a reduzir palavras a um radical comum.

Para isso, foi utilizado uma implementação do algoritmo RSLP melhorado, proposto por COELHO (2007), conforme apresentado na Seção 3.5.2.1. A implementação foi feitada seguinte forma:primeiroos dados da tabela criada na Seção 4.1 foram acessados.Em seguida, fez-se um pré-processamento doscampos *resumo* e *palavra chave*utilizando a remoção de *Stopwords* conforme visto na Seção 3.5. Logo após, o programa Java utiliza as palavras resultantes do pré-processamento para criardois arquivosque servirão de entrada para o algoritmo RSLP, entrada_keyword.txt e entrada_*resumo*.txt. A Figura 4.4 ilustra trechos dos arquivos mencionados.

.

Figura 4-4 - Exemplo de arquivos *entrada_resumo.txt* e *entrada_Keyword.txt*

Após a criação destes dois arquivos, o programa Javaexecuta um programa em C, onde foi implementado o algoritmo RSLP. Este programa lê os dois arquivos de entrada mencionados, realiza o processo de *stemming* em cada um deles e cria dois novos arquivos, chamados *saída_keyword.txt* e *saída_resumo.txt*, contendo os respectivos *stems*.

Figura 4-5 - Exemplo de arquivos *saida_resumo.txt* e *saida_keyword.txt*

De posse dos arquivos de saída, a aplicação Javaque fez a criação dos arquivos *entrada_resumo.txt* e *entrada_keyword.txt* novamente é chamada para ler os doisarquivos de saída gerados (Figura 4.5), e assim colocá-los em uma nova tabela, contendo um id sequencial, um id de base, referente ao id na tabela não *stemmizada* e os *stems* de *resumo* e*palavra chave*. A Figura 4.6 ilustra uma parte dos dados depois do processo de *stem* concluído na tabela gerada.

	id integer	id_base integer	resumo text	palavra_chave text
1	1	3311	planimetr,sgb,piaui,sister	planimetr,red,cartogram,
2	2	3312	integr,seri,servic,abrang	cartograf
3	3	3313	altimetr,espirit,brasil,est,	rn,altimetr,red,cartogram

Figura 4-6 - Exemplo dos metadadosstemmizados

Com as informações *stemmizadas* no banco de dados, verificou-se os possíveis dados inconsistentes e lixos que poderiam atrapalhar a etapa seguinte de transformação de dados e observou-se que alguns metadados do campo *resumo* e/ou *palavrachave* estavam em branco ou possuíam lixo de importação.Exemplos destas situações podem ser vistos na Figura 4.7. As linhas 4,5,6 e 9 da tabela são exemplos de dados em branco, enquanto as linhas 10 e 21 trazem um exemplo de lixo de importação.Estes lixos aconteceram devido alguns metadados não possuir dados para os campos de resumo e/ou palavra chave.

	id integer	resumo text	palavra_chave text
3	3313	represent graf red altime	geodes red altimetr rn cartogram
4	3314		sol map brasil recurs natur
5	3315		sol map brasil recurs natur
6	3316		sol map brasil recurs natur
7	3317	preench resum exempl n	sol map brasil recurs natur
8	3318	preench resum exempl n	sol map brasil recurs natur
9	3319		sol map brasil recurs natur
10	3320	est folh part integr seri c	i˜~v
11	3321	est folh part integr seri c	cartograf
12	3322	map sol est acr map ider	sol map acr recurs natur
13	3323	represent graf red altime	geodes red altimetr rn cartogram
14	3324	represent graf red altime	geodes red altimetr rn cartogram
15	964	ortofotomosa integr proj	travess rj ortofot mape topograf aerolev
16	965	ortofotomosa integr proj	ururaix rj ortofot mape topograf aeroleva
17	966	ortofotomosa integr proj	trindad rj sp ortofot mape topograf aerol
18	967	ortofotomosa integr proj	traj moral rj ortofot mape topograf aerole
19	968	ortofotomosa integr proj	toc rj ortofot mape topograf aerolevant
20	969	est folh part integr seri c	cartograf itambacur mg cart topograf vet
21	970		i˜~v

Figura 4-7 - Exemplo de metadados inconsistentes na base *stemmizada*

Com isso, fez-se o processo de retirada destes metadados da base, preservando a integridade do restante dos dados e assim a preparação dos metadados ficou pronta para a etapa de transformação, conforme mostrado na Seção 4.3.

4.3. Transformação dos dados

Nesta etapa fez-se as transformações dos dados para formatos apropriados para serem utilizados no processo de mineração de dados (Seção 4.4).

As tarefasde mineração testadas no trabalhoforam a análise de regras de associação e clusterização. Cada tarefa possui características próprias no processo de mineração e, por isso, foi preciso analisar as transformações necessárias a parte de cada uma.

Embora a transformação para a tarefa de regras de associação já estivesse pronta, o processamento dos dados teve um custo computacional alto e os resultados não foram úteis. Com isso foi necessário fazer a transformação da base (campos *resumo* e *palavras chave*) em pares para tentar diminuir o custo computacional e descobrir regras satisfatórias para a descoberta do conhecimento.

Para realizar essa transformação, criou-se um programa na linguagem PHP[6](*Hypertext Preprocessor*). O primeiro passo foi extrair as palavras dos campos *resumo* e *palavrachave* para vetores,um vetor de *resumo* e outro de *palavraschave*, onde cada posição do vetor é uma palavra referenciada nesses campos. Como próximo passo, o programa percorre o vetor *resumo* e o vetor de *palavras chave*, relacionando cada palavra do *resumo*a todas as *palavraschaves*do vetor *palavras chave*. Para finalizar faz-se a inserção do id sequencial, id base (referência da base não *stemmizada*) e de cada par gerado em uma tabela no banco de dados. Esse processo é repetido para cada registro da base de dados *stemmizada*. A Figura 4.8 ilustra o processo de transformação da base *stemmizada* para a base par de um registro.

[6]www.php.net

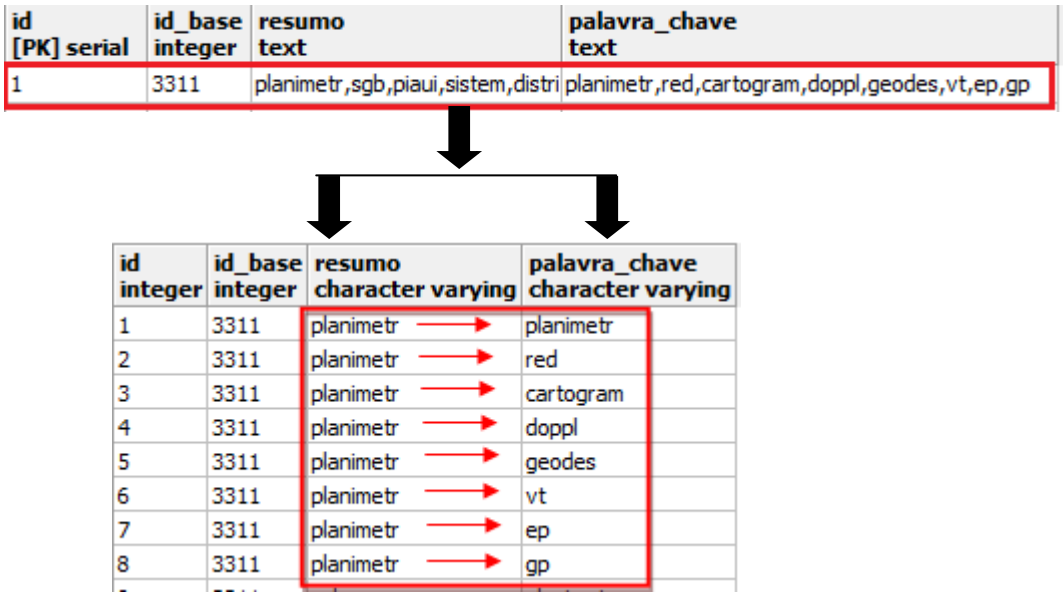


Figura 4-8 - Exemplo de transformação da base *stemmizada* para base par

Para tarefa de clusterização foi criado outro programa PHP onde a transformação foi feita em duas etapas para atender as necessidades da tarefa.

Na primeira etapa, fez-se a seleção de todas as palavras distintas da base *stemmizada*. No primeiro passo, o programa percorre cada tupla da base*stemmizada* e faz-se a união das palavras dos campos *resumo* e *palavrachave* em um vetor da tupla. Após este passo, é eliminado do vetor as palavras que se repetem.Em seguida,compara-seas palavras existentes no vetor de tuplas com as palavras do vetor de palavras distintas, caso a(s) palavra(s) já exista(m) a(s) mesma(s) é (são) descartada(s), caso contrário, é(são) inserida(s) no vetor de palavras distintas. Este processo se repete para todas as tuplas da base. A Figura 4.9 demonstraeste processo de forma ilustrada.

Na segunda etapa da transformação atribui-se valores para cada palavra distinta identificada na etapa anterior para montar a matriz de dados conforme abordado na Seção 3.4.1.

Antes de começar esta etapa foi preciso identificar o tipo de variável que seria utilizado no preenchimento da matriz. Verificou-se pelas características dos atributos da matriz (*palavras-chaves*) que o tipo de variável a ser utilizado seria binário. Com este tipo é possível identificar a existência ou não de um objeto dentro da matriz e, como o

objetivo era identificar a existência de cada palavra distinta em cada tupla, esta escolha foi justificada.

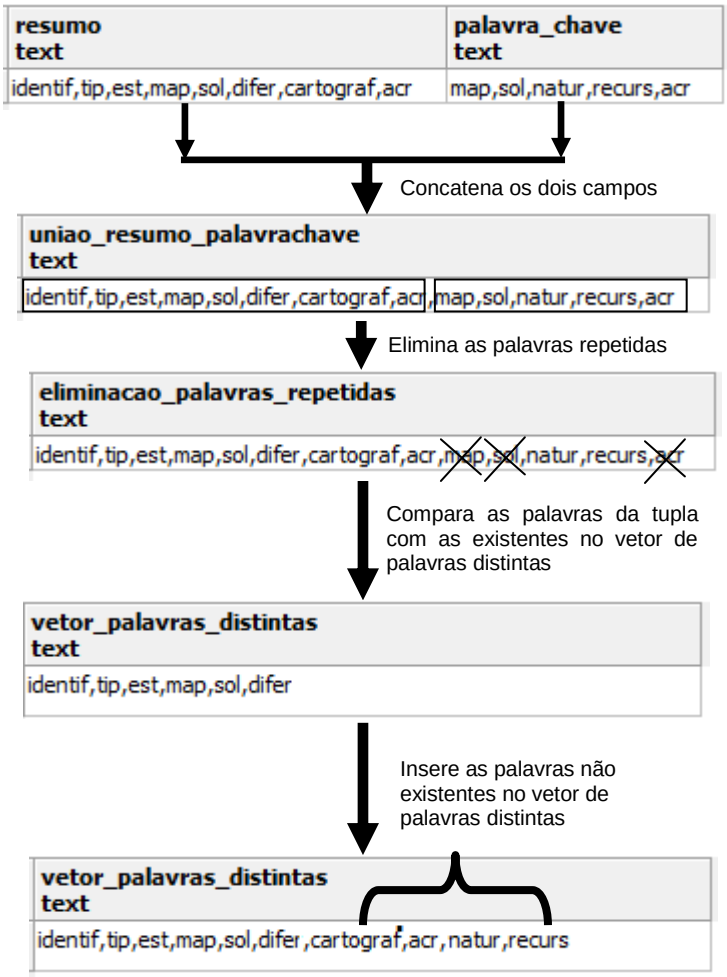


Figura 4-9 - Exemplo do processo de extração das palavras distintas da base *stemmizada*

A partir desta escolha foi feita a programação para realizar o preenchimento da matriz de dados. O programa utiliza o vetor de palavras distintas para verificar a existência ou não das palavras em cada tupla na base *stemmizada*.

Para cada palavra verificada na tupla é feito o preenchimento com 1 (para existência da palavra distinta) ou com 0 (para a ausência). Após este preenchimento os dados são inseridos em uma tabela a qual contém os campos *id* (referência da tupla na base *stemmizada*) e nomes de cada palavra distinta. A Figura4.10destacaum exemplo com parte dos dados preenchidos.

id integer	abrang integer	cart integer	cartograf integer	folh integer	geograf integer	integr integer	latitud integer	longitud integer
3312	1	1	1	1	1	1	1	1
3313	0	0	0	0	0	0	0	0
3317	0	0	1	0	0	0	0	0
3318	0	0	1	0	0	0	0	0
3321	1	1	1	1	1	1	1	1
3322	0	0	1	0	0	0	0	0
3323	0	0	0	0	0	0	0	0
3324	0	0	0	0	0	0	0	0
964	1	0	0	1	1	1	1	1
965	1	0	0	1	1	1	1	1
966	1	0	0	1	1	1	1	1
967	1	0	0	1	1	1	1	1
968	1	0	0	1	1	1	1	1
969	1	1	1	1	1	1	1	1
971	1	1	1	1	1	1	1	1
972	1	1	1	1	1	1	1	1
2310	1	1	1	1	1	1	1	1
2311	1	1	1	1	1	1	1	1
2312	1	1	1	1	1	1	1	1
2313	1	1	1	1	1	1	1	1

Figura 4-10 - Exemplo da transformação dos dados paramatriz binária

Com a fase de transformação dos dados dastarefas regras de associação e clusterização concluída, passou-se para a etapa de mineração dos dados, onde os detalhes são apresentados na Seção4.4.

4.4. Mineração da base de metadados

Depois das etapas de coleta, preparação e transformação dos dados é possível minerar e tentar descobrir algum conhecimento útilnos dados. Para realizar o processo de mineração foi preciso utilizar uma ferramenta de auxílio para receber e minerar os dados transformados.

Esta fase é extremamente prática, nela é descrita a ferramenta utilizada, além dos procedimentos, características e algoritmos para o processo de mineração com as tarefas de regras de associação e clusterização.

4.4.1. Ferramenta utilizada na mineração

Para realizar a mineração de dados, foi utilizado o framework *Weka*na versão 3.6.4 (*WEKA*, 2010). Tratado como um *workbench* (conjunto de ferramentas que facilitam o desenvolvimento de determinado tipo de aplicação ou tarefa), o *Weka* inclui algoritmos para regressão, classificação, agrupamento, associação e outros, além de disponibilizar ferramentas de visualização e de pré-processamento (HALL *et al.*, 2011). Consiste de um pacote desenvolvido pela Universidade de Waikato, em 1999, com o intuito de agregar algoritmos para mineração de dados. Foi desenvolvido em linguagem Java e é licenciado pela *General Public License* - GPL (GNU, 2007), sendo assim permitida a alteração do seu código fonte.

A ferramenta é de fácil instalação e disponibiliza interface gráfica de usuário (GUI – *Graphic User Interface*) para suas funcionalidades, sendo a principal a interface Explorer (Figura 4.11). Esta interface foi desenvolvida para o processamento de dados onde os dados são todos carregados e depois analisados.

Figura 4-11 - Opções para interface *weka*

Na interface *Explorer* está o acesso às funcionalidades da ferramenta, separadas em painéis por tarefas de mineração de dados (Figura 4.12). Os painéis estão ordenados de forma que a primeira opção disponível é o pré-processamento, também chamado de filtros. Em seguida, o painel *Classify*, do inglês classificar, trata dos algoritmos de classificação e regressão. Já os painéis *Cluster* e *Associate*, que podem ser traduzidos para agrupar e associar, tratam respectivamente de algoritmos para

agrupamento (clusterização) e associação. O painel seguinte,trata a seleção de atributos, para a qualo*WEKA* disponibiliza o *Select Attributes*. Por fim, há o painel *Visualize*, para visualização dos resultados.

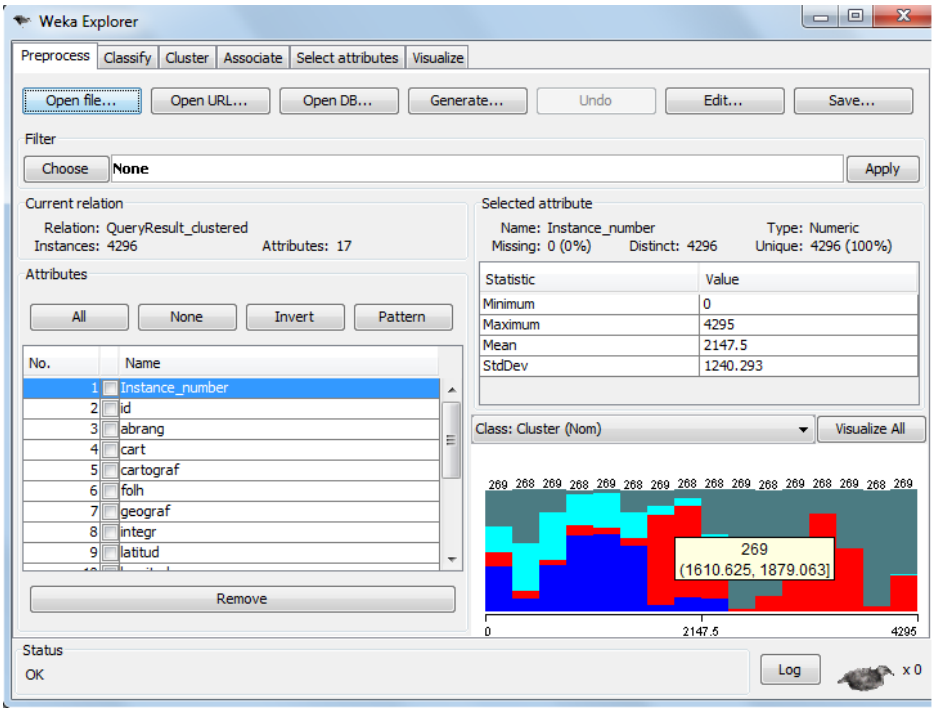


Figura 4-12 - Painéis de opções *WEKA*

Todas as funcionalidades do *Weka* partem do pressuposto de que os dados estão disponíveis em um único arquivo que pode ser acessado de forma sequencial (ou seja, flat files, como arquivos .csv) ou relações, onde cada registro é limitado por um número fixo de atributos (normalmente numéricos ou nominais).

Também é possível acessar os dados a partir de banco de dados utilizando conexões de banco de dados em Java e fazer com que os dados a serem utilizados na mineração sejam provenientes de uma consulta SQL. A Figura 4.13 ilustra os passos para realizar o acesso ao banco de dados e a seleção dos dados para mineração.O passo 1 indica a opção de conexão com banco de dados, o 2 demonstra a forma de conexão com o banco, no 3 é selecionado a opção para informado o usuário e senha, no 4 informa-se o usuário e senha correspondente, no 5 faz a conexão com banco, no 6 faz a consulta dos dados a serem carregados, no 7 executa a consulta e por fim o 8 faz a carga dos dados para dentro do Weka.

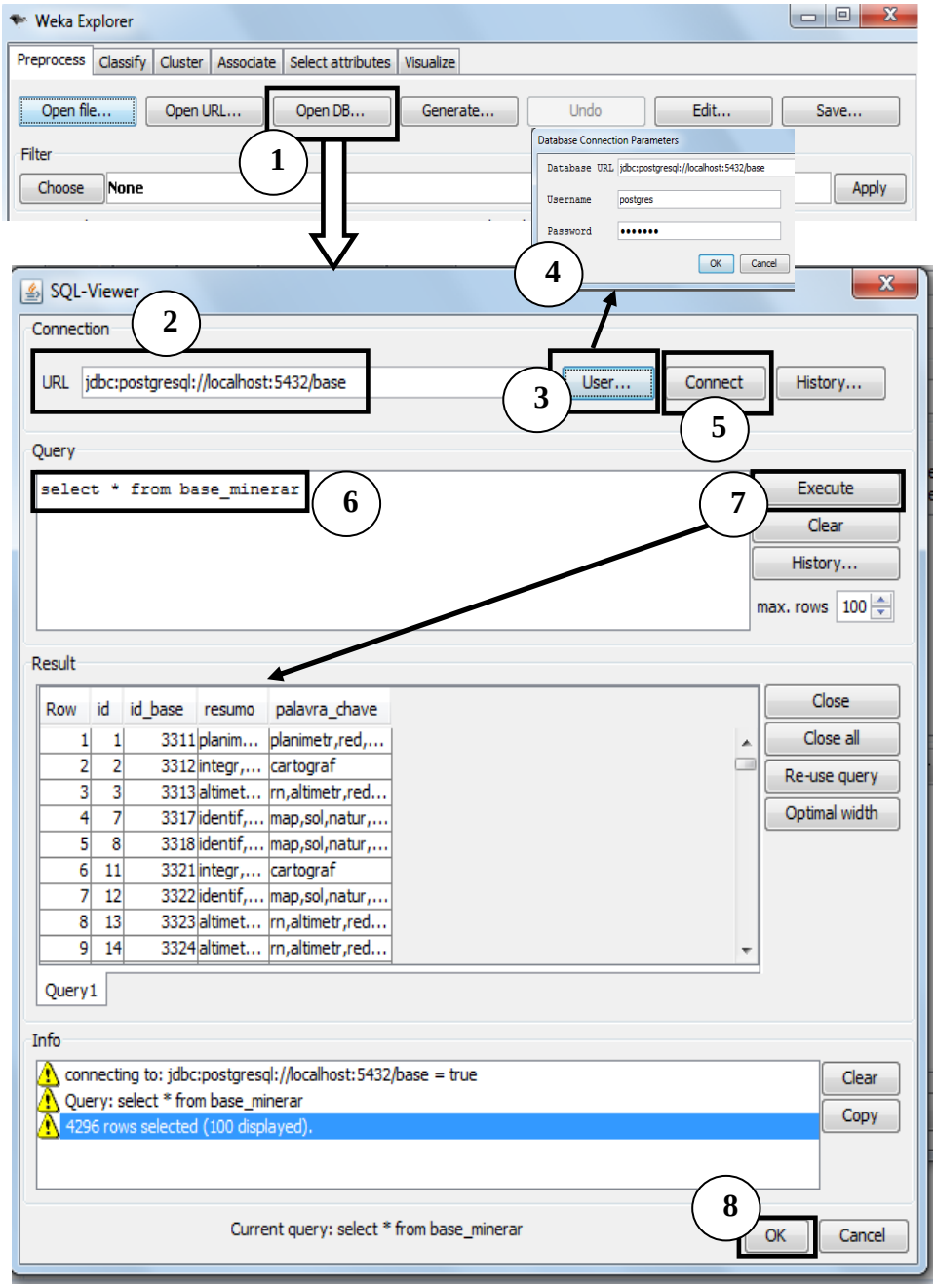


Figura 4-13 - Seleção de dados a partir de um banco de dados

A ferramenta ainda disponibiliza outras interface como: *Experimenter*, *Knowledge Flow* e *Simple CLI*, que não fazem parte do escopo deste trabalho.

Visto as características da ferramenta utilizada no trabalho, nas seções 4.4.2 e 4.4.3 é demonstrado o seu uso na mineração das tarefas de regras de associação e clusterização.

4.4.2. Regras de associação

A tarefa de associação foi a primeira tentativa de descobrir conhecimento útil na base de dados. Conforme visto na Seção 3.3 a tarefa de regras de associação analisa quão frequentemente dois ou mais objetos se correlacionam. Por exemplo, sempre que aparece a palavra “relevo” no campo *resumo*, aparece também a palavra “mapa” no campo *palavrachave*.

A mineração nesta tarefa deu-se em duas tentativas. Na primeira tentativafez-se a seleção dos dados através da conexão com a base de dados *stemmizada*. Os campos *resumo* e *palavra-chave* foram selecionados (Figura 4.14) e os campos numéricos *id* e *idbase* foram excluídos da seleção devido só fazer sentido descobrir regras de associação com dados nominais e porque esses campos não são propriamente dados, mas, só identificadores de registros.

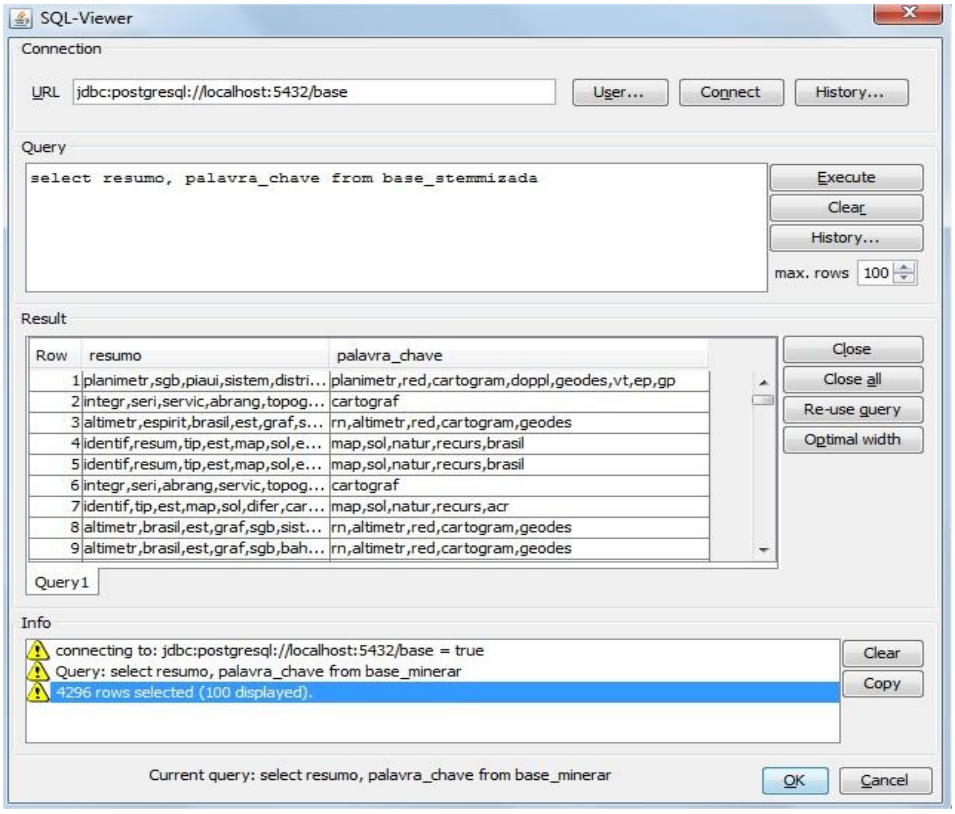


Figura 4-14 - Seleção dos dados da base *stemmizada* para mineração de associação

Após a seleção, os dados foram carregados. Escolheu-se o tipo de mineração *Associate*, o algoritmo a ser utilizado o *Apriori*e fez-se a configuração dos parâmetros

de suporte mínimo (*lowerBoundMinSupport*), tipo de métrica (*metricType*) e ovalor mínimo para métrica (*minMetric*) conforme demonstrado na Figura 4.15.

Depois de várias tentativas utilizando os parâmetros mencionados foi encontrado regras de associação na base. O algoritmo *Apriori* foi configurado com um suporte mínimo de 10% e a métrica *lift* com 1,7. Suporte mínimo, conforme descrito na Seção 3.3, se refere à proporção de transações que contém o conjunto de itens. *Lift* é uma medida simples de correlação que mede o quanto dois conjuntos de itens são independentes (HAN.; KAMBER, 2006). No caso, dado dois conjuntos A e B independentes, o *lift* entre eles é a razão entre a probabilidade deles ocorrerem juntos sobre a probabilidade de A e B ocorrerem, ou seja:

$$lift(A,B) = \frac{P(A \cup B)}{P(A)P(B)}$$

Caso essa razão seja menor que um, então A e B estão negativamente correlacionados, ou seja, a ocorrência de um "desencoraja" o outro a aparecer. Caso essa razão seja maior que um, então A e B estão positivamente correlacionados, ou seja, a ocorrência de um faz com que seja mais provável ter a ocorrência de outro. Caso essa razão seja igual a 1, A e B são independentes entre si e não há nenhuma correlação entre os dois.

Apesar de conseguir encontrar regras com os dados da base *stemmizada*, identificou-se que as regras não trouxeram resultados satisfatórios. A quantidade de palavras no campo *resumo* dificultou implicações de outros tipos de associações com o campo *palavra-chave*. O número de regras encontradas ficou restrita a palavras e tuplas homogêneas dentro da base. A Figura 4.16 mostra as regras e os seus resultados.

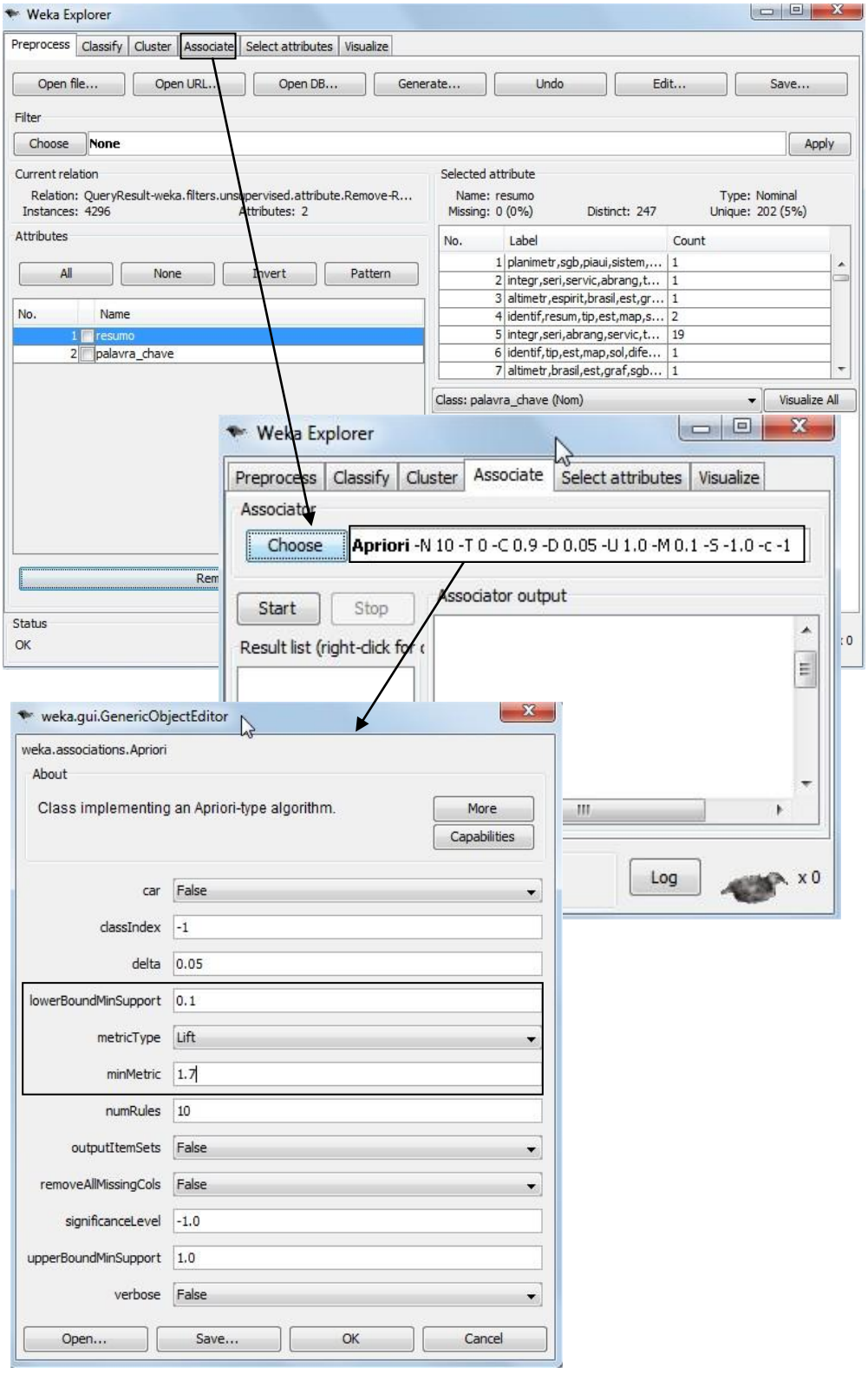


Figura 4-15 - Escolha do tipo de mineração, algoritmo e parâmetros da tarefa de associação

Visto a ineficácia da primeira tentativa, na segunda fez-se a divisão da base em pares, conforme demonstrado na Seção 4.3, na tentativa de melhoria das associações entre as palavras docampo *resumo* e *palavra chave*.

O processo de seleção dos dados seguiu o mesmo princípio apresentado na primeira tentativa com o foco direcionado para a base de dados com as palavras em pares (Figura 4.17). A escolha do tipo de mineração, algoritmo e parâmetros da tarefa de Associação também seguem o mesmo exemplo já apresentado (Figura 4.15), com exceção dos parâmetros.

Mesmo com a mudança da base em pares, não conseguiu-se estabelecer nenhuma regra de associação útil. Este resultado deve-se pelo grande número de pares que foram gerados na base (658.662 tuplas de pares). Nenhum par alcançou os parâmetros configurados como requisitos mínimos de interesse para regra.Diversas tentativas foram realizadas, variando-se os parâmetros de suporte, confiança e tipo de métrica, porém tais testes também se mostraram improdutivos. Apenas o relaxamento muito alto para o parâmetro de suporte (0,0001) retornou algumas regras, porém, também sem maiores aplicabilidades. A figura 4.18 ilustra os resultados.

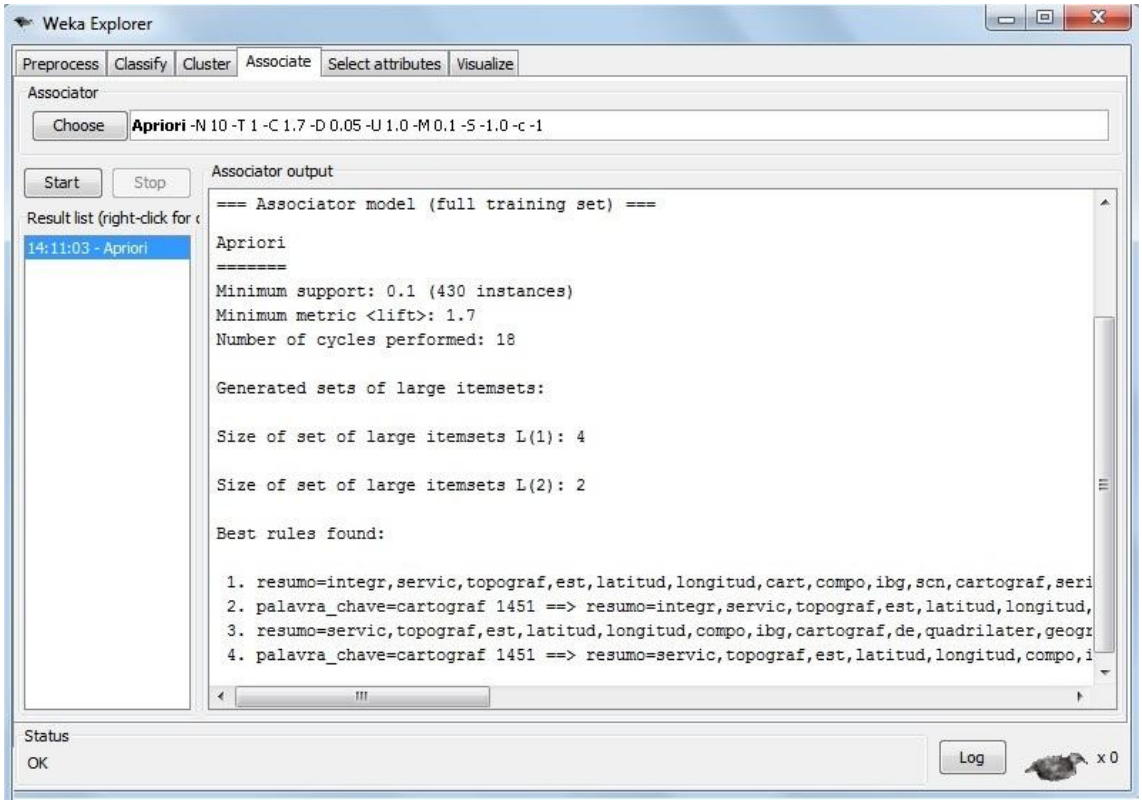


Figura 4-16 - Regras de associação encontradas com a base *stemmizada*

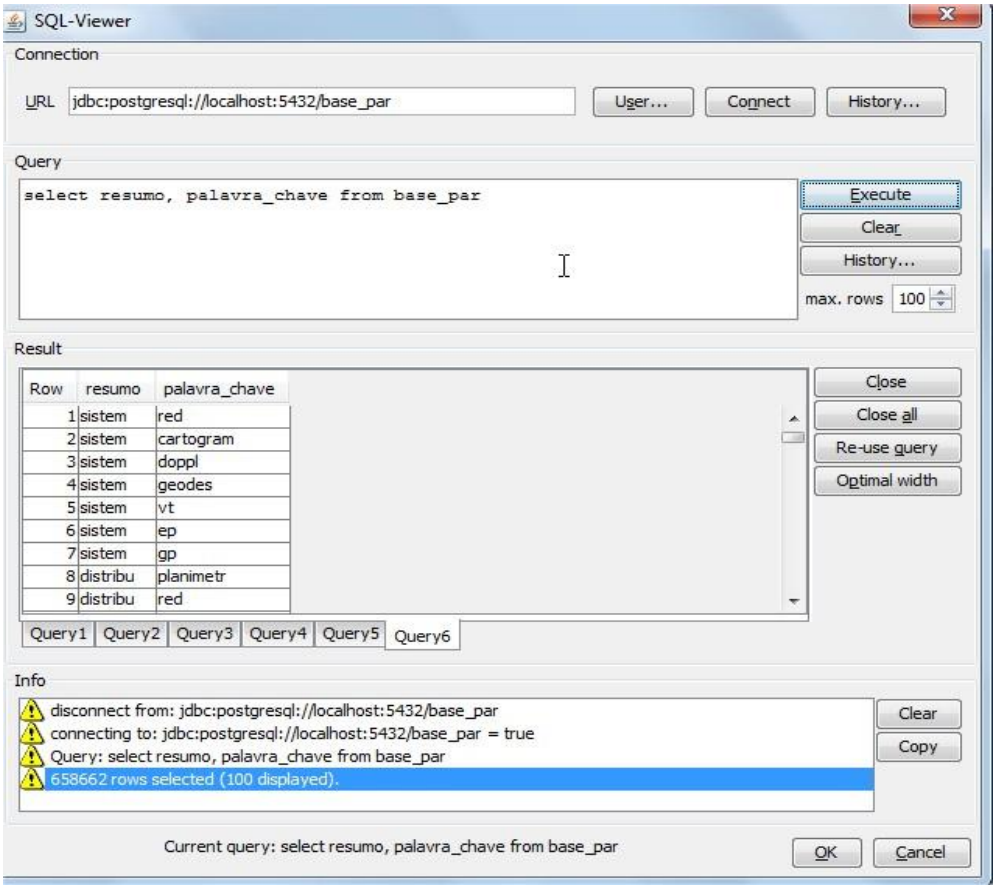


Figura 4-17 - Seleção dos dados da base par para mineração de associação

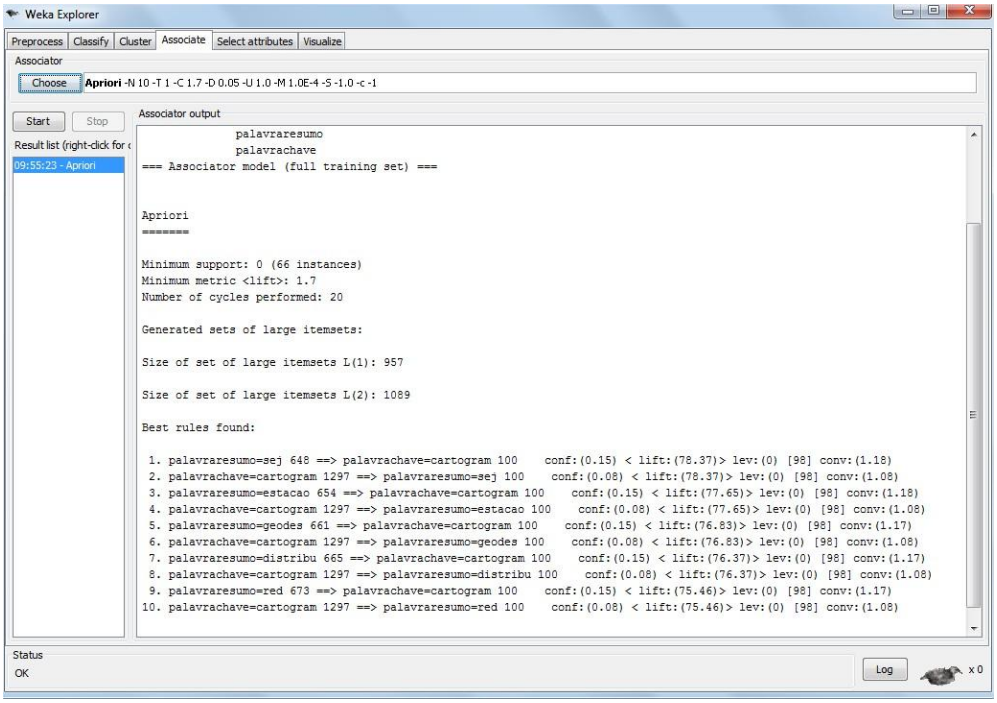


Figura 4-18 - Regras de associação encontradas com a base par

Com os resultados não satisfatórios com a tarefa de regras de associação foi feitaa tentativa de mineração utilizando a tarefa de análise de *cluster* (clusterização), a fim de se tentar descobrir outra forma de conhecimento oculto na base de dados, conforme será demonstrado na Seção 4.4.3.

4.4.3. Clusterização

Conforme apresentado na Seção 3.4, a tarefa de*cluster*ização é uma forma de agrupamento de dados. Tem por objetivo colocar os dados semelhantes juntos num mesmo grupo e os pouco semelhantes em grupos distintos.

Neste trabalho, utilizou-se a tarefa de *cluster*ização para fazer o agrupamento dos metadados semelhantes entre si em um mesmo *cluste*r e diferentes em outros*clusters*; identificar qual(is) palavra(s) poderia(m) discriminar os metadados e categorizar o conjunto de metadados relacionados em cada *cluster*.

Para realização da mineração nesta tarefa fez-se a conexão com a base de dados da matriz binária gerada na Seção 4.3. Os campos de *id base* e as palavras distintas foram selecionados (Figura 4.19) e em seguida os dados foram carregados e pré-processados (Figura4.20).

Em seguida foi feita a escolha do algoritmo para realizar a mineração (Figura 4.21). O algoritmo selecionado foi o *K-Means* chamado na ferramenta *Weka* de *SimpleKMeans*. Ressalta-se que foram testados outros algoritmos como o EM, *HierarchicalClusterer* e *K-medoids*, mas a escolha do *K-Means* deu-se pelas suas características apresentadas na Seção 3.4.2, pela forma de apresentação e visualização dos seus resultados na *Weka* e por se adequar às necessidades do trabalho. Para maiores detalhes sobre os outros algoritmos ver (CARLANTONIO, 2001).

Logo após, fez-se a configuração dos parâmetros exigidos no algoritmo (Figura 4.21). Essa configuração deu-se da seguinte forma:

1. Número de *clusters*: para cada um dos filtros executou-se a mineração para cinco conjuntos de *clusters*: 2, 4, 6, 8, 20. Estes números foram escolhidos na tentativa de facilitar a análise dos resultados e visualização das características dosmetadados semelhantes em cada *cluster*.

2. Número aleatório para escolha (alocação) dos centroides iniciais (*seed*): para os testes realizados, utilizou-se os seguintes números randômicos: 1; 2; 5; 10 (padrão da ferramenta) ; 25; 50 e 100. Estes valores foram utilizados na tentativa de melhorar os resultados do algoritmo.
3. Modo de *cluster:* para todos os testes foi utilizado o modo padrão "*use training set*", ou seja, depois da geração dos *clusters*, o algoritmo classifica as instâncias de treinamento dentro dos *clusters* já representados, e calcula a porcentagem de instâncias em cada *cluster*.
4. Medida de distância: as medidas utilizadas com frequência foram a Euclidiana e a Manhattan. Estas medidas não apresentaram diferenças significativas nos resultados da mineração no trabalho.

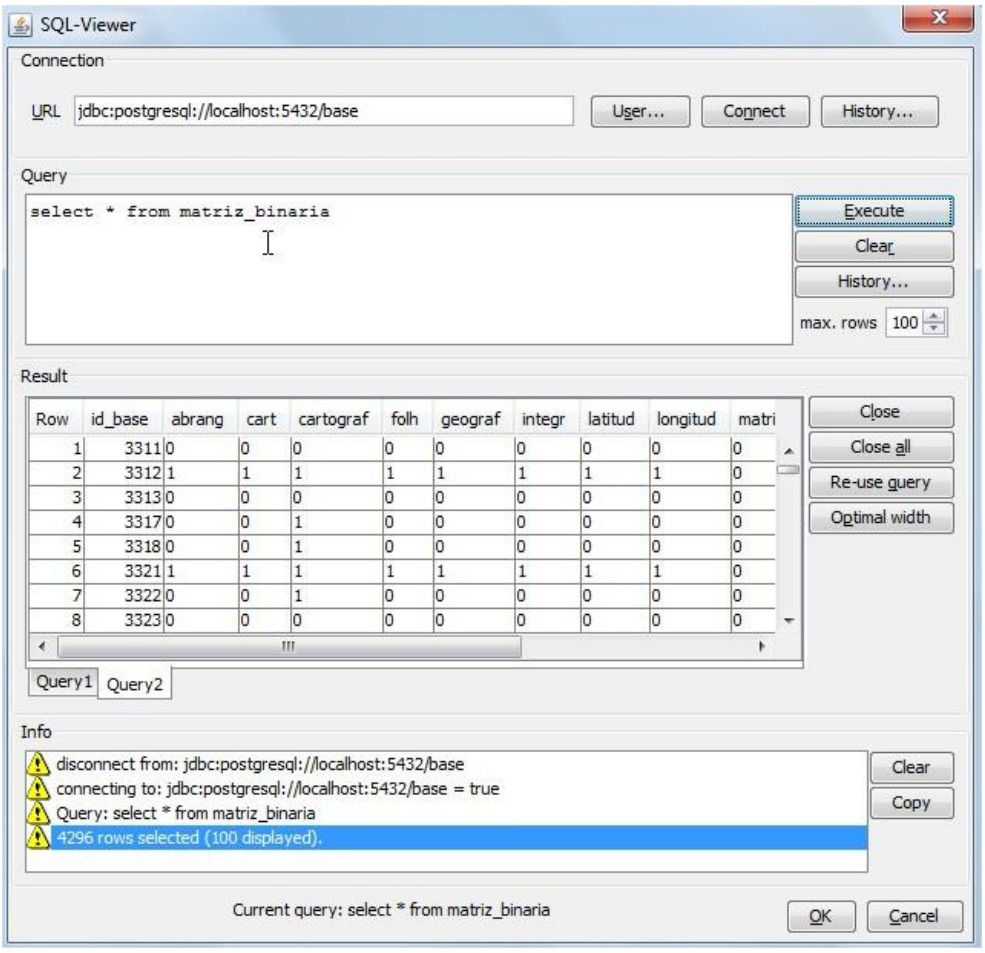


Figura 4-19 - Seleção dos dados da base matriz binária para mineração

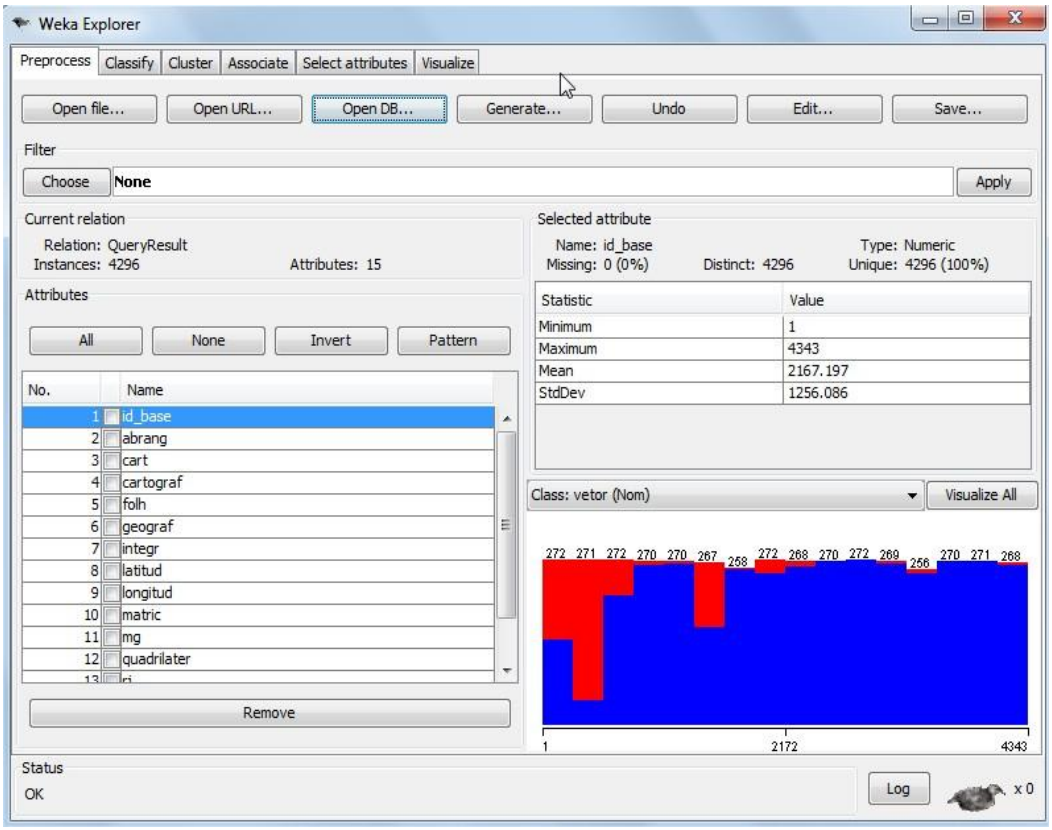


Figura 4-20 - Dados carregados e pré-processados pelo *Weka*

Após as configurações dos parâmetros, o campo*idbase* foi ignorado devido a não fazer parte do processo de mineração. Importante lembrar que mesmo o campo sendo ignorado ele não é excluído e poderá ser útil para análise dos resultados. No trabalho, este campo foi utilizado para referenciaras instâncias de metadados relacionadasem cada *cluster*e seus conteúdos. A Figura 4.21 ilustra o processo de escolha do algoritmo*K-Means* e os parâmetros de configuração.

Feitas todas as configurações necessárias, fez-se a mineração com as 1611 palavras distintas.Apesar do *Weka* separar o número e as porcentagens das palavras em cada *cluster*, percebeu-se pelos resultados apresentados a complexidade em achar as relações e características entre as palavras.A Figura 4.22 e 4.23 ilustram um exemplo de como os resultados são apresentados com o *K-Mean.*

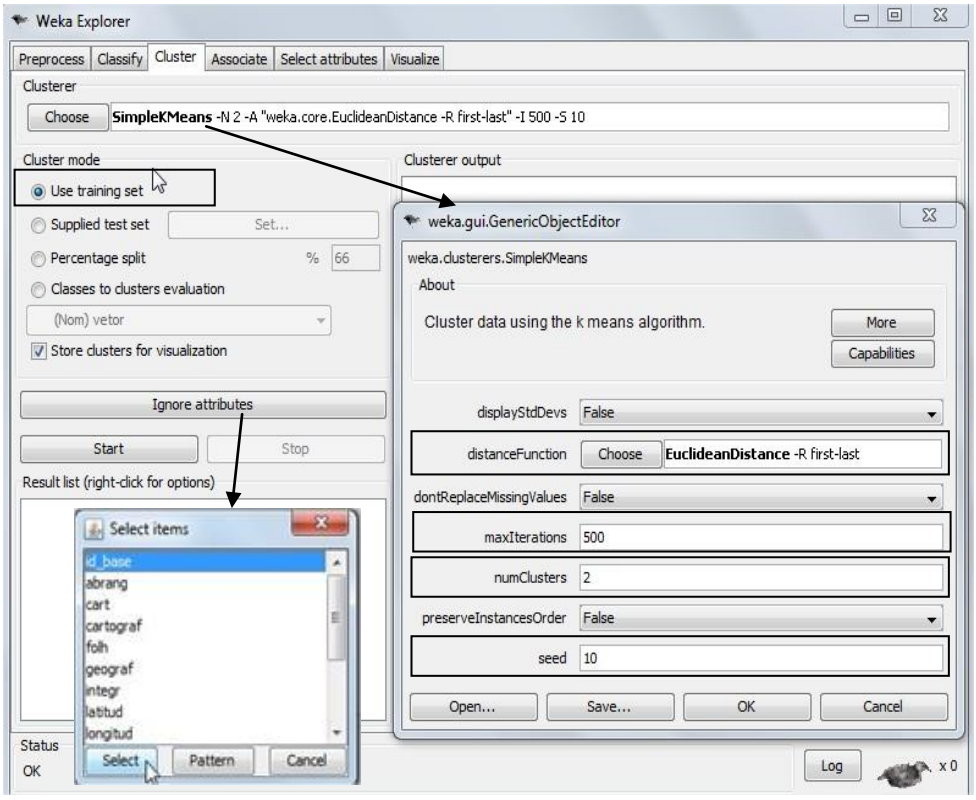


Figura 4-21 - Seleção do algoritmo *K-means* e parâmetros utilizados na mineração

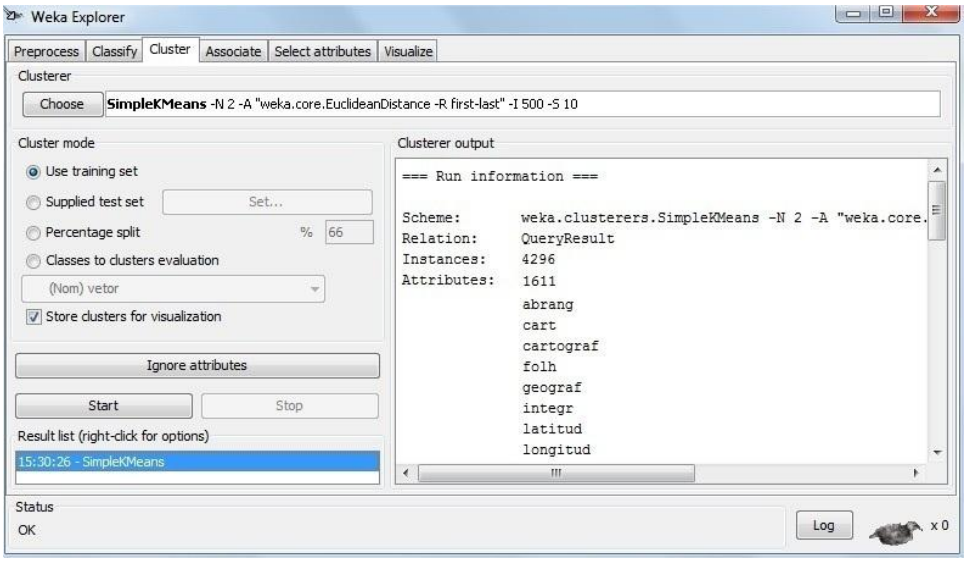


Figura 4-22 - Cabeçalho dos resultados apresentado pelo algoritmo *K-means*

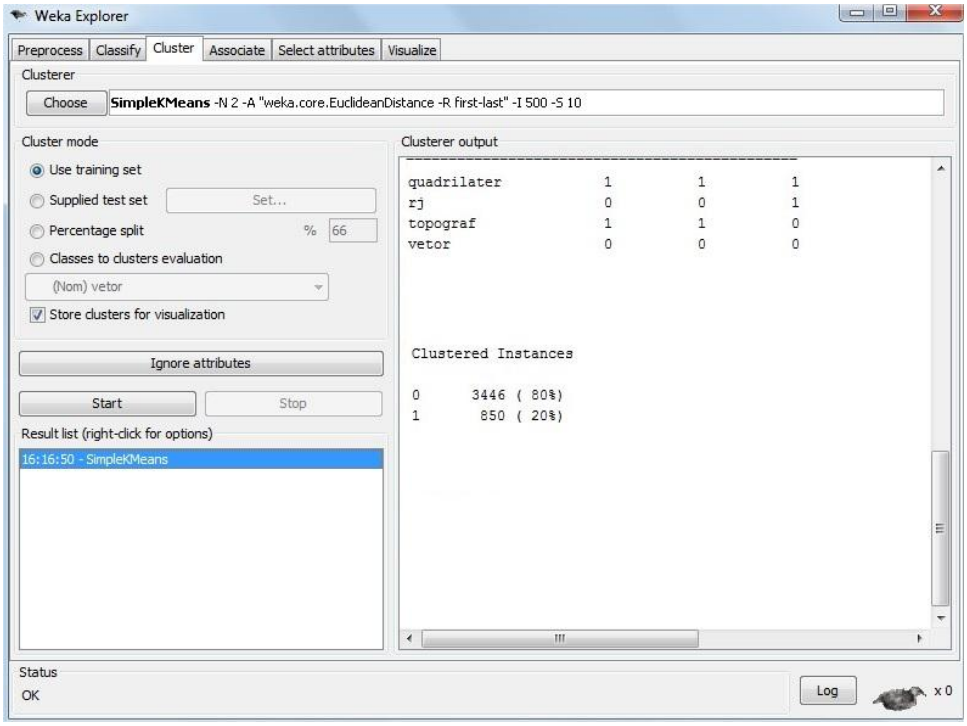


Figura 4-23 - Número e porcentagem de metadados em cada cluster com o*K-means*.

Visto esta observação, o parâmetro número de *cluster* passou a ser fundamental, já que iniciou-se a retirada do conjunto de palavras que apareciam em menor frequência nos*clusters* através dos resultados retornados por eles. A Figura 4.24 demonstra os resultados apresentados com o número de *cluster* informado em 2,4,6,8,10,20.

Percebeu-se que, em princípio, quanto menor o número de *cluster*, maior a probabilidade de encontrar metadados no mesmo *cluster* com pouca semelhança, já que o algoritmo sempre coloca o metadado no centroide mais próximo, isso faz com que uma palavra mesmo não tendo características parecidas, ficam agrupados em um mesmo *cluster*. Consequentemente a identificação das características do grupo de metadados no *cluster* se torna custosa.

Mesmo utilizando o recurso de retirada, o número de palavras ainda continuou alto. Decidiu-se então que aspalavras a serem utilizados seriam os que se apresentam com frequência mínima de 25% dos dados da base, ou seja, caso a palavra (atributo)

tenha aparecido nesta frequência dentro dos 4296 registros ela seria umapalavra escolhida. Sendo assim as fases de preparação e transformação foram acionadas novamente, fez-se as descobertas de tais palavras e criou-se a matriz binária somente com as mesmas.

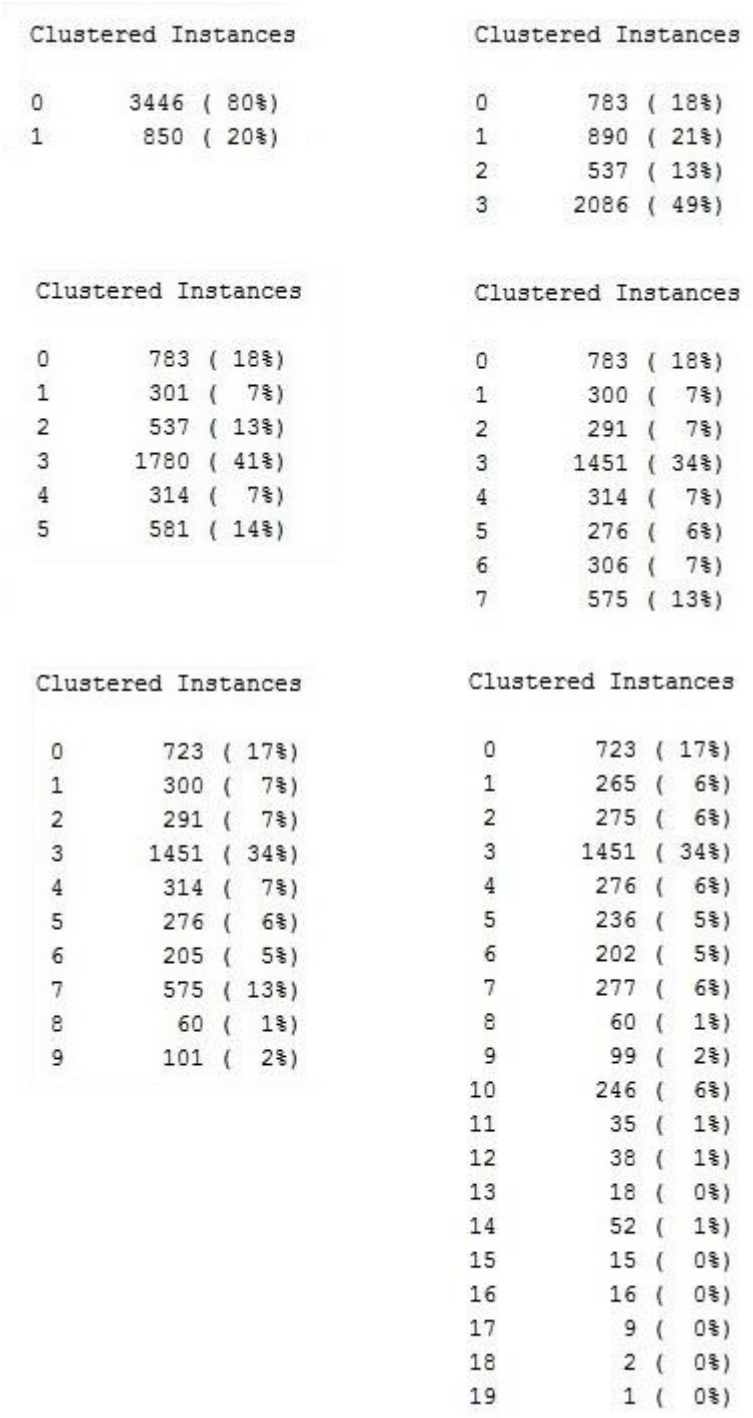


Figura 4-24 - Resultados apresentados com o número de *cluster* informado em 2,4,6,8,10,20.

A matriz binária e as 14 palavras selecionadas para mineração está ilustradana Figura4.25 e um exemplo de resultado com estaspalavras com o algoritmo *K-Means* na Figura 4.26.

id_base integer	abrang text	cart text	cartograf text	folh text	geograf text	integr text	latitud text	longitud text	matric text	mg text	quadrilater text	rj text	topograf text	vetor text
968	1	0	0	1	1	1	1	1	0	0	1	1	1	0
969	1	1	1	1	1	1	1	1	0	1	1	0	1	1
971	1	1	1	1	1	1	1	1	1	0	1	0	1	0
972	1	1	1	1	1	1	1	1	1	1	1	1	1	0
2310	1	1	1	1	1	1	1	1	0	0	1	0	1	0
2311	1	1	1	1	1	1	1	1	0	0	1	0	1	0
2312	1	1	1	1	1	1	1	1	0	0	1	0	1	0
2313	1	1	1	1	1	1	1	1	0	0	1	0	1	0
2314	1	1	1	1	1	1	1	1	0	0	1	0	1	0
2315	1	1	1	1	1	1	1	1	0	0	1	0	1	0
2316	1	1	1	1	1	1	1	1	0	0	1	0	1	0
2317	1	1	1	1	1	1	1	1	0	0	1	0	1	0
2318	1	1	1	1	1	1	1	1	0	0	1	0	1	0
2319	1	1	1	1	1	1	1	1	0	0	1	0	1	0
2320	1	1	1	1	1	1	1	1	0	0	1	0	1	0
2321	1	1	1	1	1	1	1	1	0	0	1	0	1	0
2322	1	1	1	1	1	1	1	1	0	0	1	0	1	0
2323	1	1	1	1	1	1	1	1	0	0	1	0	1	0
1	1	1	1	1	1	1	1	1	1	0	1	1	1	0
26	1	0	0	1	1	1	1	1	0	0	1	1	1	0
2	1	1	1	1	1	1	1	1	0	0	1	0	1	1

Figura 4-25 - Matriz binária com as 14 palavras selecionadas para mineração

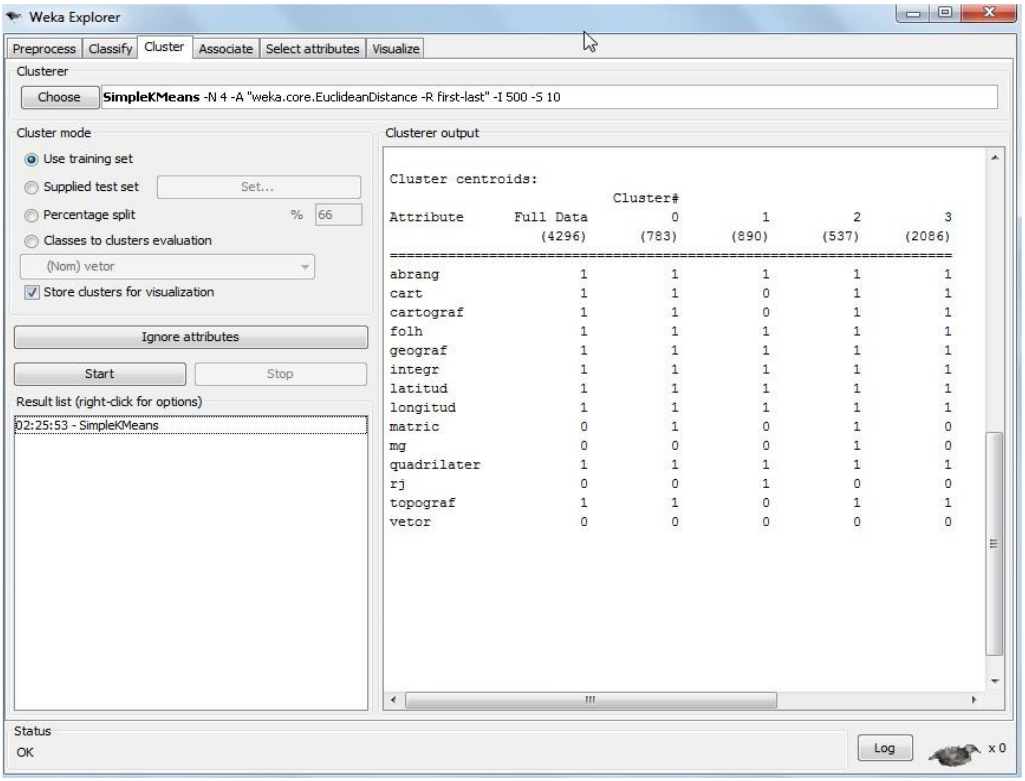


Figura 4-26 - Resultado do*K-Means*com as 14 palavras selecionadas para mineração

A partir dos resultados obtidos na fase de mineração, passou-se para a fase de visualização, análise e discussão dos resultados. A Seção 5 discorre sobre como foi realizado este processo.

5. RESULTADOS E DISCUSSÃO

A análise e visualização dos resultadosda matriz binária com as 14 palavras selecionadas na mineração foi feita em três etapas. A primeira foi identificar palavras dentro das instâncias de metadados que possuíssem potencial para serem correlacionadas com uma classe (categoria), isto é, que pudessem vir a se tornarembons classificadores para o conjunto de metadados analisado.A segunda foi analisar os comportamentos das palavras e suas relações dentro dos *clusters*, ou seja, presença ou ausência destas palavras em cada *cluster* e na terceira avaliar e classificar os agrupamentos das instâncias de metadados feitos em cada *cluster*pela weka.

Para tanto, foiutilizado o recurso de visualização da ferramenta*Weka*. Esta visualização pode ser acessada pelo painel principal na aba *Visualize* ou na lista de resultados. No trabalho, a segunda opção foi a escolhida. A Figura 5.1 ilustra o processo de seleção e a Figura 5.2 a visualização desta opção.

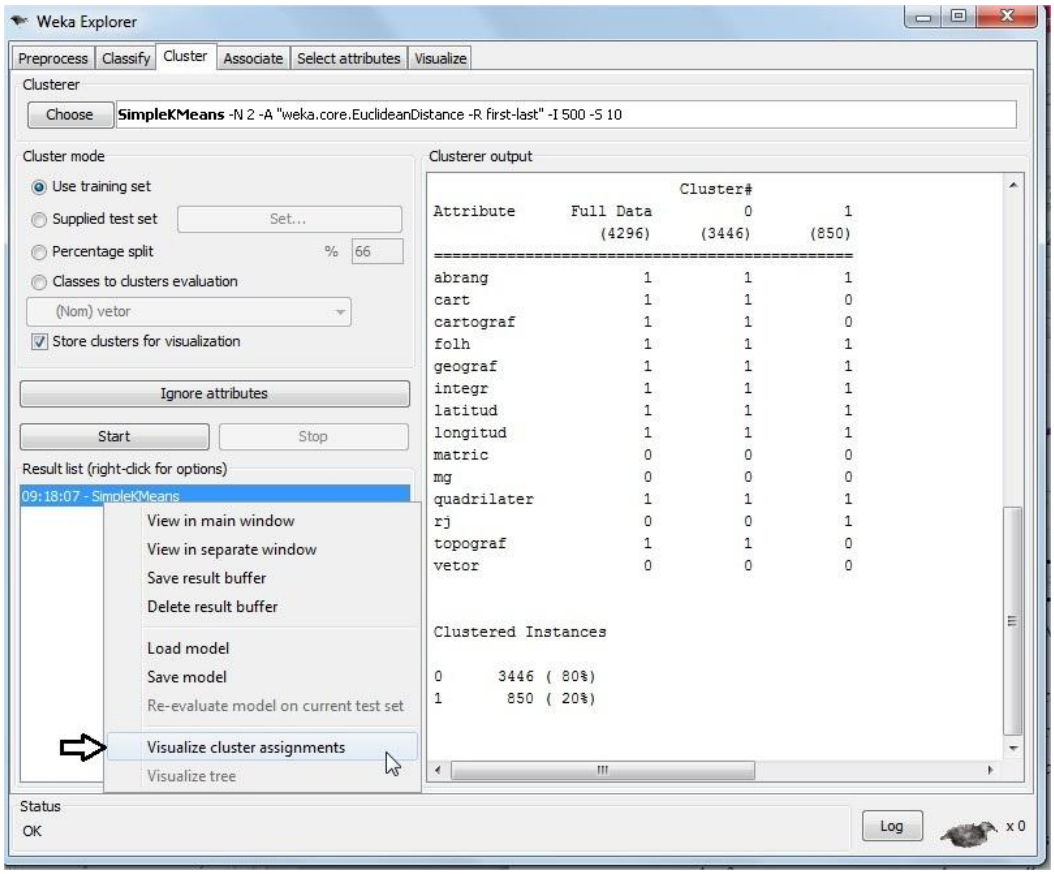


Figura 5-1 – Seleção da opção de visualização pela lista de resultados

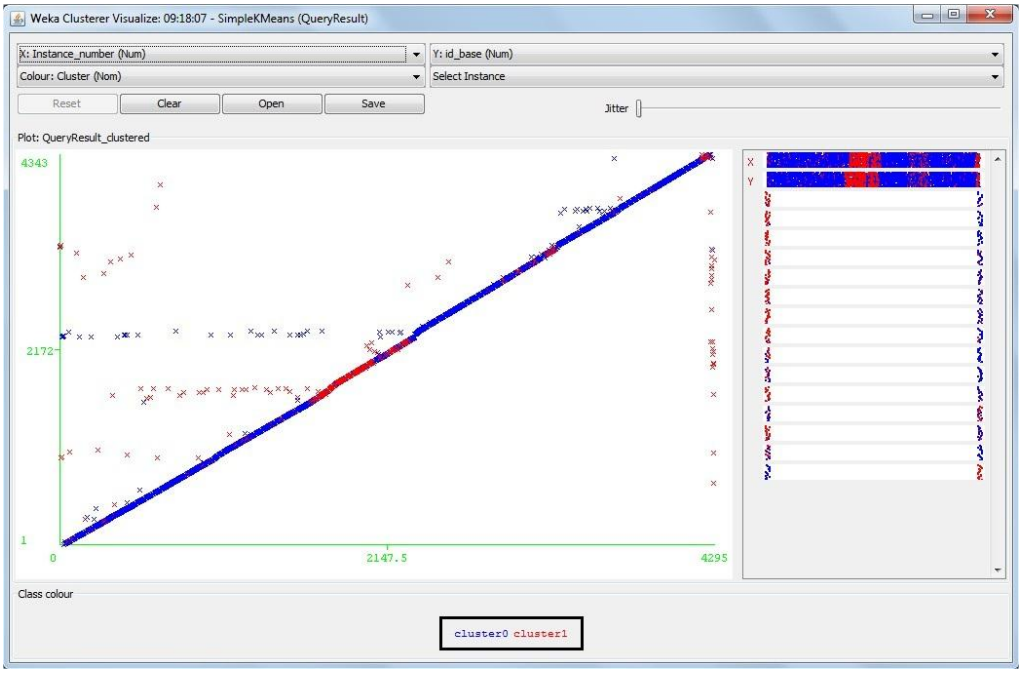


Figura 5-2 - Modo de visualização dos resultados

Na primeira etapa o *K-means*foi parametrizado da seguinte maneira: 2 *clusters*, 500 interações, 10 para o parâmetro randômico e medida de distância como a Euclidiana. A escolha do número de clusters deu-se por facilitar inicialmente a compreensão e visualização dos resultados pelo usuário. Com os resultados apresentados pelo algoritmo a ferramenta dividiu os clusters nas cores azul (canto superior esquerdo) e vermelho (canto inferior direito). A cor azul representa o cluster 0 e o vermelho representa o cluster 1. Os parâmetros do eixo X e Y foram selecionados, sendo X as informações de cluster e Y as informações das palavras (atributos) (Figura 5.3).

Após a parametrização, o zoom (*jitter*) foi ampliado para identificar visualmente a presença (1) ou ausência (0) das instâncias de metadadosem cada cluster conforme palavra selecionada. A Figura 5.4 ilustra resultados com algumaspalavras, já que as outras possuem características similares as representadas.

Na figura 5.4 pode-se observar 3 características de resultados. Nas telas 1 e 2as instâncias dos metadados são agrupadas no cluster azul quando existe a presença da palavra selecionada nas instâncias e vermelho quando há ausência. Na tela 3 percebe-se que quando há presença da palavra, as instancias podem ser agrupadas tanto no cluster

azul quanto no vermelho, da mesma forma acontece quando há ausência. Já na tela 4 observa-se que quando há presença da palavra, as instâncias são agrupadas no cluster azul, mas quando há ausência pode ser agrupada no azul ou no vermelho.

Os resultados apresentados na Figura 5.4, em especial nas telas 1 e 2, representadas, respectivamente, pelas palavras *cart* e *cartograf*, destacam-se pelas suas características. Nota-se que os *clusters* representam de forma consistente a separação da presença (1) e da ausência (0) das instâncias de metadados em cada cluster com estas palavras. Portanto, conclui-se que estas palavras denotam um potencial bom classificador deste conjunto de dados, uma vez que são boas referências para organizar os metadados. Nos outros resultados percebe-se uma dissimilaridade entre os resultados, onde em princípio não é possível fazer afirmações.

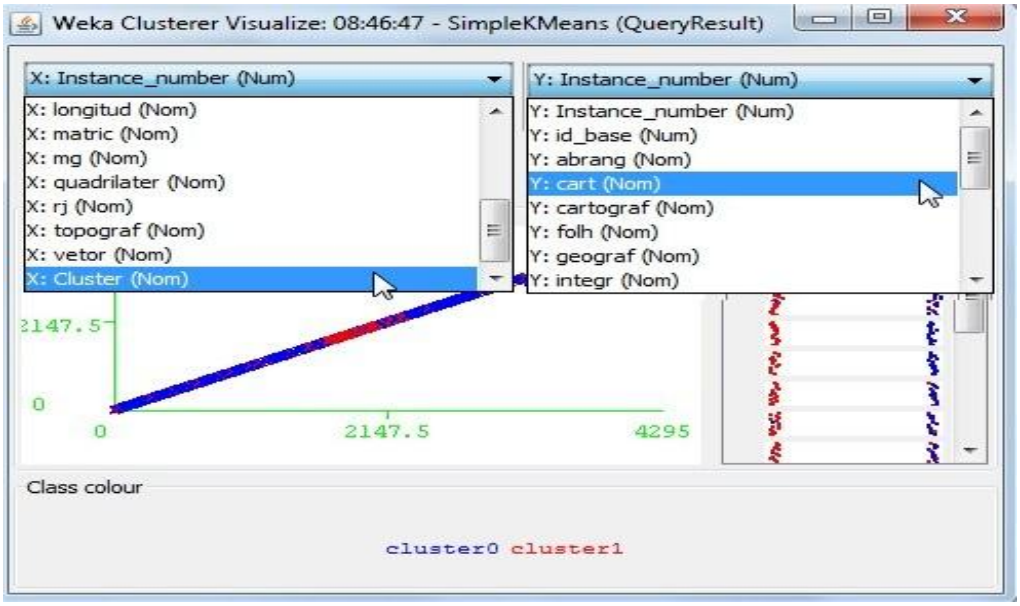


Figura 5-3 - Escolha dos parâmetros de X e Y para visualização dos resultados

Para reafirmar que aspalavras*cart* e *cartograf* são realmente bons classificadores, utilizou-se novamente o algoritmo *K-means*. Os parâmetros prévios destaetapa foram mantidos (número de *clusters*, número randômico e medida de distância) e a opção de treinamento foi alterada para "*class to clusters evalution*" para identificar o índice de erros apresentados por estas palavras quando consideradas como uma classe dentro do conjunto de metadados.

Do resultado obtido dessa nova parametrização, observou-se o percentual de instâncias clusterizadas de maneira erradapara cada classificador (Figura 5.5 e Figura 5.6), identificada pela informação de "*incorrectly clustered instances*". O índice de erros para as duaspalavras foi menor que 1%, sendo respectivamente 0.9311% e 0.9777%, demonstrando um percentual aceitável para se considerar estas palavras como possíveis classes dentro do conjunto de palavras dos metadados. Para exemplificar o resultado de um mau classificador, apalavra matric foi utilizada e observou um índice de 44.7858% (Figura 5.7).

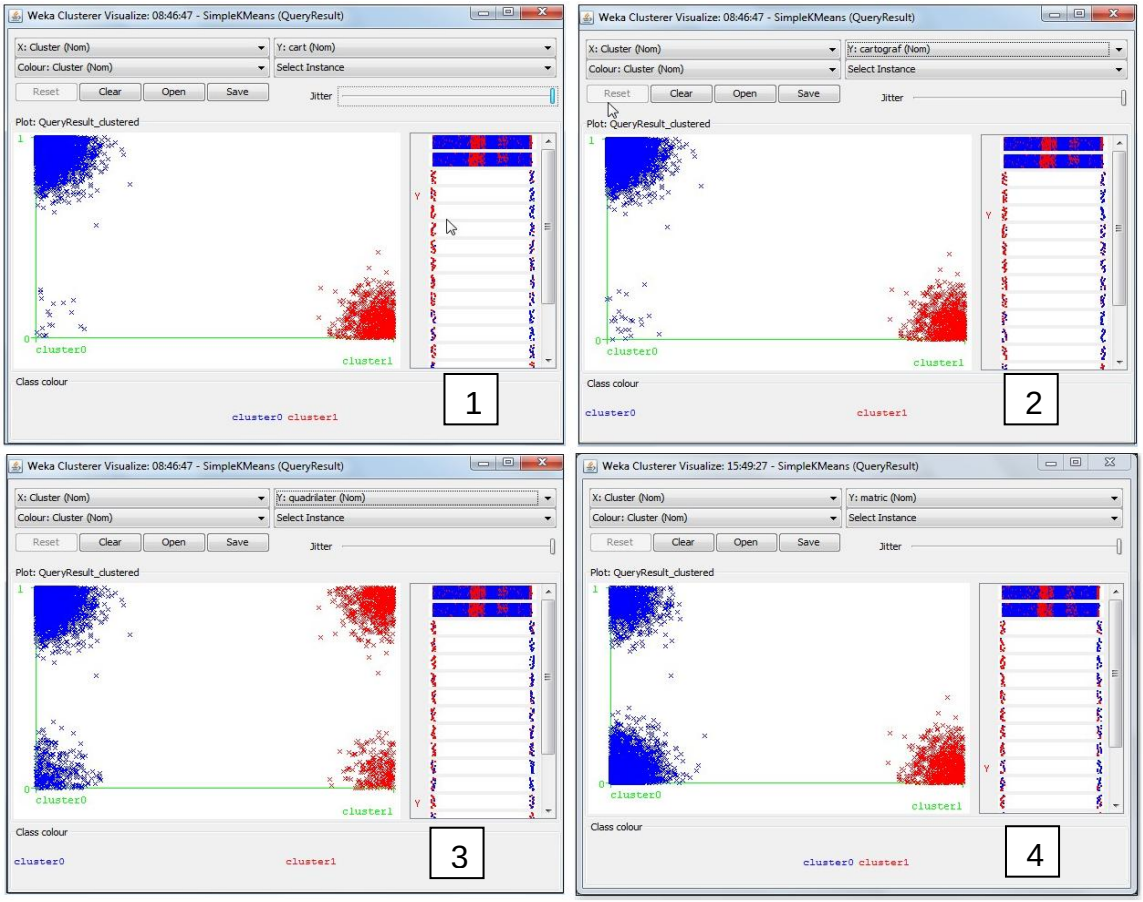


Figura 5-4 - Visualização das características dos resultados com o algoritmo *K-Means*

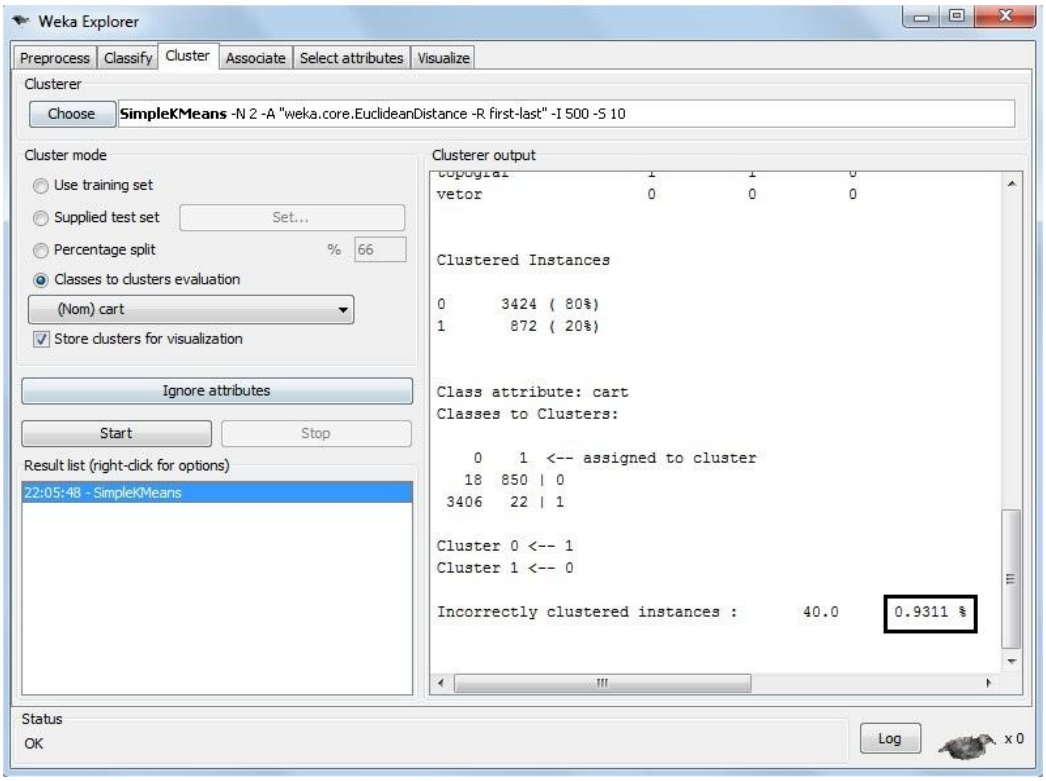


Figura 5-5 - Resultado do modo classes com a palavra *cart*

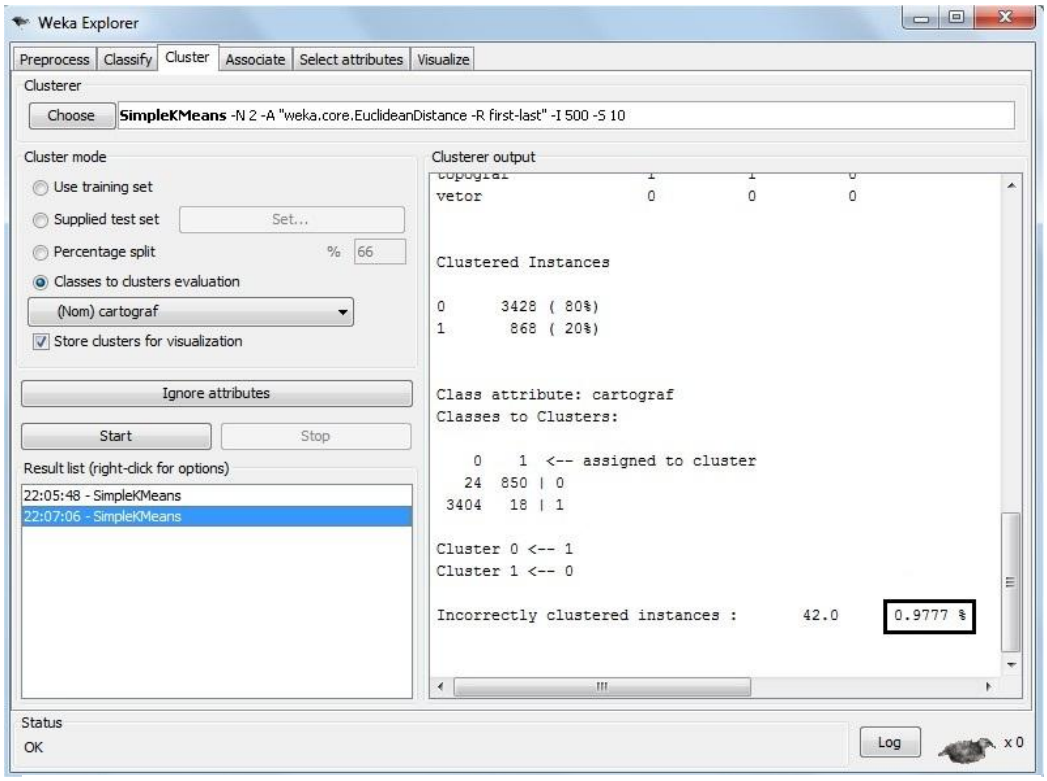


Figura 5-6 - Resultado do modo classes com a palavra *cartograf*

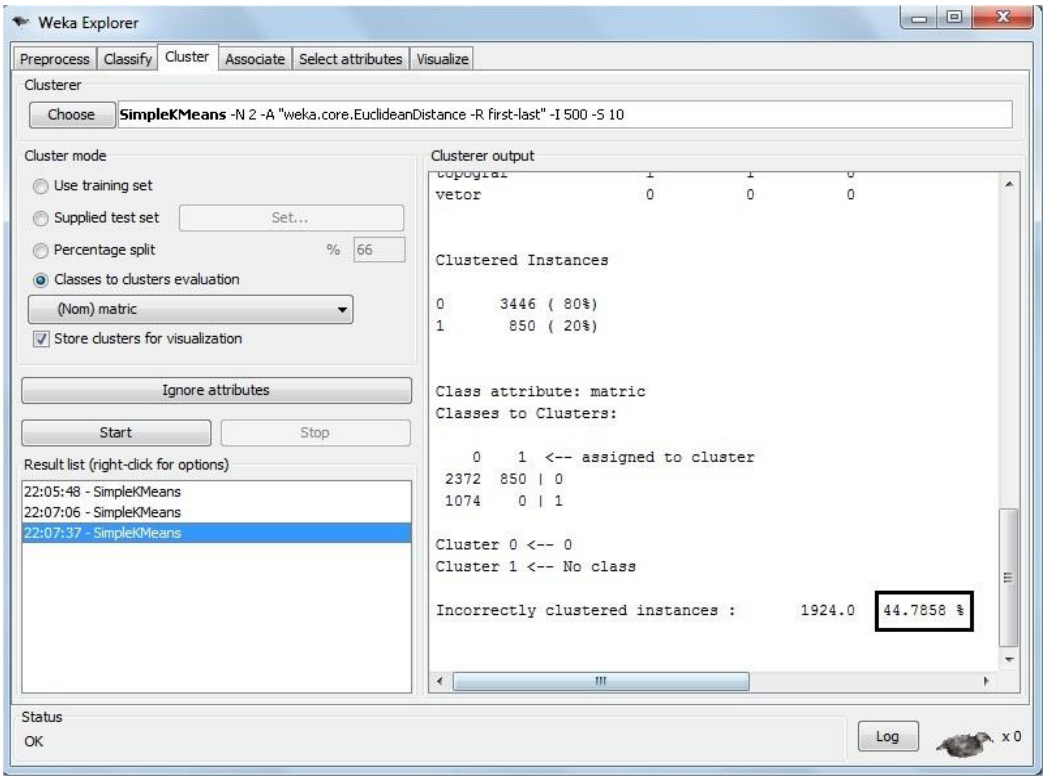


Figura 5-7 - Resultado do modo classes com a palavras*matric*

Identificado as palavras *cart* e *cartograf*dentro do conjunto de palavras como bons classificadores, a segunda etapa da análise dos resultados foi iniciada. Nesta etapa o objetivo foi analisaro comportamento daspalavras e suas relações dentro de cada *cluster* e ajudar a caracterizar as palavras classificadoras*cart* e *cartograf* .

Inicialmente, investigou-se de maneira empírica, uma possível gama de valores de K *clusters*(2,4,8,10,20)para assim identificar dentre estes valores qual seria o melhor valor deK no algoritmo*K-Means*para avaliar o comportamento das palavras classificadoras e posteriormente das outras palavras. Para isso,a taxa de erro quadráticoapresentada no resultado de cada cluster foi utilizadacomo critério de seleção para escolha do melhor número de agrupamento dos metadados dentro dos centroides, sendo esta taxa referenciada no Wekacomo "*within cluster sum of squared errors*".

Para o valor de K igual a dois *clusters*, observou-se que a taxa do erro quadrático começou alta (7149.0), diminuiu com quatro*clusters* (5724.0), alcançou o melhor resultado com oito *clusters* (699.0), subiu com 10 *clusters* (2538.0), e com 20 *clusters* (1551.0) teve-se uma queda. A tabela 5.1 demonstra os comparativos das taxas

de erro para cada K *clusters* e a Figura 5.8 ilustra o melhor resultado encontrado para o agrupamento dos metadados dentre os analisados.

Tabela 5.1 – Comparativo de taxas de erro encontrada com *clusters* 2,4,8,10 e 20

Nº de agrupamentos	Taxa de erro
2	7149.0
4	5724.0
8	699.0
10	2538.0
20	1551.0

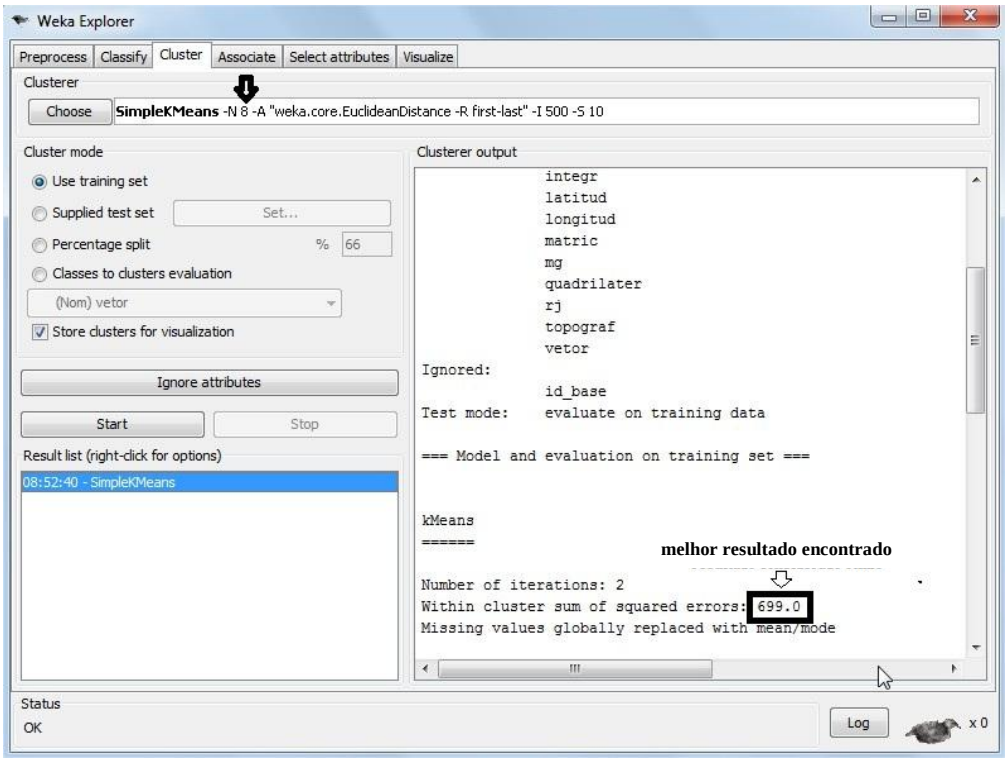


Figura 5-8 - Resultado com a menor taxa de erro quadrático

,

Encontrado o melhor número de *clusters*(8 *clusters*), o próximo passo foi analisar o comportamentodas palavras classificadoras *cart* e *cartograf*com este número de *clusters.* As Figuras 5.9 e 5.10 ilustram os comportamentos.

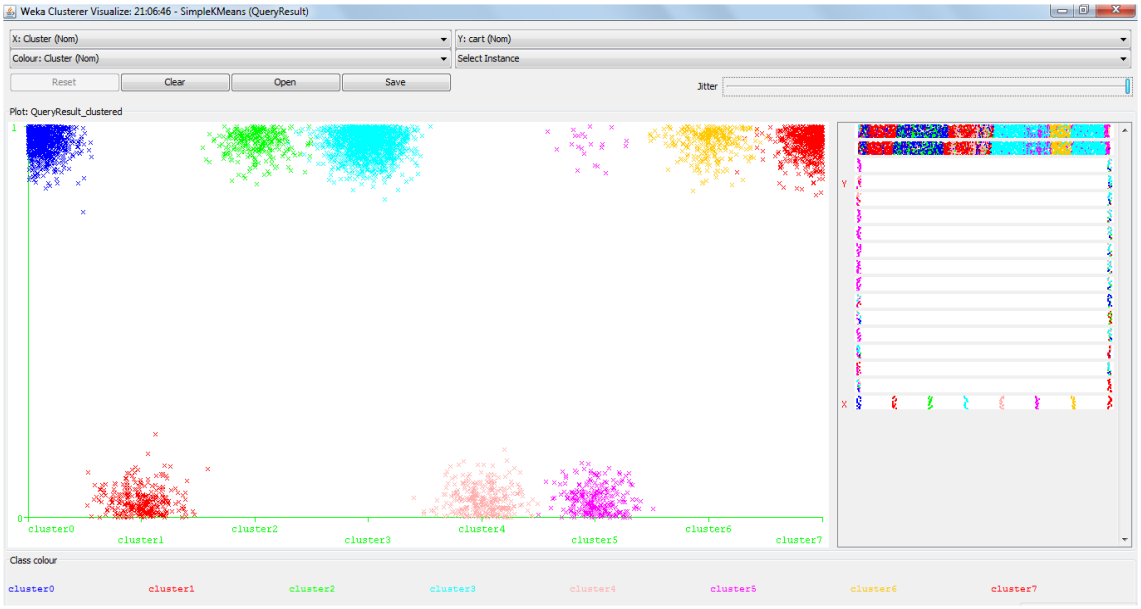


Figura 5-9 – Comportamento da palavra *cart* em cada *cluster*

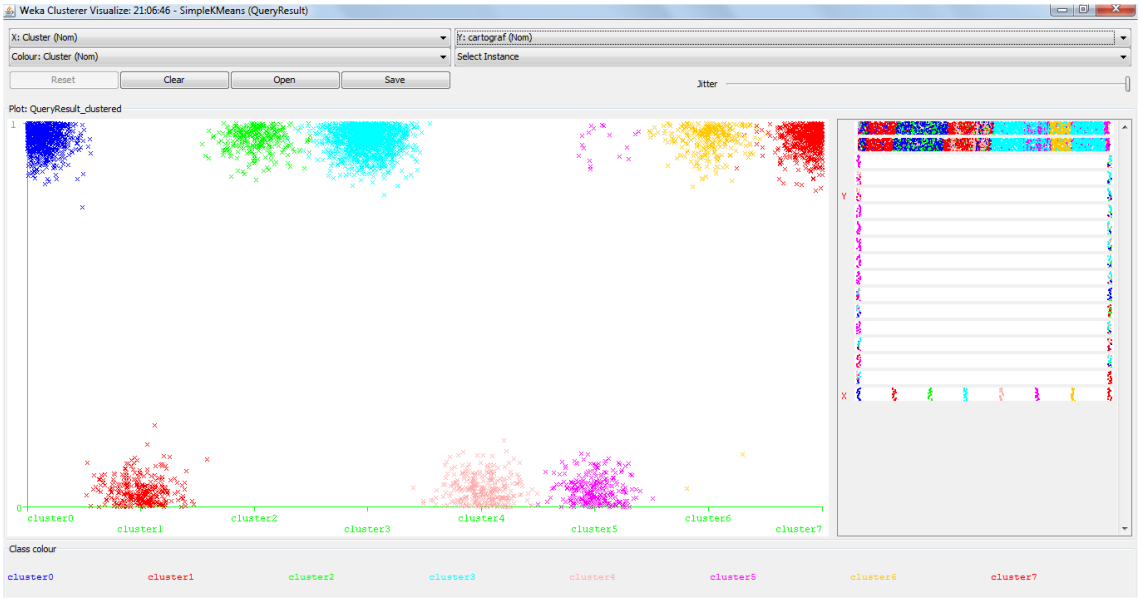


Figura 5-10 - Comportamento da palavra *cartograf* em cada *cluster*

Depois de conhecer o comportamento de *cart* e *cartograf*, verificou-se ocomportamento com as outras palavras. Para ilustrar, escolheu-se como exemplo apenas algumas palavras para demonstração de seu comportamento. As Figuras 5.11, 5.12 e 5.13 ilustram estes resultados e na Tabela 5.2 pode-se observar os comportamentos tabulados de todas palavras .

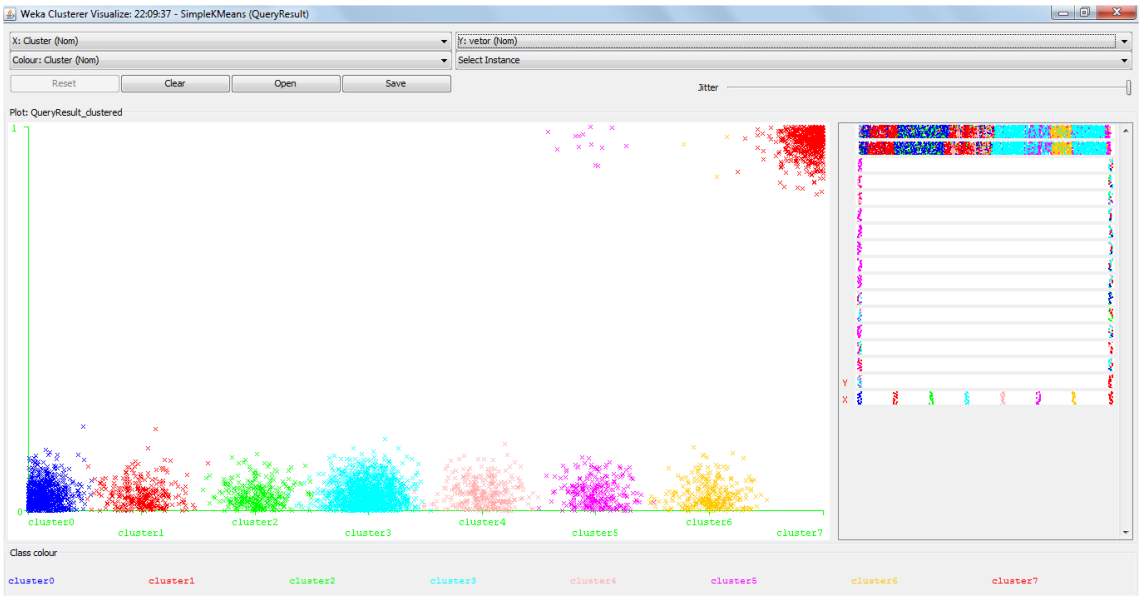


Figura 5-11 - Comportamento da palavra *vetor* em cada *cluster*

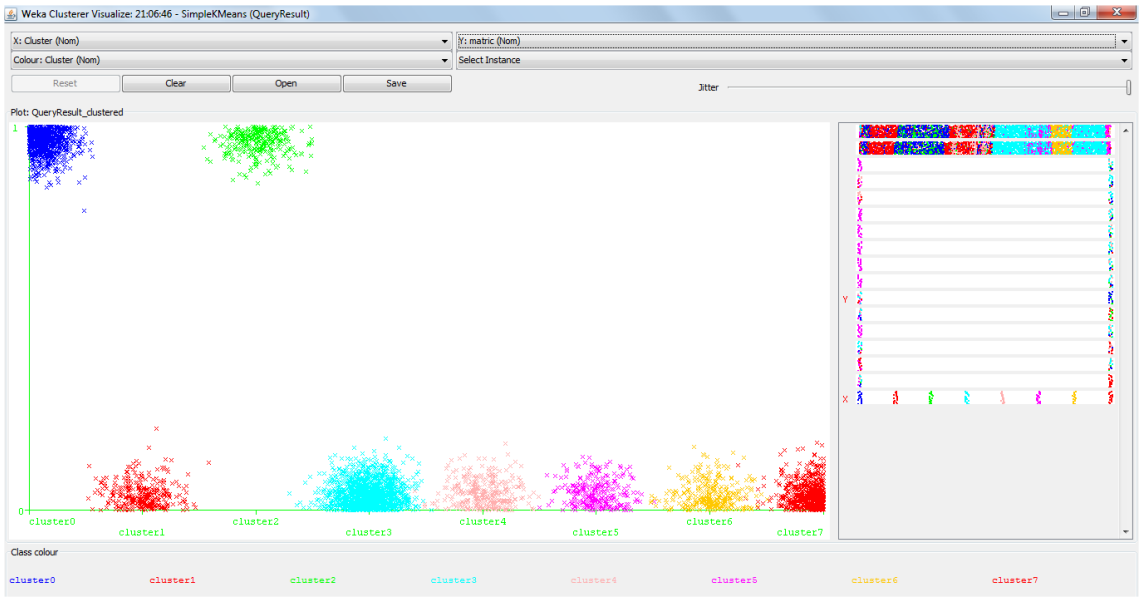


Figura 5-12 - Comportamento da palavra*matric* em cada *cluster*

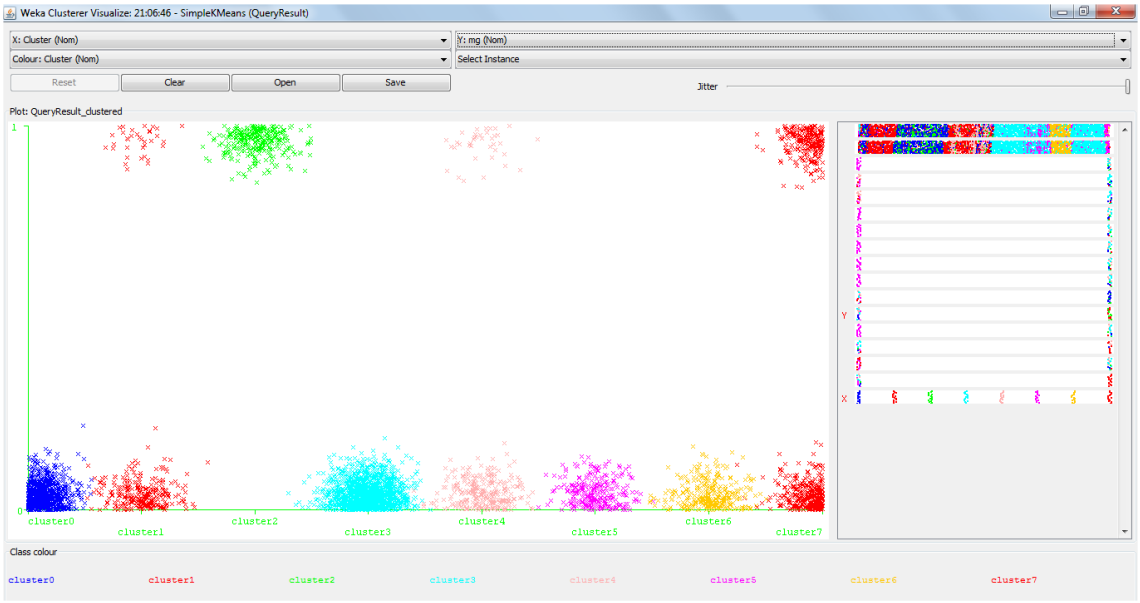


Figura 5-13 - Comportamento da palavra *mg*em cada *cluster*

Tabela 5.2 – Resultado tabulado da análise docomportamento das palavras em cada *cluster*

N°	**Atributo**	**Clusters**							
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	cart	1	0	1	1	0	0	1	1
2	cartograf	1	0	1	1	0	0	1	1
3	abrang	1	1	1	1	1	0	0	1
4	folh	1	1	1	1	1	0	0	1
5	geograf	1	1	1	1	1	0	0	1
6	integr	1	1	1	1	1	0	0	1
7	latitud	1	1	1	1	1	0	0	1
8	longitud	1	1	1	1	1	0	0	1
9	matric	1	0	1	0	0	0	0	0
10	mg	0	0	1	0	0	0	0	indefinido
11	quadrilater	1	1	1	1	1	0	0	1
12	rj	0	1	0	0	1	0	0	0
13	topograf	1	0	1	1	1	0	1	1
14	vetor	0	0	0	0	0	0	0	1

Com os comportamentostabulados, foi possível observarocomportamentode cada palavra dentro dos*clusters*. Como, por exemplo, o mesmo comportamento das palavras *abrang, folh, geograf, integr, latitud, longitude e quadrilater*, isto é, estão predominantemente presentes ou ausentes em cada um dos 8 *clusters* num mesmo padrão.

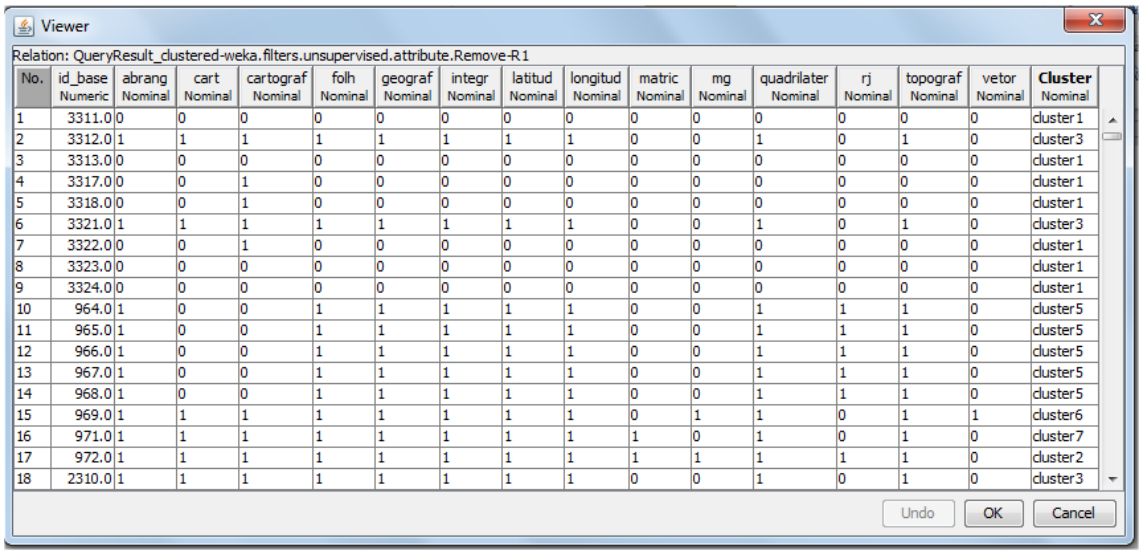
Observou-se também que não foi possível identificar de forma clara e bem separada a presença (1) ou ausência (0) da palavra *mg* no *cluster* 7 (Tabela 5.2), por isso, registrou-se como indefinido. Essa indefinição dificulta a análise dos resultados

para estapalavra em específico e pode atrapalhar outros resultados que dependem deste valor para serem analisados.

Já no*cluster* 5 observou-se que não possuía presença de nenhuma palavra. Estes resultados representam as instâncias de metadados que não apresentam nenhuma relação com as 14 palavras selecionadas para a mineração. Desta maneira, estas instâncias foram agrupadas neste *cluster*.

Concluído a análise dos comportamentos e relações das palavras passou-se para a terceira etapa. Esta etapateve como objetivo avaliar e classificar os agrupamentos das instâncias de metadados feitos pela weka em cada *cluster*.

Inicialmente, foi escolhido a palavra classificadora*cart*(poderia ser *cartograf* também devido possuir os mesmos comportamentos de *cart*) para visualizar os agrupamentos feitos em cada cluster, ou seja, identificar dentro de cada cluster quais instâncias de metadados estão relacionadas a ele. A Figura 5.14 ilustra parte das instâncias (representadas pelo campo id_base) com o cluster a que pertence (representado pelo campo *Cluster*).


Viewer

Relation: QueryResult_clustered-weka.filters.unsupervised.attribute.Remove-R1

No.	id_base Numeric	abrang Nominal	cart Nominal	cartograf Nominal	folh Nominal	geograf Nominal	integr Nominal	latitud Nominal	longitud Nominal	matric Nominal	mg Nominal	quadrilater Nominal	rj Nominal	topograf Nominal	vetor Nominal	**Cluster** Nominal
1	3311.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	cluster1
2	3312.0	1	1	1	1	1	1	1	1	0	0	1	0	1	0	cluster3
3	3313.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	cluster1
4	3317.0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	cluster1
5	3318.0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	cluster1
6	3321.0	1	1	1	1	1	1	1	1	0	0	1	0	1	0	cluster3
7	3322.0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	cluster1
8	3323.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	cluster1
9	3324.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	cluster1
10	964.0	1	0	0	1	1	1	1	1	0	0	1	1	1	0	cluster5
11	965.0	1	0	0	1	1	1	1	1	0	0	1	1	1	0	cluster5
12	966.0	1	0	0	1	1	1	1	1	0	0	1	1	1	0	cluster5
13	967.0	1	0	0	1	1	1	1	1	0	0	1	1	1	0	cluster5
14	968.0	1	0	0	1	1	1	1	1	0	0	1	1	1	0	cluster5
15	969.0	1	1	1	1	1	1	1	1	0	1	1	0	1	1	cluster6
16	971.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster7
17	972.0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	cluster2
18	2310.0	1	1	1	1	1	1	1	1	0	0	1	0	1	0	cluster3

Undo OK Cancel

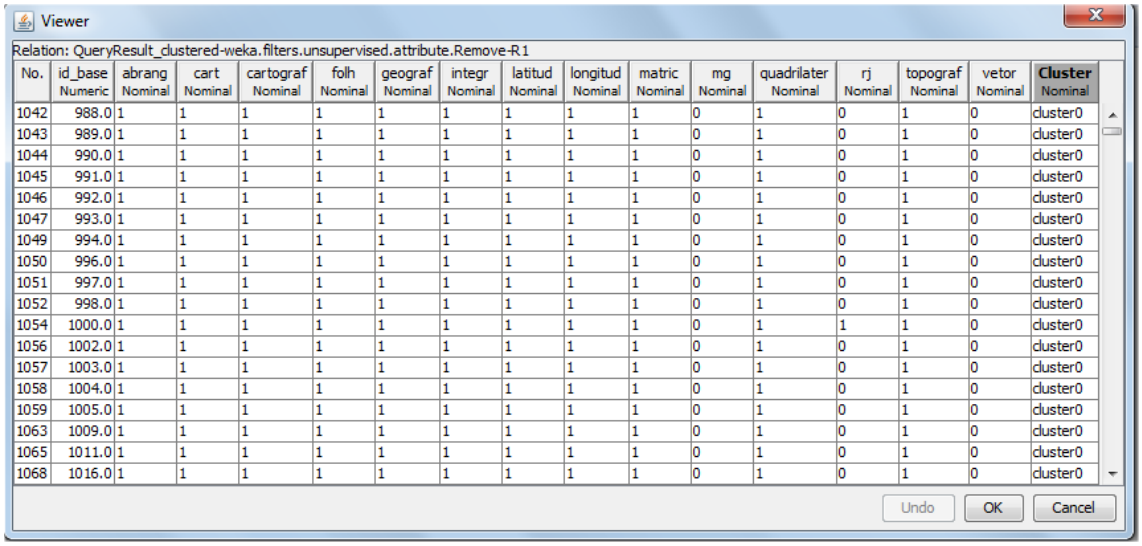
Figura 5-14 – Relacionamento de parte das instâncias de metadados com cada *cluster*

O próximo passo foi ordenar as instâncias de metadados por *cluster*(Figura 5.15) em ordem crescente para identificar todas as instâncias relacionadas aos mesmos.Em seguida, as instâncias foram utilizadas como referência para visualizar o conteúdo de outros campos dos metadadoscomo título, resumo e palavra chave. Dessa

forma estes campos foram visualizados através de consultas na base de dados e serviram de apoio para análise,avaliação e classificação dos metadados em cada cluster. A Figura 5.16 demonstra um exemplo de formato e visualização do conteúdo de algumas instâncias do *cluster* 0.

Apesar do campo título não ter sido considerado no processo de mineração, para a classificação dos metadados foi importante. Com ele foi possível fazer analogias e relacionamentos com os conteúdos apresentados nos campos resumo e palavra chave em cada *cluster* e assim ajudar na definição da classificação.

Após análise e avaliação dos conteúdos dos metadados feitos para cada *cluster*,foi possível fazer a classificação dos metadados neles representados. O resultado da classificação é demonstrado na Tabela 5.3 e discutido posteriormente.



No.	id_base Numeric	abrang Nominal	cart Nominal	cartograf Nominal	folh Nominal	geograf Nominal	integr Nominal	latitud Nominal	longitud Nominal	matric Nominal	mg Nominal	quadrilater Nominal	rj Nominal	topograf Nominal	vetor Nominal	Cluster Nominal
1042	988.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0
1043	989.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0
1044	990.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0
1045	991.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0
1046	992.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0
1047	993.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0
1049	994.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0
1050	996.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0
1051	997.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0
1052	998.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0
1054	1000.0	1	1	1	1	1	1	1	1	1	0	1	1	1	0	cluster0
1056	1002.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0
1057	1003.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0
1058	1004.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0
1059	1005.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0
1063	1009.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0
1065	1011.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0
1068	1016.0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	cluster0

Figura 5-15 - Exemplo de relacionamento das instâncias de metadados por *cluster* em ordem crescente

	id_base integer	titulo text	resumo text	palavra_chave text
1	988	Carta Topográfica Matricial 1:50.000 - TABATINGA	est folh part integr seri cart topograf	cartograf tabating sp cart topograf matric
2	989	Carta Topográfica Matricial 1:50.000 - TAIÚVA SF	est folh part integr seri cart topograf	cartograf taiuv sp cart topograf matric
3	990	Carta Topográfica Matricial 1:50.000 - TAMARANA	est folh part integr seri cart topograf	cartograf tamar pr cart topograf matric
4	991	Carta Topográfica Matricial 1:50.000 - TAQUARAL	est folh part integr seri cart topograf	cartograf taquar sp cart topograf matric
5	992	Carta Topográfica Matricial 1:50.000 - TAQUARITI	est folh part integr seri cart topograf	cartograf taquariting sp cart topograf matric
6	993	Carta Topográfica Matricial 1:50.000 - TAQUARITU	est folh part integr seri cart topograf	cartograf taquaritub sp cart topograf matric
7	994	Carta Topográfica Matricial 1:50.000 - TAUBATÉ S	est folh part integr seri cart topograf	cartograf taubate sp cart topograf matric
8	996	Carta Topográfica Matricial 1:50.000 - SERRANA S	est folh part integr seri cart topograf	cartograf serr sp cart topograf matric
9	997	Carta Topográfica Matricial 1:50.000 - SERTANÓPO	est folh part integr seri cart topograf	cartograf sertanopolil pr cart topograf matric
10	998	Carta Topográfica Matricial 1:50.000 - SEVERÍNIA	est folh part integr seri cart topograf	cartograf severin sp cart topograf matric
11	1000	Carta Topográfica Matricial 1:50.000 - SILVA JARD	est folh part integr seri cart topograf	cartograf silv jardim rj cart topograf matric
12	1002	Carta Topográfica Matricial 1:50.000 - SOMBRIO S	est folh part integr seri cart topograf	cartograf sombri sc cart topograf matric
13	1003	Carta Topográfica Matricial 1:50.000 - SUBAÚMA	est folh part integr seri cart topograf	cartograf subaum sp cart topograf matric
14	1004	Carta Topográfica Matricial 1:50.000 - SARUTAIÁ	est folh part integr seri cart topograf	cartograf sarutai sp cart topograf matric

Figura 5-16 - Exemplo de visualização do conteúdo dos metadados no *cluster* 0

Tabela 5.3 - Classificação dos metadados em cada*cluster*

Clusters	**Comportamento das palavras nos clusters**		**Classificação das instâncias de metadados nos clusters**	**Observação**
	Presença	Ausência		
0	cart, cartograf, abrang, folh, geograf, integr, latitud, longitud, matric, quadrilater, topograf	mg, rj, vetor	Carta topográfica matricial	Cartas de outros estados, exceto MG e RJ
1	abrang, folh, geograf, integr, latitud, longitud, quadrilater, rj	cart, cartograf, matric, mg, topograf, vetor	Mapas de vegetação, cartograma digital, cartas aeronáutica de pilotagem, cartas aeronáutica mundial, folha	Referências destes itens em vários estados, exceto MG
2	cart, cartograf, abrang, folh, geograf, integr, latitud, longitud, matric,mg, quadrilater, topograf	rj, vetor	Carta topográfica matricial	Cartas somente do estado de MG
3	cart, cartograf, abrang, folh, geograf, integr, latitud, longitud, quadrilater, topograf	matric, mg, rj, vetor	Carta topográfica impressa	Cartas de outros estados, exceto MG e RJ
4	abrang, folh, geograf, integr, latitud, longitud, quadrilater, rj, topograf	cart, cartograf, matric, mg, vetor	Modelo digital de elevação	Modelos somente do estado do RJ
5		cart, cartograf, abrang, folh, geograf, integr, latitud, longitud, matric, mg, quadrilater ,rj, topograf, vetor	Ortofoto	Ortofotograma de outros estados, exceto MG e RJ
6	cart, cartograf, topograf	abrang, folh, geograf, integr, latitud, longitud, matric, mg, quadrilater, rj, vetor	Carta topográfica vetorial	Cartas de outros estados, exceto MG e RJ
7	cart, cartograf abrang, folh, geograf, integr, latitud, longitud, quadrilater, topograf, vetor	matric, rj, mg	Carta topográfica matricial	Cartas de outros estados, exceto RJ e MG (indefinido)

Analisando os resultados da Tabela 5.3percebe-seque os comportamentos das palavras quando analisados de maneira conjunta ajuda na descoberta da classificação dos metadados e quando analisado de maneira isolada dificulta esta descoberta.

Um exemplo pode ser visto com as palavras *cart, cartograf, abrang, folh, geograf, integr, latitud, longitud, quadrilater, topograf* quando estão presentes de maneira conjunta no *cluster* pode-se dizer que ajudam a caracterizar uma carta topográfica conforme demonstrado nos *clusters* 0, 2, 3 e 7.

Outro exemplo encontra-se comas palavras *abrang*, *folh*, *geograf*, *integr*, *latitud*, *longitud*, *quadrilater*, quando estão presentes de maneira conjunta e as palavras *cart* e *cartograf* estão ausentes no *cluster*, pode-se afirmar que ajudam a caracterizar mapas de vegetação, cartograma digital, cartas aeronáutica de pilotagem, cartas aeronáutica mundial, folha, modelo digital de elevação, exceto, carta topográfica conforme demonstrado nos *clusters* 1 e 4.

Já a ausência em conjunto de todas as palavras no *cluster*, caracteriza a classificação de um Ortofoto conforme apresentado no *cluster* 5, o que demonstra que não existe nenhum tipo de relação destas palavras com as que a caracterizam.

Analisando o comportamento das palavras *vetor* e *matric* de forma isolada nos *clusters* 6 e 7 respectivamente, percebe-se que as classificações feitas não seriam corretas devido ausência da palavra no *cluster*, mas, ao avaliar o conteúdo e o conjunto de comportamentos de forma conjunta com as outras palavras, foi possível definir a classificação como carta topográfica vetorial e carta topográfica matricial.

A observação feita na palavra *mg* no *cluster* 7, deu-se devido seu comportamento definido na Tabela 5.2 como indefinido. Ao realizar o processo de classificação percebeu-se que o percentual de sua presença dentro desse cluster foi baixa, mas não inexistente, por isso foi colocado como indefinido no campo observação.

Estes foram os principais fatores que se destacaram ao analisar as instâncias de metadados dentro de cada *cluste*r ajudaram na definiçãoda classificação.Com base nos resultados e discussões apresentadas aqui, na Seção 6 são tiradas algumas conclusões a respeito do conhecimento que pode ser obtido do trabalho.

6. CONCLUSÕES

Com base nos estudos e nas pesquisas realizadas, também conforme referencial bibliográfico, este trabalho propôs um estudo da mineração semântica de metadados em IDE. Esse estudo descreve um conjunto de etapas, em que a base de metadados de uma IDE é criada e evolui por meio das etapas do processo de KDD. Sob esta perspectiva, o objetivo geral dessa dissertação foi alcançado.

Para alcançar os objetivos específicos propostos, a experimentação da mineração proposta no Capítulo 4 foi aplicada, por meio das etapas de KDD, o que possibilitou demonstrar na prática os passos para a descoberta do conhecimento em uma base de metadados sobre uma IDE.

Apesar da base possuir muitos metadados de natureza parecida dificultando a descoberta de conhecimento útil, foi possível extrair relações e conhecimento, como, por exemplo, descobrir as palavras classificadoras dos metadados, as relações do conjunto de outras palavras associadas a estas e as categorias (classes) dos conjuntos de metadados em cada cluster.

No trabalho, é possível identificar as palavras “cart” e “cartograf” como classes relevantes, de fácil identificação e, em seguida, discriminar, também de forma automática e com baixa expectativa de erro as características de seus *clusters* dentro da base de metadados identificando quais conjuntos de metadados pertencem eles.

Os resultados obtidos tanto no processo de *cluster*ização como no processo de associação estão intimamente ligados à qualidade dos dados. No entanto, alguns problemas podem ser diminuídos e até contornados pela utilização de técnicas auxiliares às tarefas principais, como foi mostrado nesta pesquisa. Foi possível observar também que com o emprego do *stemming*, melhora-se a qualidade do processo de mineração.

O uso de ferramenta Wekanamineração de dadosajudouna descoberta de conhecimentos; auxiliou no processo de exploração dos dados contidos na base, permitindo a seleção e geração de informações importantes para as tomadas de decisões; facilitou a análise e interpretação dos resultados durante o processo dos testes; contribuiu com a possibilidade de realizar vários testes em pouco espaço de tempo e deu suporte a diversos algoritmos de aprendizagem de máquina.

Como principais contribuições desta dissertação, pode-se citar:

- Considerações a respeitodos desafios inerentes às etapas do processo de KDD, das tarefas de regras de associação e clusterização, quando aplicadas à mineração de bases de dados de metadados;
- Ilustrar uma metodologia para mineração semântica de metadados em uma IDE para representação do conhecimento e suas dificuldades;
- Estudo inicial para prover apoio para categorizar base de metadados e documentos textuais, destacando pontos que merecerão atenção de pesquisadores futuros.

Esta dissertação admite extensões que podem contribuir para o desenvolvimento desta linha de pesquisa. As sugestões de trabalhos futuros estão relacionadas tanto aos aspectos de pesquisa quanto da implementação.Algumas das quais estão listadas a seguir:

- Aplicar os mesmos passos aqui desenvolvidos em outras bases de dados;
- Desenvolver testes comparativos utilizando metadados de outrasIDEs;
- Comparar o desempenho de outros algoritmos para mineraçãode *clusters*;
- Criar um método classificador baseado nos resultados obtidos pela dissertação;
- Implementar uma interface para realizaçãoda criação e transformação dos metadados para regras de associação e clusterização.

REFERÊNCIAS BIBLIOGRÁFICAS

ADITYA, T.; KRAAK,M.J. A search interface for an SDI: implementation and evaluation of metadata visualization strategies. **Transactions in GIS**, v. 11, n. 3,p. 413-435, 2007.

AGRAWAL, R; SRIKANT, R. Fast Algorithms for Mining Association Rules. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATABASES (VLDB) CONFERENCE. 20th., 1994,Santiago. **Proceedings..** Chile, 1994. p. 487-498.

ALVARES, R.V.; GARCIA, A.C.B.; FERRAZ, I. STEMBR: a stemming algorithm for the Brazilian Portuguese Language. In: PORTUGUESE CONFERENCE ON ARTIFICIAL INTELLIGENCE, 12.,2005. **Proceedings..** Covilhã, Portugal.

ALVES, R. C. V. **Web Semântica: uma análise focada no uso de metadados**. 2005. 180 f. Dissertação (Mestrado em Ciência da Informação)-Faculdade de Filosofia e Ciências, Universidade Estadual Paulista. Marília, 2005.

ARANHA CHRISTIAN ; PASSOS, E. P. L. . A Tecnologia de Mineração de Textos. RESI. **Revista Eletrônica de Sistemas de Informação**, v. 2, p. 2, 2006.

AZEVEDO, M. I. M. **Recuperação de Informação em Documentos XML**. 2005. 103 f. Dissertação (Mestrado em Ciência da Computação) -Universidade Federal de Minas Gerais, Belo Horizonte.2005.

BARION. E.C.N; LAGO. D. Mineração de textos. **Revista de Ciências Exatas e Tecnologia**. v.3, n.3, p.123-140, 2008.

BERNARD, L.; CRAGLIA, M. SDI: from spatial data infrastructure to service driven infrastructure. In: RESEARCH WORKSHOP ON CROSS-LEARNING BETWEEN SPATIAL DATA INFRASTRUCTURES (SDI) AND INFORMATION INFRASTRUCTURES, 2., 2005. **Proceedings…** Enschede, Holanda, 2005.

CARLANTONIO, L.M.di. **Novas metodologias para clusterização de dados**. 2001. 148f. Dissertação (Mestrado em Engenharia Civil) - Universidade Federal do Rio de Janeiro, Rio de Janeiro.2001.

CHENG, J; KE, Y.; NG, W. **Effective elimination of redundant association rules**. Data Mining and Knowledge Discovery, v. 16, p. 221-249, 2008.

COELHO, A. R. **Stemming para a língua portuguesa: estudo, análise e melhoria do algoritmo RSLP**. 2007. 69f. Monografia (Graduação em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, Porto Alegre.2007.

COLE, R. M. **Clustering with genetic algorithms**. 1998. 110 f. Dissertação (Mestrado em Ciência da Computação) - Department of Computer Science, University of Western Australia, [s.l.], 1998.

CONCAR. **Perfil de Metadados Geoespaciais do Brasil**. Disponível em: < http://www.concar.ibge.gov.br/arquivo/Perfil_MGB_Final_v1_homologado.pdf>. Acesso em: 8 out. 2011.

DAMASCENO, F. R.; RIBEIRO, A.; REATEGUI, E. **Aplicação educacional de uma ferramenta de mineração de textos integrada a uma ontologia de domínio na área da saúde.** Disponível em: <http://seer.ufrgs.br/renote/article/download/21912/12713>. Acesso em: 11 ago. 2011.

DAVIS JR., C. A.; ALVES, L. L. Local spatial data infrastructures based on a serviceoriented architecture. In: BRAZILIAN SYMPOSIUM ON GEOINFORMATICS. **Proceedings**... [s.l. : s.n.], 2005. p. 30-45.

DAVIS JR., C.A.; FONSECA, F.T.; CÂMARA, G. Infraestruturas de dados espaciais na integração entre ciência e comunidades para promover a sustentabilidade ambiental. In: WORKSHOP DE COMPUTAÇÃO APLICADA À GESTÃO DO MEIO AMBIENTE E RECURSOS NATURAIS, 1., 2009, Bento Gonçalves. **Proceedings...**Porto Alegre: SBC, 2009.

DEVILLERS, R. *et al*. **Spatial data quality: from metadata to quality indicators and contextual end-user manual**. Disponível em: <http://www.cassini.univ-mrs.fr/publis/OEEPE_ISPRS_Devillers.pdf >. Acesso em: 20 mai 2011.

DUBLINCORE. **Dublin Core Metadata Initiative**. Disponível em<http://dublincore.org/metadata-basics/>. Acesso em: 20 out. 2011.

ESCOVAR, E.L.G. **Algoritmo SSDM para a mineração de dados semanticamente similares**. 2004. 86 f.Dissertação (Mestrado em Ciência da Computação), Universidade Federal de São Carlos, São Carlos.2004.

FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P. **From Data Mining to Knowledge Discovery in Databases**. Artificial Intelligence Magazine, v. 17, n. 3, p. 37-54, 1996.

FEDERAL GEOGRAPHIC DATA COMMITTEE - FGDC. **Content standard for digital geospatial metadata**. Washington: D.C.: Federal Geographic Data Committee, 1998. 78 p.

FONTANA, A.; NALDI, M. C. **Estudo de comparação de métodos para estimação de números de grupos em problemas de agrupamento de dados**. Disponível em: < http://www.icmc.usp.br/~biblio/BIBLIOTECA/rel_tec/RT_340.pdf>. Acesso em: 20 out. 2011.

FRAKES, W.B.; BAEZA-YATES, R. **Information retrieval: data structures algorithms**.New Jersen: Prentice Hall. 1992.

FRAWLEY W.; PIATETSKY-SHAPIRO G.; MATHEUS C. Knowledge Discovery in Databases: an Overview. **AI Magazine**, p.213-228, 1992.

GOLDSCHMIDT, R.; PASSOS, E. **Data Mining: um guia prático**. Rio de Janeiro: Elsevier, 2005.

HALL, M.*et al*.**The w*eka* data mining software: an update**. Disponível em: < http://www.kdd.org/explorations/issues/11-1-2009-07/p2V11n1.pdf>. Acesso em: 10 out. 2011.

HAN, J.; KAMBER, M. **Data Mining: Concepts and Techniques**. Morgan Kaufmann, San Francisco, CA, 2001.

HAN, J.; KAMBER, M. **Data Mining: Concepts and Techniques**. 2.ed. Morgan Kaufmann, San Francisco, CA, 2006.

HAMED, O.; RAJABIFARD, A.; KALANTARI, M. **Automatic spatial metadata update: a new approach**.Disponível em: <www.fig.net/pub/fig2010/.../ts05b%5Cts05b_olfat_rajabifard_et_al_4079.pdf>. Acesso em: 25 nov. 2010.

ISO 19115. **Geographic information -** Metadata. Geneva: International Organizationfor Standardization (ISO), 2003.

ISO 19139. **Geographic information –** Metadata XML. Geneva: International Organizationfor Standardization (ISO), 2007.

INFRASTRUCTURE FOR SPATIAL INFORMATION IN THE EUROPEAN COMMUNITY - INSPIRE. European parliament and the european council. **Directive of the European Parliament and of the Council establishing an Infrastructure for Spatial Information in the European Community**.Disponível em: <http://register. consilium.europa.eu/pdf/en/06/st03/st03685.en06.pdf>. Acesso em: 8 out. 2010.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. **ACM computing surveys**, v. 31, p. 264-322, 1999.

LOPES, M.C.S. **Mineração de dados textuais utilizando técnicas de clustering para o idioma português**. 2004. 180f. Dissertação(Doutorado em Engenharia Civil)- Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2004.

MANSO-CALLEJO M.; WACHOWICZ M.; BERNABÉ-POVEDA M. **Automatic Metadata Creation for Supporting Interoperability Levels of Spatial Data Infrastructures**. Disponível em: <www.gsdi.org/gsdi11/papers/pdf/194.pdf>. Acesso em: 15 abr. 2011.

MELANDA,E.A. **Pós-processamento de regras de associação**. 2004. 183f. Dissertação (Doutorado em Ciências da Computação e Matemática Computacional)– Universidade de São Paulo, São Paulo, 2004.

MITRA, S. *et al.*. Data mining in soft computing framework: **IEEE Transactions on Neural Networks**, v.13, p. 3-4, 2002.

MONTEIRO, L.O *et al*. Etapas do Processo de Mineração de Textos – uma abordagem aplicada a textos em Português do Brasil. In: WORKSHOP DE COMPUTAÇÃO E APLICAÇÕES (WCOMPA), 26., **Anais:** Campo Grande, Mato Grosso do Sul, p.78-81, 2006.

NEBERT, D.D. **Developing spatial data infrastructures**: the SDI Cookbook, version 2.0, 2004. (GSDI-Technical Working Group). Disponível em: <www.gsdi.org/ docs2004/Cookbook/cookbookV2.0.pdf>.

OLIVEIRA,T.B.S. **Clusterização de dados utilizando técnicas de redes complexas e computação bioinspirada**. 2008. 91f. Dissertação (Mestrado em Ciências da Computação e Matemática Computacional) –Universidade de São Paulo, São Carlos. 2008.

ORENGO, V.M; HUYCK, C. A stemming algorithm for the portuguese language. In: INTERNATIONAL SYMPOSIUM ON STRING PROCESSING AND INFORMATION RETRIEVAL (SPIRE), 8., **Anais:** Laguna de San Raphael, Chile, p.183-193, 2001.

PITONI, R. M. **Mineração de Regras de Associação nos Canais de Informação do Direto**. 2002. 61 f. Monografia (Graduação em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, Porto Alegre. 2002.

PORTER, M.F. **The stemming algorithm**.2005. Disponível em: <http://tartarus.org/~martin/PorterStemmer/>. Acesso em: 10 set. 2011.

PRADO; B. *et al.*. Padrões para metadados geográficos digitais: modelo ISO 19115:2003 e modelo FGDC. **Revista Brasileira de Cartografia**, v. 62, p. 1, 2010.

RAVIKUMAR, K.; GNANABASKARAN, A. **ACO based spatial data mining for traffic risk analysis.** Disponível em: <www.bioinfo.in/uploadfiles/12692391721_1_2_IJCI.pdf>.Acesso em: 19 set. 2010.

RAJABIFARD, A.; WILLIAMSON, I.P. Spatial data infrastructures: concept, SDI hierarchy and future directions. In: GEOMATICS'80, 2001, Tehran. **Proceedings...** Tehran, Iran, 2001.

SFERRA, H.H; JORGE CORREA. A.M.C. Conceitos e aplicações de data mining. **Revista de Ciência e Tecnologia**, v.11, n.22, p.19-34, 2003

SANTOS, P.L.V.A.C; ALVES, R.C.V. Metadados e Web Semântica para estruturação da Web 2.0 e Web 3.0. **Revista de Ciência da Informação**, v. 10, p. 6, 2009.

SILVA, H. MIG - **Metadados para Informação Geográfica**. Instituto Geográfico Português, Lisboa, Portugal. 2007.

SILVA, O.C.; LISBOA FILHO, J.; BRAGA, J. L.; BORGES, K. A. V. **Searching for metadata using knowledge base and topic maps in Spatial Data Infrastructures**. Earth Science Informatics, Springer, v.2, n.4, p.235-247, 2009**.**

WU,X.*et al*.Top 10 algorithms in data mining. **Knowledge and InformationSystems,**v. 14, p. 1-37, 2008.

UNIVERSIDADE OF SHEFFIELD (USFD). **Infra-estruturas de Dados Espaciais: Recomendações de actuação. Rede Européia de Informação Geográfica**. 2000. Disponível em: <http://www.ec-gis.org/ginie/doc/PG_SDI_pt.pdf>. Acessado em: 13 mar. 2010.

XU, R.; WUNSCH, D. Survey of clustering algorithms. **IEEE Transactions on neural networks**, v. 16, n.3, p.645-678, 2005.

ZHANG, C.; ZHANG, S. **Association Rules Mining: Models and Algorithms**. Springer-Verlag Berlin Heidelberg, 2002.