UNIVERSIDADE FEEVALE

PABLO FREDERICO OLIVEIRA THIELE

EXTRAÇÃO DE MAPAS CONCEITUAIS A PARTIR DE TEXTO TÉCNICO

Novo Hamburgo
2011

PABLO FREDERICO OLIVEIRA THIELE

EXTRAÇÃO DE MAPAS CONCEITUAIS A PARTIR DE TEXTO TÉCNICO

Universidade Feevale

Instituto de Ciências Exatas e Tecnológicas

Curso de Ciência da Computação

Trabalho de Conclusão de Curso

Professor Orientador: Ricardo Ferreira de Oliveira

Novo Hamburgo
2011

AGRADECIMENTOS

Primeiramente à minha família pelo apoio que sempre recebi. Minha noiva Juliana que sempre me ajudou e foi muito importante para que chegasse até aqui. Agradeço também o professor, orientador e amigo Ricardo Ferreira pelo apoio neste trabalho.

RESUMO

Uma abordagem eficiente de ensino utiliza diversas ferramentas que colaboram na educação dos alunos. As ferramentas de ensino que prezam uma aprendizagem construtivista possuem uma ampla gama de opções. Dentre essas, temos os mapas conceituais, mapas gráficos onde ideias complexas são apresentadas ordenando-se hierarquicamente conceitos e subconceitos de um referido tema. Sabendo que o ser humano consegue aprender mais e de forma mais contundente fazendo-se valer de meios visuais e não apenas texto, este tipo de ferramenta é especialmente útil para educadores. Na questão de colaborar tanto na criação quanto na validação de um mapa conceitual, esse projeto tem como objetivo gerar um software que a partir de um texto técnico, gere um mapa conceitual adequado, sendo esse utilizado como base de estudo ou comparações. Essa abordagem é complexa, devido à necessidade de se utilizar algoritmos de análise léxica e semântica focados na questão de processamento de linguagem natural, para que os conceitos dos textos possam ser identificados e posteriormente organizados da maneira mais correta possível.

Palavras-chave: Mapas conceituais. Método de Ensino. Aprendizagem construtivista.

ABSTRACT

An efficient teachings approach uses several teaching tools that help in the education of students. The instruction tools that cherish a constructivist learning possess a wide range of options. Among these tools, we have the concept maps, graphs where complex ideas are presented as hierarchically ordered concepts and sub-concepts of one such theme. Knowing that humans can learn more and better by making recourse to visual media and not just text, this tool type is especially useful for educators. On the issue of working both in creation and validation of a conceptual map, this project aims to generate software that from a technical text, create an appropriate conceptual map, this being used as a basis for study or comparison. This approach is complex because of the need to use algorithms of lexical and semantic analysis focused on the issue of natural language processing, so that concepts of texts can be identified and subsequently organized in the most correct way possible.

Keywords: Concept maps. Education Method. Constructivist learning.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

AG Algoritmo Genético

API *Application Program Interface*

ID3 *Iterative Dichotomiser*

IE *Information Extraction*

IR *Information Retrieval*

KDD *Knowledge Discovery in Databases*

KDT *Knowledge Discovery in Text*

PLN Processamento de Linguagem Natural

RNA Rede Neural Artificial

SUMÁRIO

INTRODUÇÂO

Muitas aplicações têm como objetivo obter informação relevante a partir de uma grande massa de dados. Em muitos casos essa massa de dados está armazenada no formato de texto. Existem muitas informações que por vezes estão encobertas, perdidas nas grandes quantidades de texto que são apresentadas aos alunos. Os métodos de ensino devem se modernizar a ponto de colaborarem mais com o aluno, provendo ferramentas que o ajudem de maneira intuitiva e facilitada.

Dentre esses métodos, aqueles que prezam uma aprendizagem construtivista ao invés da memorística são os que possuem a maior gama de opções. Dentre essas ferramentas, temos os mapas conceituais, mapas gráficos onde ideias complexas são apresentadas ordenando-se hierarquicamente conceitos e subconceitos de um referido tema.

A fundamentação teórica de mapas conceituais está baseada na teoria de Aprendizagem ou teoria de Assimilação, desenvolvida por David Ausubel (1980). Em sua teoria, Ausubel explica como o conhecimento é adquirido e em que forma este fica armazenado na estrutura cognitiva do indivíduo. Dentro dessa ideia são organizados itens que são retirados do texto como termos relevantes e no mapa serão transformados em conceitos. Estes conceitos serão montados hierarquicamente em relação à sua relevância no contexto.

A proposta deste trabalho é elaborar uma ferramenta que, a partir de textos, consiga gerar mapas conceituais, levando em consideração os termos mais importantes.

A observação e posterior classificação dos conceitos, item chave de um mapa conceitual, será um processo crítico na ferramenta. Realizar a compreensão dos textos, buscando os itens que tenham maior relevância no contexto, além de filtrar palavras que não possuam valor significativo não constitui uma tarefa corriqueira.

Essa abordagem é complexa, devido à necessidade de se utilizar algoritmos de análise léxica e semântica focados na questão de processamento de linguagem natural, para que os conceitos dos textos possam ser identificados e, posteriormente, organizados da maneira mais correta possível.

Nos capítulos seguintes serão apresentados os itens relacionados à ferramenta. Primeiramente será explicada a ideia de mapa conceitual, no segundo capítulo serão apresentados os passos necessários para obter-se conhecimento a partir de texto. No capítulo 3 encontram-se alguns exemplos de algoritmos utilizados na mineração de dados. Na sequencia, o quarto capítulo apresenta a proposta de trabalho, o último capítulo comenta sobre a ferramenta idealizada. Logo em sequência surgem as considerações finais e referências bibliográficas do trabalho.

1 MAPAS CONCEITUAIS

Segundo Faria (1995), um mapa conceitual, pode ser definido como um esquema gráfico que apresenta uma estrutura de partes do conhecimento sistematizado, sendo representado por meio de uma rede de conceitos e proposições relevantes. O mapa conceitual na visão de Ontoria (1999), por sua vez, apresenta-se como uma técnica idealizada por Joseph Novak demonstrando-se como uma estratégia, método e recurso esquemático. Como definição de estratégia cita: “exemplos de estratégias simples, embora poderosas, para ajudar os estudantes a aprender e para ajudar os educadores a organizar os materiais que serão objetos desse estudo” (Novak e Gowin 1988 apud Ontoria 1999). Para definir método cita: “A construção dos mapas conceituais [...], que é um método para ajudar os estudantes e educadores a captar o significado dos materiais que se vão aprender” (Novak e Gowin 1988 apud Ontoria 1999). Por fim, a ideia de mapas conceituais como recurso é compreendida através da citação “Um mapa conceitual é um recurso esquemático para representar um conjunto de significados conceituais incluídos numa estrutura de proposições” (Novak e Gowin 1988 apud Ontoria 1999). Nas palavras de seu criador, o mapa conceitual tem a intenção de representar relacionamentos significativos de dois ou mais conceitos conectados por uma palavra de ligação, formando assim uma proposição. (Novak e Gowin 1988)

O mapa conceitual de acordo com a definição de Novak é composto por três elementos fundamentais:

- O conceito;
- A proposição;
- As palavras de ligação;

1.1 CONCEITO

A definição de conceito relacionada aos mapas conceituais entende-se como “uma regularidade nos acontecimentos ou objetos designados por algum termo” (Novak e Gowin, 1988). Entende-se que o conceito é um termo que referencia acontecimentos ou objetos. Os acontecimentos são qualquer coisa que ocorre, ou pode ser provocada, objetos, no entanto, são qualquer coisa existente e que pode ser observada. Ausubel (1980) define conceito como “objetos, eventos, situações ou propriedades que possuem atributos comuns, e que são designados por algum signo ou símbolo, tipicamente uma palavra com significado genérico”.

Os conceitos na ideia de Novak, o que seria também observado nos indivíduos, seriam imagens mentais provocadas em nós a partir das palavras ou símbolos com que exprimem regularidades. Embora seja semelhante nos indivíduos, a idealização dessas imagens mentais é pessoal, trazendo normalmente alguma diferença. No pensamento de Novak e Gowin, “os significados são, por natureza, idiossincráticos. Este caráter idiossincrático explica-se pela forma peculiar que cada um tem para captar inicialmente o significado de um termo, a experiência acumulada sobre a realidade, os sentimentos que provoca etc. O termo ‘automóvel’ por exemplo, não significa o mesmo para um corredor de Fórmula Um e para um ecologista”. Um mapa conceitual que explica essa personalidade na geração de conceitos foi feito por Novak (1998) e é reproduzido na figura 1.

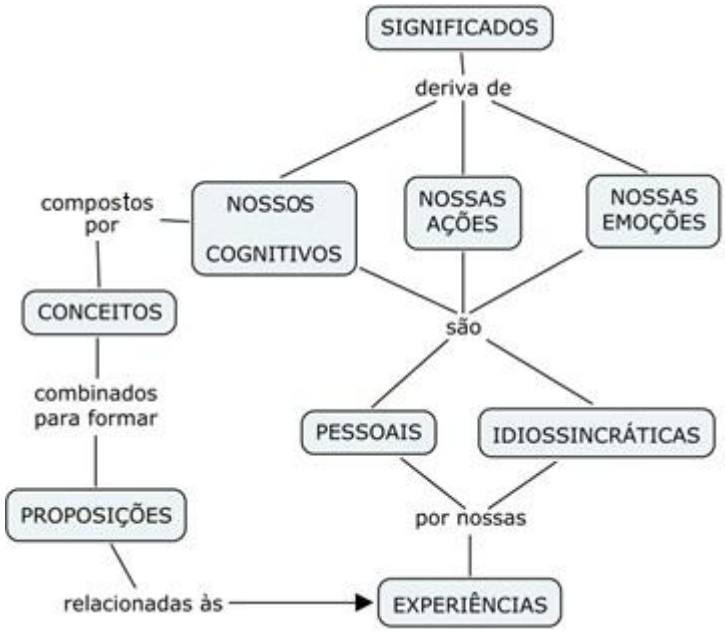

Figura 1 - Mapa conceitual que explica o entendimento personalizado dos conceitos.

Fonte: Novak, 1998

Para Hernández e García (1991), conceitos e imagens mentais se distinguem. Imagens teriam um caráter sensorial, enquanto os conceitos teriam um caráter abstrato. Ontoria (1999) resume essas ideias, comentando que os conceitos seriam imagens de imagens. Afirmam também que adquirimos através de descoberta apenas alguns conceitos, sendo que a maioria dos significados, relacionados às palavras, são aprendidos via proposições, que incluem o novo conceito.

1.2 PROPOSIÇÃO

Proposição, na definição de Ausubel (1980), “consiste de uma ideia composta expressa verbalmente numa sentença, contendo tanto um sentido denotativo quanto um sentido conotativo e as funções sintáticas e relações entre

palavras". No entendimento de Faria (1995) uma proposição "é formada por dois ou mais conceitos ligados à estrutura de uma sentença. A sua aprendizagem implica o domínio do significado dos conceitos que a compõem." Um conceito pode ser definido a partir de uma proposição. A proposição é a menor unidade semântica que possui um valor válido, uma vez que afirma ou nega algo de um conceito;

1.3 PALAVRAS DE LIGAÇÃO

Como o nome sugere, as palavras de ligação são utilizadas para unir conceitos e apontar um tipo de ligação que exista entre eles. Novak (1988) separa a proposição em dois elementos diferentes, em termos conceituais sendo estes as palavras que exprimem imagens mentais e representam regularidades, e palavras de ligação, os termos de união para outros conceitos e que não geram os mapas mentais. Um exemplo que ilustra claramente essa ideia é a frase: "O céu é azul.". Existem dois termos conceituais, "céu" e "azul", sendo unidos pela palavra de ligação "é". Assim essa frase pode gerar uma proposição simples com dois conceitos, sendo demonstrado na figura 2.

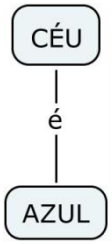

Figura 2 - Mapa conceitual simples com apenas uma proposição.

Fonte: Autor

Ao contrário do exemplo apresentado anteriormente, Ontoria (1999) lembra que existem mapas mais complexos, que se compõem em diversas ramificações,

com linhas de conceito diferenciadas. Com isso podem ocorrer relações cruzadas entre conceitos, estas são linhas de união que não ligam termos conceituais contíguos, ao contrário tem como objetivo ligar conceitos que estejam em ramificações, ou linhas conceituais diferentes.

Existe um terceiro tipo de termo que também provoca imagens mentais, porém não exprime regularidade e sim singularidades. Estes são os nomes próprios que podem designar exemplos de conceito. Eles são úteis especialmente para definir uma ideia singular a partir dos conceitos superiores, como é o caso de “banana” e “pêra” na figura 3.

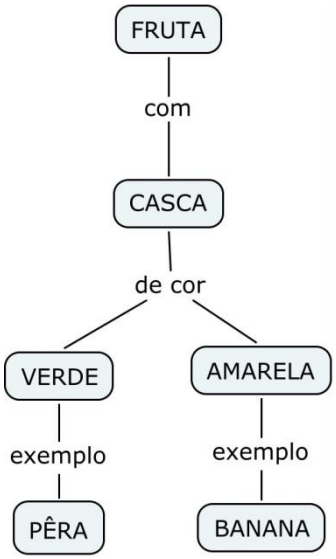

Figura 3 - Mapa conceitual, com definição de frutas.

Fonte Autor

Embora estejam aqui explicados os elementos básicos necessários para a elaboração de um mapa conceitual, não se pode esquecer que o mapa é apenas uma representação gráfica da estrutura mental elaborada via conceitos e proposições de um determinado tema. Existe uma vertente de pensamento sobre a elaboração gráfica que diferencia essa abordagem de outras técnicas semelhantes.

Segundo Ontoria (1999) “Esta vertente é a que permite classificar o mapa conceitual como técnica cognitiva e relacioná-lo com a aprendizagem significativa.” Estas características intrínsecas serão explicadas a seguir como pontos de diferenciação de outras técnicas cognitivas.

1.4 HIERARQUIZAÇÃO

Uma característica facilmente observada em um mapa conceitual é a hierarquia entre os conceitos que o compõe. Os conceitos mais abrangentes se encontram acima dos objetos mais específicos, lembrando que os nomes próprios que exemplificam um conceito são aqueles que surgem em último lugar, devido à sua representação singular. Embora o conceito de hierarquização seja de simples entendimento, Ontoria indica a necessidade de algumas considerações. No mesmo mapa conceitual, um conceito deve aparecer somente uma vez. Em certas situações é interessante para facilitar o entendimento, terminar as linhas das palavras de enlace com uma “seta” que aponta ao conceito derivado, isso colabora em situações em que ambos os conceitos estejam no mesmo nível hierárquico, ou ocorram relações cruzadas.

1.5 SELEÇÃO

Os mapas constituem um resumo, uma síntese composta pelas partes mais importantes e significativas de uma mensagem, tema ou texto. Para obter um mapa significativo, sobre um determinado assunto, a fase de seleção de termos relevantes que se tornarão conceitos dentro do mapa, é um processo muito importante, e determinante para a qualidade final da mensagem a ser passada pelo método. Como a utilidade que o mapa deve possuir, define os critérios de obtenção dos

termos importantes, existem situações onde diversos conceitos devem ser deixados de lado, para que o mapa utilize outros, que vão ao encontro à sua real finalidade. Outro parâmetro que modifica essa escolha de termos é se o objetivo será um mapa expositivo para demais pessoas, a partir de um tema, ou se o mapa tem escopo estritamente pessoal, sendo este último normalmente elaborado com menos cuidado.

Uma boa prática para contemplar o tema em diversos níveis é a elaboração de mapas com níveis de generalidade distintos. Assim, um mapa pode prover a visão global do assunto, contendo os itens mais abrangentes, e dando um panorama não específico. Agregado a este, são elaborados outros mapas, que focam em partes do mapa global, ou subtemas deste, proporcionando uma visão mais concreta e específica.

1.6 IMPACTO VISUAL

O impacto visual, e a primeira impressão obtida com o mapa conceitual, ajudam a definir sua qualidade e utilidade na exposição de conhecimento. Corrobora essa ideia a definição de Novak e Gowin (1988): “Um bom mapa conceitual é conciso e mostra as relações entre as ideias principais de um modo simples e vistoso, aproveitando a notável capacidade humana para a representação visual”. Obter esse tipo de resultado, logo na primeira tentativa de expor um tema no formato de mapa conceitual, não é uma tarefa simples, sendo assim, é recomendado por Ontoria (1999) que se revise um mapa depois de pronto, considerando-o um rascunho para que uma próxima representação consiga um melhor desempenho, resultando em uma visualização clara e precisa. Algumas sugestões dadas por ele incluem escrever os termos conceituais com letras maiúsculas, enquadrando-os em elipses, figura geométrica preferível ao retângulo, visando melhorar o contraste entre letras e o fundo.

Unindo todas essas ideias e recomendações, Ontoria (1999) elaborou um mapa conceitual com o objetivo de definir o próprio mapa conceitual. Uma representação do original é encontrada na figura 4.

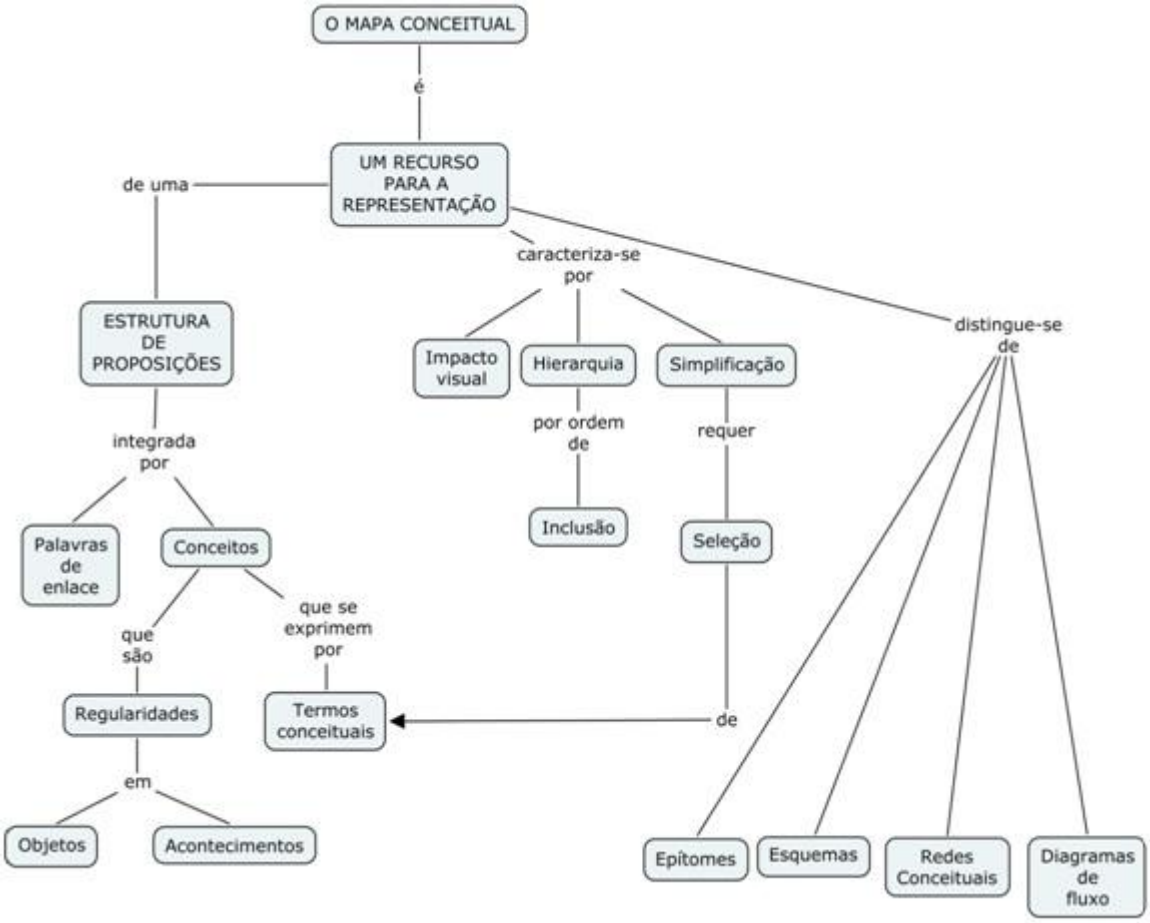

Figura 4 - Mapa conceitual, que explica o tema mapa conceitual.

Fonte: Ontoria et al., 1999

1.7 MAPA CONCEITUAL E MAPA COGNITIVO

Embora os termos possuam semelhanças, ambos definem resultados distintos. “O termo mapa cognitivo veio de Tolman (1948), um psicólogo neo-

condutista.” (Ontoria, 1999) Sua ideia de mapa cognitivo, começa a ser explicada com a hipótese de um animal perdido em um labirinto. Se nele, o animal reunir indícios auditivos, táteis e olfativos que tragam expectativa de comida, ele acaba por criar padrões que estabelecem o mapa cognitivo.

Tendo definições na psicologia ambiental, o mapa cognitivo representa um esquema que permite desenvolvimento em nosso meio ambiente e ajudar na resolução de problemas de localização, deslocamento e orientação. Como se pode imaginar, essa ideia forma uma estrutura muito dinâmica e flexível. “Neste campo, os mapas cognitivos tiveram muita ressonância para o conhecimento espacial ou ambiental, sobretudo quando se trata de analisar o conhecimento que se tem do meio ambiente físico ou geográfico”. (Ontoria apud De Veja, 1999)

Segundo Martin (1989) a psicologia ambiental define mapa cognitivo em três elementos básicos: os marcos, as rotas e configurações. Esses marcos seriam objetos atraentes do meio, que trazem recordações, além de coordenar atitudes relacionadas. Atuam como pontos estratégicos, tanto para manutenção de rumo ou mudança do mesmo. Rota seria “uma rotina motora e sensorial que permite a uma pessoa mover-se de um marco A para um marco B” (Kirasikk e Kail apud Ontoria 1999)

Em uma opinião de fora da psicologia ambiental, Novak diferencia mapa conceitual de mapa cognitivo da seguinte forma:

“Mapa cognitivo é o termo com o qual designamos a representação daquilo que cremos ser a organização dos conceitos e proposições na estrutura cognitiva de um determinado estudante. Os mapas cognitivos são idiossincráticos, ao passo que os mapas conceptuais devem representar uma área de conhecimento tal como a considerariam válida os especialistas na respectiva matéria. Pode acontecer que os especialistas não estejam de acordo com certos detalhes de um mapa (em parte

porque os conceitos mais importantes alteram-se constantemente com as novas investigações), muito embora a maioria admita que um mapa de conceitos bem concebido constituísse uma representação razoável de qualquer corpo de conhecimentos".

Algo contrastante entre o mapa conceitual e o mapa cognitivo é que o primeiro possui um caráter social, enquanto o segundo caráter individual psicológico. (López, 1991) Devido à grande variedade de organizações que se pode realizar com um mesmo bloco de conceitos, Novak admite a possibilidade de existirem diversos mapas cognitivos "corretos". Os mapas conceituais, a partir de ideias prévias ou estruturas cognitivas de um indivíduo, antes que este receba mais informações externas para comparação, seriam mapas cognitivos.

1.8 DISCUTIR SIGNIFICADOS TRABALHANDO EM GRUPO

Baseando-se nas propostas de Novak e Gowin (1988) utilizar mapas conceituais podem alcançar os objetivos almejados de compartilhar o conhecimento e adequar os pontos pertinentes através de discussão. Os mapas conceituais possuem, além de seu resultado prático final (um gráfico hierárquico explicando um determinado tema), um grande valor educativo vinculado ao processo que deve ser seguido durante sua elaboração. Como se trata de uma técnica de explicitar os conceitos adquiridos e ideias prévias estabelecidas nos indivíduos deve-se levar em conta essas noções prévias dos assuntos de todos. A partir dessas ideias é que serão geradas as proposições iniciais e os mapas de pré-conceitos que serão rascunho de um mapa conceitual mais conciso e definido. Esse exercício de repensar os conceitos existentes, dá espaço para a criatividade e pensamento diferenciado fazendo vislumbrar concepções de novos significados.

Essa procura de novos relacionamentos entre conceitos não é simples se agregada à função de assimilar novas ideias apresentadas sobre determinado assunto. Dentro da tese do pensamento reflexivo Novak e Gowin (1988), eles comparam a criação de um mapa conceitual com conceitos sendo inseridos e retirados ao treino de algum esporte. Para realizar um bom mapa conceitual é necessário o apoio de colegas de time que podem ajudar a adicionar ou remover conceitos que assim como no treino de futebol o jogador precisa da ajuda de seus companheiros.

Reforçando a ideia de valor que gerar um mapa conceitual possui, do ponto de vista educacional, nota-se que não existem mapas conceituais definitivos ou absolutamente sem equívocos. Durante sua elaboração o aluno, no caso de uma aula, deve muitas vezes desapegar-se dos conceitos próprios já estabelecidos, já que as novas ideias compartilhadas podem ser a melhor opção sendo utilizadas sempre que possível. Sabendo que a mesma explicação de um professor é absorvida de forma singular por cada aluno, pode haver discordâncias nos mapas conceituais gerados por eles. Nessa linha de raciocínio é comum ver alunos com resultados diferentes, e que não possuem os mesmos pontos de relevância instaurados por seu professor. Isso pode ser considerado comum, já que cada indivíduo terá sua percepção particular sobre o tema, o que fará com que, além de mapas diferentes, poderão apontar como conceitos equivocados.

Sobre essa distinção de resultados, Ontoria (1999) afirma que “a aprendizagem é uma experiência que se vive de forma individual, embora o conhecimento seja algo que pode ser partilhado”. Essa união de experiências particulares é muito enriquecedora do ponto de vista do aprendizado, gerando um exercício de argumentação, discussão e negociação de ideias. Como exemplo do enriquecimento proporcionado pela discussão das ideias, e o amadurecimento das escolhas feitas na elaboração de um mapa conceitual, Ontoria (1999) apresenta três mapas referentes ao período Paleolítico.

Esses mapas, realizados em sequência, possuem as seguintes distinções: um teve como foco a ordem temporal das ações, no outro, conceitos relacionados à sua estrutura receberam maior atenção. O terceiro mapa foi realizado em negociação com a turma toda, onde foram discutidas as ideias centrais que mereciam espaço no mapa generalista, essa negociação teve que escolher alguns itens e logicamente renegar outros. Esses mapas, e sua evolução como atividade podem ser verificados nas figuras 5, 6 e 7 a seguir.

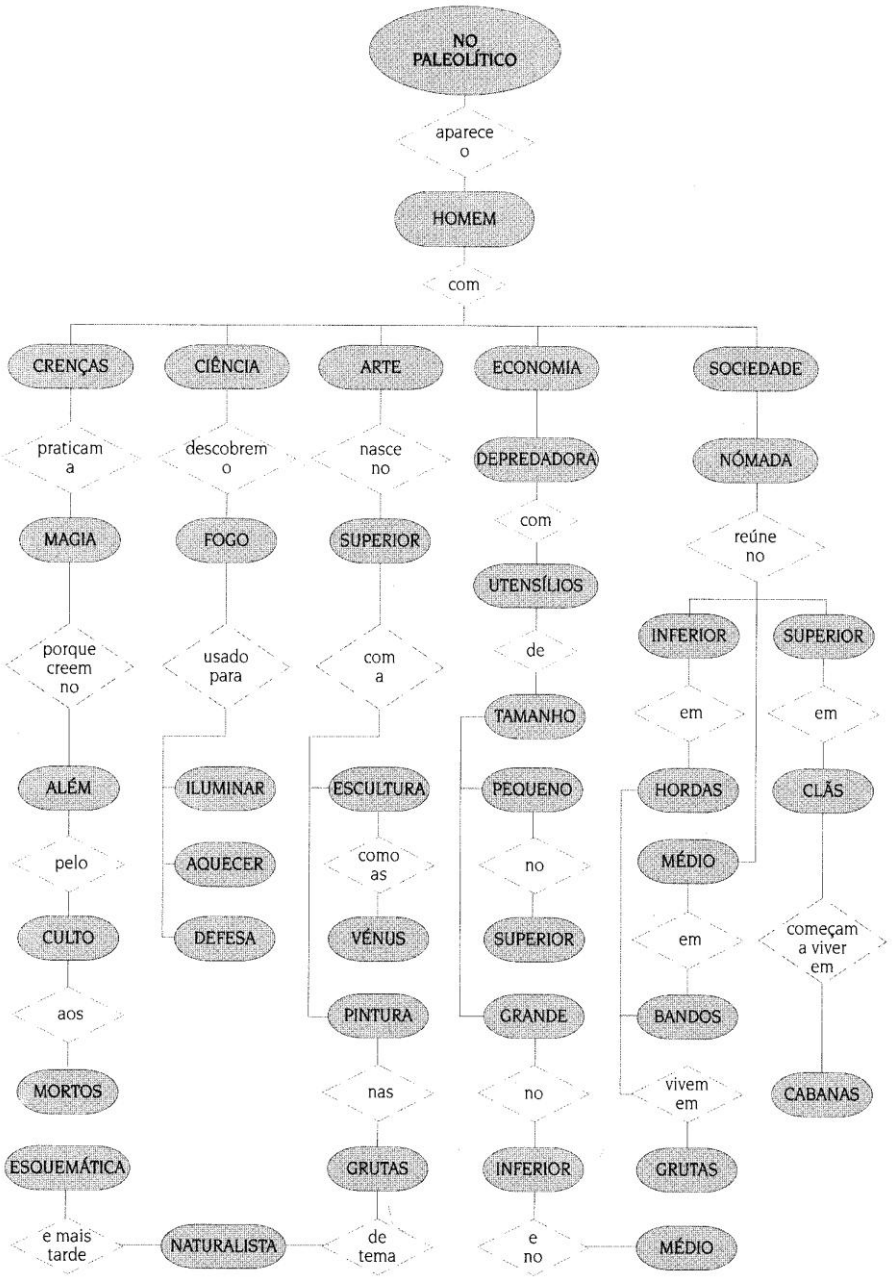

Figura 5 - Mapa conceitual do período paleolítico, de acordo com a ordem temporal.

Fonte: Ontoria et al., 1999

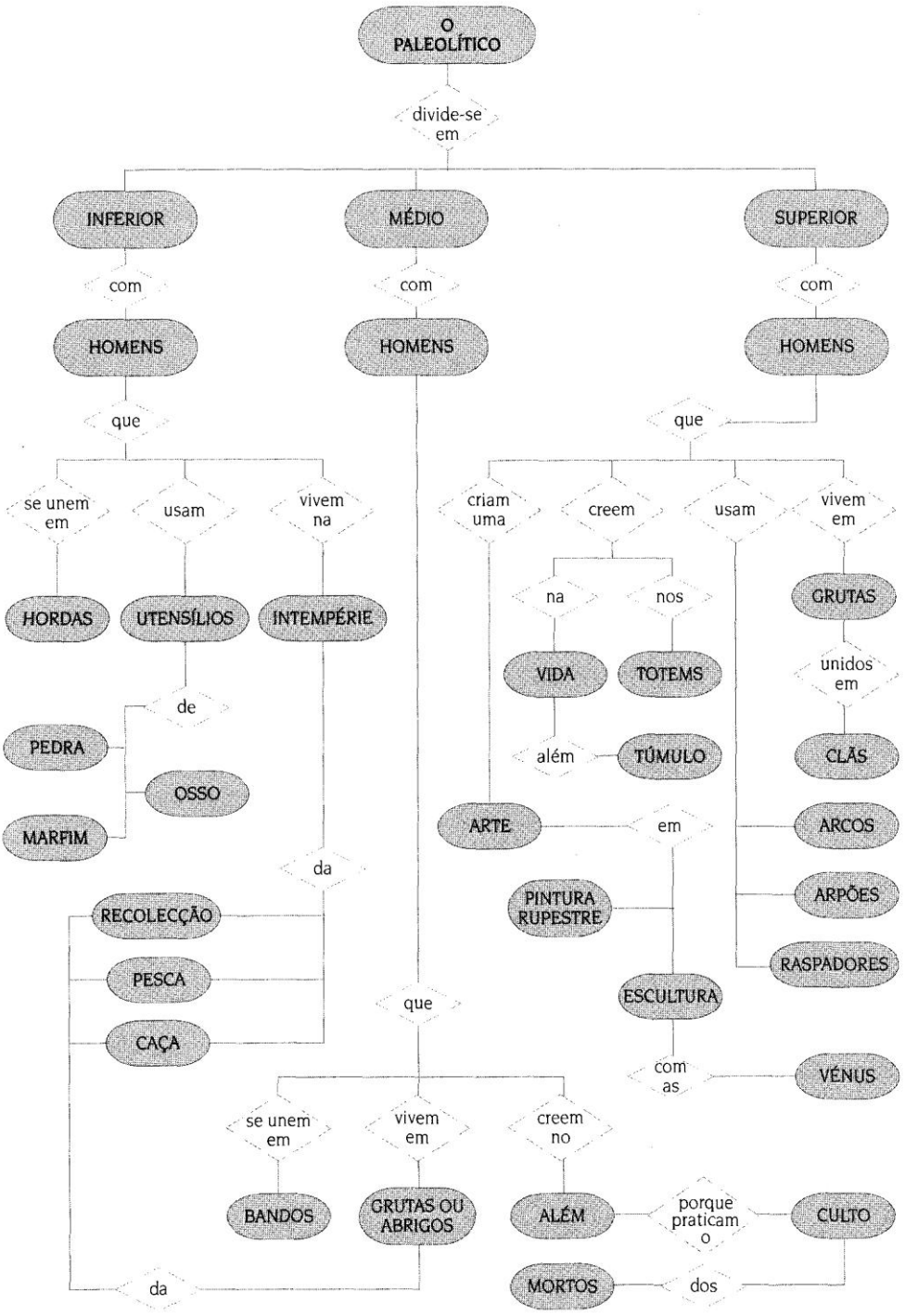

Figura 6 - Mapa conceitual do período paleolítico, de acordo com conceitos estruturais.

Fonte: Ontoria et al., 1999

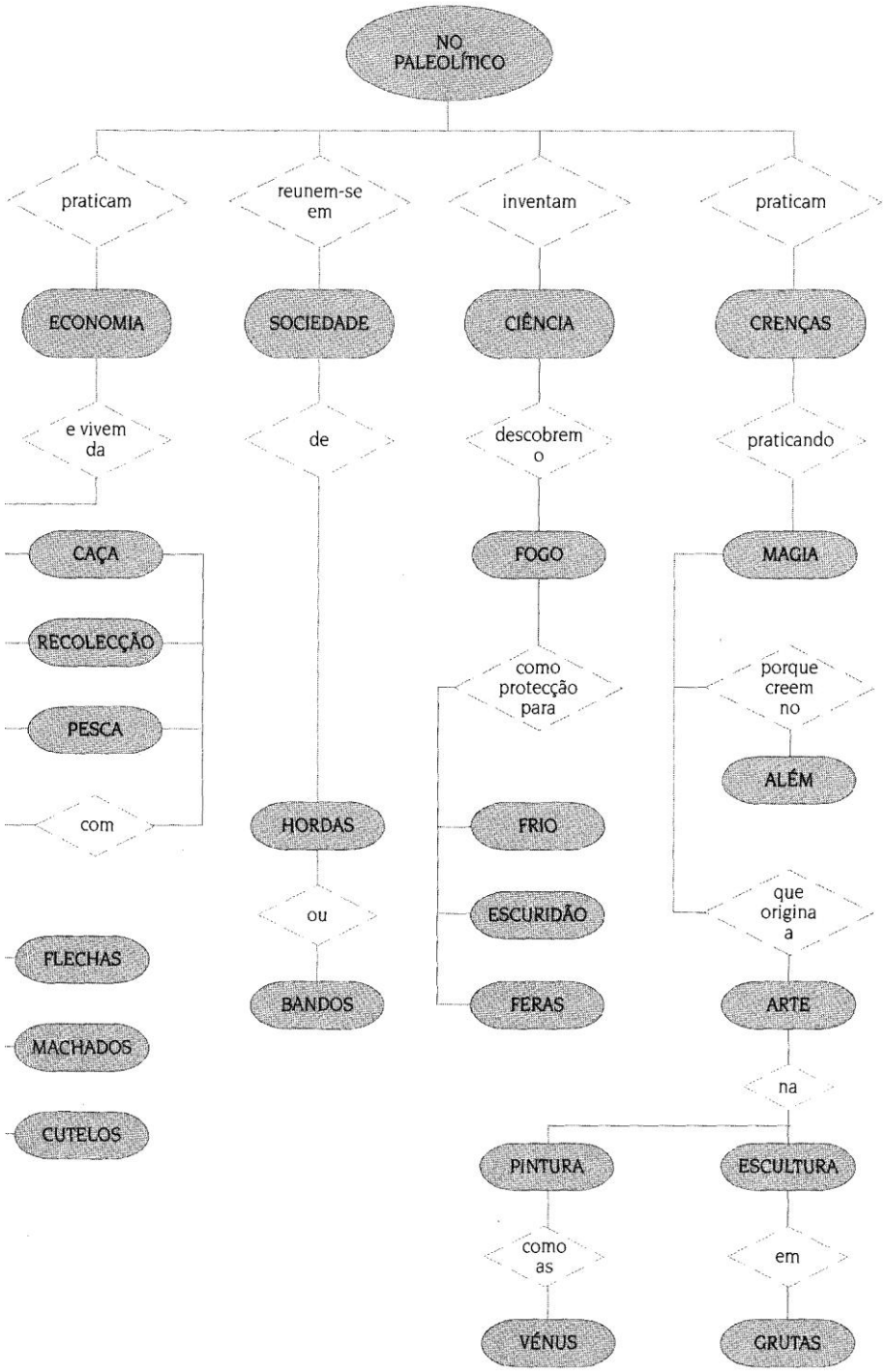

Figura 7 - Mapa conceitual do período paleolítico, feito com a participação da turma.

Fonte: Ontoria et al., 1999

2 DESCOBERTA DE CONHECIMENTO EM TEXTOS

Durante a elaboração de mapas conceituais, a etapa de seleção de itens relevantes é uma das mais importantes. No caso do deste trabalho, será uma etapa crucial, já que esse processo apontará os itens a serem utilizados como conceitos, e como serão distribuídos graficamente. Nesta abordagem serão utilizadas diversas técnicas de descoberta de conhecimento e *text mining*.

Enquanto o *data mining* é vinculado à procura de padrões de dados, o processo de *text mining* também conhecido como KDT relaciona-se busca de padrões de texto. Comparando-se ambas as técnicas, o KDT encontra maiores dificuldades, pois trabalha com dados sem forma ou estrutura definida. Mesmo com esses contratempos é possível a descoberta de conhecimento a partir de textos já que não se faz necessário que se entenda o sentido de um determinado texto para que se extraia dele informações úteis. (WITTEN, FRANK. 2000).

De acordo com Silva (2002), a utilização da KDT pode ser encarada como o uso do processo de KDD em dados que não sejam estruturados, como por exemplo, as informações que podem sem encontradas na internet, ou informações relevantes que são armazenadas nas corporações.

Segundo Ah-Hwee Tan (1999), cerca de 80% de todas as informações armazenadas por uma organização se encontra em um formato textual, o que indica a necessidade de extratores de conhecimento a partir de textos. Diversas linhas de pesquisas direcionam-se ao uso extensivo de KDT, esta sendo considerada a forma mais natural de armazenamento de informação. (Tan, 1999)

Para obter resultados satisfatórios, a KDT une técnicas de PLN, sumarização de documentos, extração e recuperação de informação em conjunto com os métodos de data mining (Dixon, 1997). Embora exista a recomendação de

uso dessas técnicas não há uma definição clara de um plano de uso para elas. O que de acordo com Loh (2000), acaba deixando uma lacuna sobre a definição de como uma coleção de textos deve ser investigada, de maneira semi ou completamente automática para que sejam validadas as hipóteses.

Em sua abordagem comum as aplicações de *data mining* fazem uso de informações que são cuidadosamente preparadas. Existem processos de transformação dos dados na etapa conhecida como pré-processamento. Além deste ajuste nos dados deve-se ter atenção sobre a seleção das informações que serão posteriormente encaminhadas à etapa de mineração. Bem como o *data mining*, o text mining necessita desse cuidado prévio. Antes de iniciar-se um processo de descoberta de conhecimento esse melhoramento é requerido, já que os textos normalmente são uma coleção de dados não estruturados e não possuem um tratamento para a composição de documentos (SHOLOM, 2005).

Tanto o *text mining* quanto o *data mining*, possuem como objetivo principal, extrair informações que sejam úteis partindo de um banco de dados, utilizando para tal a identificação e exploração de padrões que sejam interessantes à aplicação. Obviamente quando falamos de *text mining* tratamos como base de dados, uma coleção de textos, onde padrões interessantes são encontrados não em registros e sim nos textos não estruturados informados. (FELDMAN, SANGER. 2007).

De maneira que o *text mining* é oriundo de pesquisas relacionadas ao data mining, não surpreende que ambas as técnicas possuam similaridades. Ambas utilizam rotinas de pré-processamento, utilizam algoritmos de descoberta de padrões e, por fim, usam pós-processamento que será responsável por prover a visualização dos resultados (FELDMAN, SANGER. 2007).

A KDT realiza uma combinação de técnicas de extração, recuperação de informação, processamento da linguagem natural e sumarização de documentos

com os métodos de DM. Por utilizar dados não estruturados, esta é considerada mais complexa que o KDD (SILVA. 2002).

O *text mining* pode receber a definição de uma aplicação dos recursos e técnicas computacionais sobre dados textuais, com o objetivo de encontrar informações intrínsecas e relevantes que se escondiam anteriormente na massa de dados e que, por este mesmo motivo, eram desconhecidas por seus usuários (PRADO, OLIVEIRA, FERNEDA, WIVES, SILVA, LOH. 2005).

Seguindo modo de execução que se assemelha bastante ao KDD, o KDT utiliza-se de tarefas de pré-processamento, mineração dos dados, representação dos resultados obtidos e reformulação do processo como um todo se necessário (quando o resultado obtido ficou abaixo do esperado). Na sequência será apresentado um detalhamento sobre essas etapas que fazem parte do KDT.

2.1 PRÉ-PROCESSAMENTO

Como etapa inicial do processo de KDT ela se foca na limpeza dos dados, a fim de facilitar a análise da etapa seguinte, que é a extração do conhecimento propriamente dita.

O pré-processamento pode dividir-se em até cinco grandes fases: correção ortográfica, remoção de stopwords, processo de stemming e seleção dos termos relevantes. As etapas devem ser aplicadas nessa ordem, porém não há obrigatoriedade de execução de todas elas. Em alguns casos existe uma fase adicional chamada identificação de termos. (GOMES, MONTEIRO, OLIVEIRA. 2006).

Durante essa etapa do pré-processamento, as operações estão focadas na identificação e extração de representações de linguagem natural em documentos. É dessa operação a responsabilidade de transformar as informações não estruturadas, em coleções de documentos que se encontrem no formato mais estruturado possível, o que é uma preocupação irrelevante para maioria dos sistemas que executam data mining. A etapa de Pré-processamento é “obrigatória” durante o processo de *text mining*, já que geralmente os algoritmos que realizam descoberta de conhecimento não são utilizados em coleções de textos que estejam despreparadas (FELDMAN, SANGER. 2007).

Depois de realizados os processos que têm como objetivo transformar os dados textuais não estruturados ou semiestruturados, em dados estruturados, as técnicas apresentadas anteriormente de mineração de dados podem ser utilizadas agora sobre o texto processado (BARION, LAGO. 2008). Desta maneira existem aplicações de *text mining* que apenas realizam a etapa de Pré-processamento, assim, os dados textuais já estarão estruturados e técnicas de *data mining* poderão ser utilizadas.

2.1.1 Correção ortográfica

Seguindo a lógica de seu nome, esta fase destina-se à verificação de palavras que estejam em conformidade com sua ortografia. Normalmente um verificador ortográfico faz uso de um dicionário e faz uma comparação palavra por palavra com os termos do dicionário. Quando a palavra encontrada também existir no dicionário do verificador, esta é considerada uma palavra ortograficamente correta e aceita no texto. (GOMES, MONTEIRO, OLIVEIRA. 2006).

Segundo Toniazzo (2005), os termos que são obtidos durante o processo de extração podem conter erros. Estes erros podem vir através da má digitação, sendo

assim, uma correção deve ser realizada para permitir que o processo de tratamento das informações da aplicação funcione corretamente.

2.1.2 Remoção de *stopwords*

Essa fase basicamente consiste na retirada de palavras que se repetem inúmeras vezes no decorrer do texto ou palavras que possuem pouco significado, tais como artigos, preposições, e algumas conjunções. Assim, palavras que não possuam nenhuma relevância aparente para o entendimento do texto, serão consideradas stopwords. Geralmente define-se um conjunto dessas palavras em um arquivo, contendo a lista de todas as palavras consideradas stopwords de acordo com a aplicação. (GOMES, MONTEIRO, OLIVEIRA. 2006).

De acordo com Silva (2002), como existe um uso bastante frequente das stopwords, sua eliminação pode levar a redução entre 40% a 50% dos textos que serão analisados. Na figura 8 é apresentada uma lista de palavras geralmente consideradas stopwords em aplicações de *text mining*.

coisa	deste	ela	eu	lo	nestas	pelos	poucos	seu	toda
coisas	destes	elas	fazendo	mas	ninguém	pequena	primeiro	seus	todas
com	deve	ele	fazer	me	no	pequenas	primeiros	si	todavia
como	devem	eles	feita	mesma	nos	pequeno	própria	sido	todo
contra	devendo	em	feitas	mesmas	nós	pequenos	próprias	só	todos
contudo	dever	enquanto	feito	mesmo	nossa	per	próprio	sob	tu
da	deverá	entre	feitos	mesmos	nossas	perante	próprios	sobre	tua
daquele	deverão	era	foi	meu	nosso	pode	quais	sua	tuas
daqueles	deveria	essa	for	meus	nossos	pôde	qual	suas	tudo
das	deveriam	essas	foram	minha	num	podendo	quando	talvez	última
de	devia	esse	fosse	minhas	numa	poder	quanto	também	últimas
dela	deviam	esses	fossem	muita	nunca	poderia	quantos	tampouco	último
delas	disse	esta	grande	muitas	o	poderiam	que	te	últimos
dele	disso	está	grandes	muito	os	podia	quem	tem	um
deles	disto	estamos	há	muitos	ou	podiam	são	tendo	uma
depois	dito	estão	isso	na	outra	pois	se	tenha	umas
dessa	diz	estas	isto	não	outras	por	seja	ter	uns
dessas	dizem	estava	já	nas	outro	porém	sejam	teu	vendo
desse	do	estavam	la	nem	outros	porque	sem	teus	ver
desses	dos	estávamos	la	nenhum	para	posso	sempre	ti	vez
desta	e	este	lá	nessa	pela	pouca	sendo	tido	vindo
destas	é	estes	lhe	nessas	pelas	poucas	será	tinha	vir
deste	e'	estou	lhes	nesta	pelo	pouco	serão	tinham	vos

Figura 8 - Lista de stopwords frequentes em aplicações de Text Mining.

Fonte: Adaptado de LOH, 2008.

2.1.3 Processo de *stemming*

Dentro de um mesmo texto, diversas palavras podem se apresentar com variações morfológicas. Como exemplo dessas variações pode-se citar: formas de gerúndio, sufixos temporais e o plural. O processo de *stemming* tem como objetivo a remoção dessas variações, sendo o seu resultado denominado stem (caule) (GOMES, MONTEIRO, OLIVEIRA. 2006).

Segundo Silva (2002), a vantagem é que, em uma busca, o usuário não necessita preocupar-se com a classe da palavra, podendo ela aparecer no texto

como um substantivo, um verbo ou um adjetivo. No entanto, isto implica diminuição na precisão da pesquisa.

2.1.4 Seleção de termos relevantes

Assim como existem as *stopwords*, palavras consideradas sem nenhum significado importante e pouca relevância para o texto, existem também as palavras que são o oposto. Estas palavras possuem grande importância no texto, como por exemplo, as palavras que são encontradas em tópicos, como títulos e substantivos.

De acordo com Silva (2002), deve-se considerar que em um texto as palavras possuem níveis de destaque distintos. Aquelas mais frequentes, excluindo-se as *stopwords*, são mais importantes que outras palavras que aparecem com menos frequência.

Segundo Toniazzo (2005), existem três maneiras para verificar-se a importância de uma palavra em relação à sua frequência no texto. Essas frequências, na verdade, servem para identificar o peso do termo ou a força do termo. Estas formas de verificação estão apresentadas a seguir.

• **Frequência Absoluta:** indica a quantidade total de ocorrências de uma determinada palavra no documento ou conjunto analisado.

• **Frequência Relativa:** frequência de uma palavra em relação a todas as outras palavras do documento.

• **Frequência Inversa de Documentos:** é uma medida que leva em consideração tanto a Frequência Absoluta quando a Frequência Relativa.

2.1.5 Identificação dos termos relevantes

Baseia-se na identificação das palavras que são importantes no texto, ignorando, nesse caso, símbolos e caracteres de controle de arquivo ou formatação. Isso se deve ao fato de que existem sequências de caracteres que muitas vezes podem ser substituídos por outras palavras ou mesmo termos compostos que não podem ser tratados individualmente. Como exemplo desses casos há os termos compostos: processo judicial e processo computacional. Este é um caso onde esta fase se faz necessária para que não haja perda de significado (SILVA, 2002).

2.2 MINERAÇÃO DE TEXTO

Da mesma forma que ocorre no KDD, esta etapa consiste na utilização de um algoritmo de análise sobre os dados com o objetivo de identificar e extrair informações até então desconhecidas e que possam ser úteis. Dentre os métodos existentes para a realização desta etapa, é muito importante preocupar-se com uma característica fundamental: a relevância da informação (TONIAZZO, 2005).

De acordo com Gomes, Monteiro e Oliveira (2006), os algoritmos e técnicas desenvolvidos para esta etapa podem ser divididos em duas categorias: a de geração de conhecimento, que utiliza técnicas para gerar conhecimento a partir de informações contidas em um determinado texto, e a de extração de conhecimento, que utiliza técnicas para retirar o conhecimento que esteja explícito no texto. A seguir serão apresentados alguns dos principais tipos de mineração em textos.

2.2.1 Recuperação da informação

Segundo Silva (2002), este tipo específico de mineração de texto (também conhecida pelo termo inglês *Information Retrieval* - IR), tem como objetivo localizar documentos que contenham informações relevantes para atender às necessidades definidas pelo usuário em uma consulta. Desta forma, o usuário ainda necessita examinar os documentos resultantes dessa busca a fim de encontrar a informação, o que torna a tarefa demorada.

Na intenção de diminuir esse esforço, utiliza-se a indexação, que provê uma busca mais rápida e eficiente. Esta indexação, considerada um tipo de filtro, que possui a capacidade de realizar a seleção e identificação das características de um determinado documento. Posteriormente extrair os termos mais relevantes e desconsiderando aquilo que não possui muita importância.

2.2.1.1 Indexação

De acordo Barion e Lago (2008), a indexação pode ser definida como um processo onde as palavras contidas no texto são armazenadas em uma estrutura de índices, a fim de viabilizar a pesquisa por documentos através das palavras encontradas.

A indexação tem como objetivo realizar uma busca rápida de documentos através de palavras chave. Utilizar uma estrutura de dados de armazenamento inteligente, o que possibilita um aumento drástico no desempenho. A indexação, fora fornecer a recuperação de dados textuais, ainda pode realizar cálculos com diversas palavras chave de busca, ordenando de acordo com a avaliação de cada documento. (ARANHA, PASSOS. 2006).

A fim de obter um melhor desempenho nos processos de indexação, são realizados alguns processos que reduzem o número de termos do texto. Esses processos normalmente se encontram na fase de pré-processamento, como a correção ortográfica, identificação de termos mais relevantes e remoção de stopwords. Após esses passos, o índice que será gerado, por ser possivelmente bem reduzido, proporcionará maior rapidez e agilidade do sistema.

2.2.2 Extração de informação

Esta tarefa de mineração de texto (também conhecida pelo termo inglês *Information Extraction* - IE) possui como objetivo primordial, analisar dados não estruturados ou semiestruturados, que no caso seriam os textos, e a partir destes extrair informações úteis visando armazená-las em um banco de dados.

Segundo Silva (2002), a IE aborda metodologias, técnicas e ferramentas que têm como objetivo encontrar dados específicos dentro de um texto, extraindo de forma automática os valores vinculados aos atributos, da mesma forma que extrairia de um banco de dados. Geralmente as aplicações voltadas para esse tipo de solução são deveras dependentes de seu domínio, isto é, apresentarão bom desempenho apenas nas classes de documentos inclusas em seu domínio.

Essa extração de informação é resultado do Processamento de Linguagem Natural (PLN), área que estuda os problemas ocorridos nos momentos de geração e compreensão automáticas das línguas humanas naturais.

Os processos que envolvem a extração de informação são mais simples do que o PLN, necessitando de definições no que diz respeito às informações que devem ser extraídas e exatamente as regras a serem seguidas para que a extração possa ocorrer (TONIAZZO, 2005). Este processo específico de extração de

informações deve identificar as palavras dentro de conceitos pré-especificados, contendo ainda uma etapa onde ocorre uma transformação que modifica a informação extraída, tornando-a compatível com um banco de dados. (BARION, LAGO. 2008).

O processo de extração necessita ser realizado de acordo com um domínio pré-definido, domínio este que carrega as informações que se deseja encontrar no texto. O nome deste domínio é *Slot*. Realizando-se uma analogia simples, os slots podem comparar-se com os atributos dos bancos de dados no formato atributo-valor. Trazendo um exemplo prático, se forem verificados textos que contenham o assunto relacionado às doenças e suas transmissões. Neste caso, os slots que poderiam ser preenchidos com as seguintes informações: origem da doença, nome da doença, forma de transmissão, tratamentos, etc. Essas lacunas que deverão ser preenchidas com as informações obtidas no texto denominam-se templates. (BARION, LAGO. 2008).

De modo a exemplificar esse formato de mineração de dados, serão apresentadas, na sequência, duas figuras que demonstram sua utilização referenciando um texto que veicula os requisitos necessários de uma determinada vaga de trabalho, adaptado de Mooney e Bunescu (1995).

Job Title: Senior DBMS Consultant
Location: Dallas,TX
Responsibilities:
DBMS Applications consultant works with project teams to define DBMS based solutions that support the enterprise deployment of Electronic Commerce, Sales Force Automation, and Customer Service applications.
Desired Requirements:
3-5 years exp. developing Oracle or SQL Server apps using Visual Basic, C/C++, Powerbuilder, Progress, or similar.
Recent experience related to installing and configuring Oracle or SQL Server in both dev. and deployment environments.
Desired Skills:
Understanding of UNIX or NT, scripting language. Know principles of structured software engineering and project management

Figura 9 - Texto que descreve os requisitos necessários de uma vaga de emprego.

Fonte: Adaptado de MOONEY, BUNESCU, 1995, p. 4.

title: Senior DBMS Consultant
state: TX
city: Dallas
country: US
language: Powerbuilder, Progress, C, C++, Visual Basic
platform: UNIX, NT
application: SQL Server, Oracle
area: Electronic Commerce, Customer Service
required years of experience: 3
desired years of experience: 5

Figura 10 - Resultado da aplicação de Extração de Informação sobre a Figura 9.

Fonte: Adaptado de MOONEY, BUNESCU, 1995, p. 4.

3 ALGORITMOS DE CLASSIFICAÇÃO DE DADOS

Para se realizar a classificação de dados, existem diversos algoritmos, os principais serão apresentados neste capítulo. É conveniente destacar que não existe um formato ou algoritmo que possa ser utilizado exclusivamente. Diante de um problema, a escolha dos algoritmos a serem utilizados se dará em grande parte relacionada à natureza dos dados a serem classificados. Desta maneira, não há um algoritmo melhor em todos os casos, já que a cada situação e necessidade influenciarão nessa escolha. Para se encontrar o melhor modelo a ser utilizado se faz necessária uma variedade de tecnologias e ferramentas. (TWO CROWS CORPORATION. 1999).

3.1 ÁRVORES DE DECISÃO

Como o próprio nome leva a crer, as árvores de decisão possuem uma estrutura semelhante à de uma árvore e definem-se como forma de representar resultados obtidos em tarefas de mineração de dados baseando-se em algoritmos de classificação. Uma árvore é composta por diversos nós internos, sendo que cada um tem o papel de representar uma decisão sobre um atributo. Isso acaba por determinar como os dados serão particionados através de seus nós filhos.

Ross Quinlan, da Universidade de Sydney, Austrália, é reconhecido por muitos estudiosos da área de mineração de dados como o “pai das árvores de decisão”. Ross concebeu, em 1983, um novo algoritmo chamado ID3. Tanto o ID3 quanto suas sucessoras evoluções (ID4, ID6, C4.5, See 5) são adaptadas para o uso combinado com árvores de decisão, ao passo que estes algoritmos produzem regras que são ordenadas de acordo com sua importância. (JERÔNIMO. 2001).

Segundo Two Crows Corporation (1999), as árvores de decisão definem-se como um modelo de representação de um conjunto de regras que manipulam uma classe ou um valor. Podemos usar como exemplo, empresas que precisam classificar os clientes como bons ou maus pagadores.

De acordo com Santos (2008), uma árvore de decisão se compõe por um grupo de nós que se conectam entre si por meio de ramificações, este grupo de nós pode dividir-se em:

- Nodo raiz: nodo base, que inicia a árvore;
- Nodos comuns: são aqueles que representam uma decisão, manipulam um atributo dividindo-o e gerando novas ramificações;
- Nodo folha: possui as informações de classificação do algoritmo.

Para se realizar a interpretação de uma árvore de decisão, devem-se seguir os seguintes passos: cada um dos nós não folha representam uma decisão dentro da árvore, decisão essa que envolve um atributo e um conjunto de valores possíveis. Os nós folha são aqueles que representam a atribuição de um valor ou grupo de valores a um atributo do problema. Cada caminho da árvore que se inicia no nó raiz e termina em algum nó folha, irá corresponder a uma regra na seguinte forma: SE <condições> ENTÃO <conclusão> (GOLDSCHMIDT, PASSOS. 2001).

Diversos métodos de classificação apresentam como inconveniente o fato de não proporcionar fácil assimilação para as pessoas, o RNA é um exemplo disso. No entanto, as árvores de decisão formam um dos algoritmos de classificação mais simples de entender, devido à clareza de como é possível chegar-se a determinado resultado, utilizando as regras criadas pelo modelo (FELDMAN e SANGER. 2007).

Segundo Goldschmidt e Passos (2001), geralmente os algoritmos que são baseados em abstração das árvores de decisão possuem duas fases: construção da árvore de decisão e simplificação da árvore de decisão.

Os passos a serem seguidos para criar uma árvore de decisão são: a partir de um conjunto de dados estabelecido, cabe ao usuário definir uma das variáveis existentes como objeto de saída. Tendo essa informação, o algoritmo se encarrega de encontrar o fator relacionado à variável de saída que possua maior importância, e o define como raiz da árvore. Todos os fatores restantes são então catalogados como nós definindo cada nível até o final, o nível denominado folha.

Exemplificando esse feito, a tabela 1 demonstra um conjunto de dados, que será utilizado para formar uma árvore de decisão, na sequência é apresentada a árvore gerada a partir destes dados. As regras que compõe a árvore de decisão apresentada na figura 11 são mostradas na figura 12.

Tabela 1- Base de dados antes da aplicação do algoritmo de Árvore de Decisão.

NOME	ESCOLARIDADE	IDADE	RICO (atributo classe)
Alva	Mestrado	>30	Sim
Amanda	Doutorado	<=30	Sim
Ana	Mestrado	<=30	Não
Eduardo	Doutorado	>30	Sim
Inês	Graduação	<=30	Não
Joaquim	Graduação	>30	Não
Maria	Mestrado	>30	Sim
Raphael	Mestrado	<=30	Não

Fonte: GONÇALVES

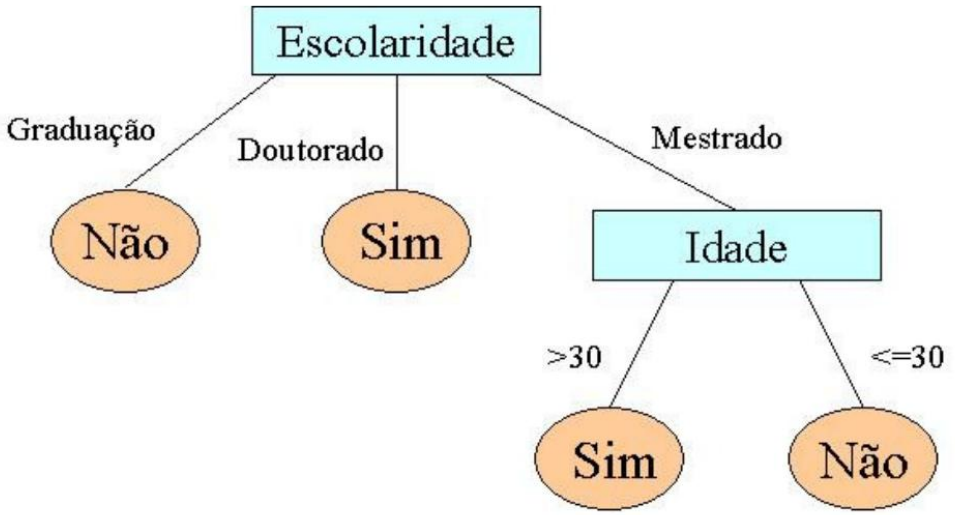

Figura 11 - Árvore de decisão construída a partir do conjunto de dados da Tabela 3.1.

Fonte: GONÇALVES.

Se (Escolaridade = Graduação) então Rico = Não
Se (Escolaridade = Doutorado) então Rico = Sim
Se (Escolaridade = Mestrado) e (Idade >30) então Rico = Sim
Se (Escolaridade = Mestrado) e (Idade <=30) então Rico = Não

Figura 12 - Regras que formam a árvore de decisão da figura 11.

Fonte: GONÇALVES

Embora seu simples entendimento e facilitada utilização, a árvore de decisão possui desvantagens. Segundo Silva (2008), entre as desvantagens existentes na técnica que envolve árvores de decisão, é a necessidade de existir uma considerável quantidade de dados, para que estruturas complexas possam ser descobertas, além da possibilidade de erros de classificação se esta envolver muitas classes.

De acordo com Jerônimo (2001), entretanto, há vantagens quando se usa árvores de decisão, entre elas destacam-se o fato de que essa abordagem realiza decisões levando em consideração o grau de relevância e o fato de serem perfeitamente entendidas pela maior parte das pessoas.

A fim de expor um maior esclarecimento a respeito dos algoritmos de árvores de decisão, é apresentado na sequência um destes, o C4.5

3.1.1 Algoritmo C4.5

Dentre os algoritmos conhecidos e utilizados para as tarefas de classificação, o C4.5 é um dos mais famosos. O objetivo deste método é abstrair as árvores de decisão, utilizando uma abordagem recursiva de particionamento das bases de dados. Este algoritmo foi baseado em seu antecessor, o ID3. Os dois algoritmos foram desenvolvidos por John Ross Quinlan (GOLDSCHMIDT, PASSOS. 2005).

O objetivo desse algoritmo, segundo Stahnke (2008), é abstrair as árvores de decisão em um método recursivo de particionamento das bases de dados. Geralmente, esses algoritmos possuem duas fases: a construção da árvore de decisão e, posteriormente, sua simplificação, da mesma forma como citado anteriormente.

De acordo com Silva (2008), este algoritmo transforma a árvore de decisão em um conjunto de regras, estas ordenadas de acordo com sua importância, com o intuito de facilitar a identificação dos fatores mais importantes dentro da estrutura.

A regra detectada como sendo a mais relevante para o contexto da árvore é apresentada como sendo o seu primeiro nó (nó raiz), enquanto as demais, menos

relevantes, são posicionadas nos nós abaixo, ou seja, quanto mais alto for o nó em questão, maior é a sua relevância dentro das decisões encontradas pela árvore de decisão. (JERÔNIMO. 2001).

3.2 REDES NEURAIS

Segundo Jerônimo (2001), essa tecnologia é a que oferece o mais profundo poder de mineração, em contra partida, é a mais difícil de entender, pois se tenta construir representações internas de modelos ou padrões achados nos dados, porém essas representações não são apresentadas ao usuário. Elas são utilizadas pelo processo de descoberta de padrões, que acaba trabalhando em uma espécie de “caixa-preta”.

As redes neurais são modelos matemáticos que foram inspirados nos princípios de funcionamento dos neurônios biológicos, e na estrutura encontrada no cérebro. Esses modelos possuem a capacidade de obter, armazenar e utilizar conhecimento experimental, buscando simular computacionalmente habilidades humanas tais como aprendizado, generalização, associação e abstração. (GOLDSCHMIDT, PASSOS. 2003)

Na figura 13 pode-se observar a representação gráfica de uma rede neural simples. Nela os neurônios são representados por círculos enquanto as linhas representam o peso das conexões entre eles. Nota-se que um RNA é basicamente divido em três camadas: camada de entrada, responsável por receber os dados externos; camada de saída, responsável por apresentar os dados obtidos; por fim, a camada interna ou escondida, que é a camada onde ocorre o processamento interno da rede. Salienta-se que podem existir diversas camadas escondidas, sendo esse número relacionado à complexidade do problema.

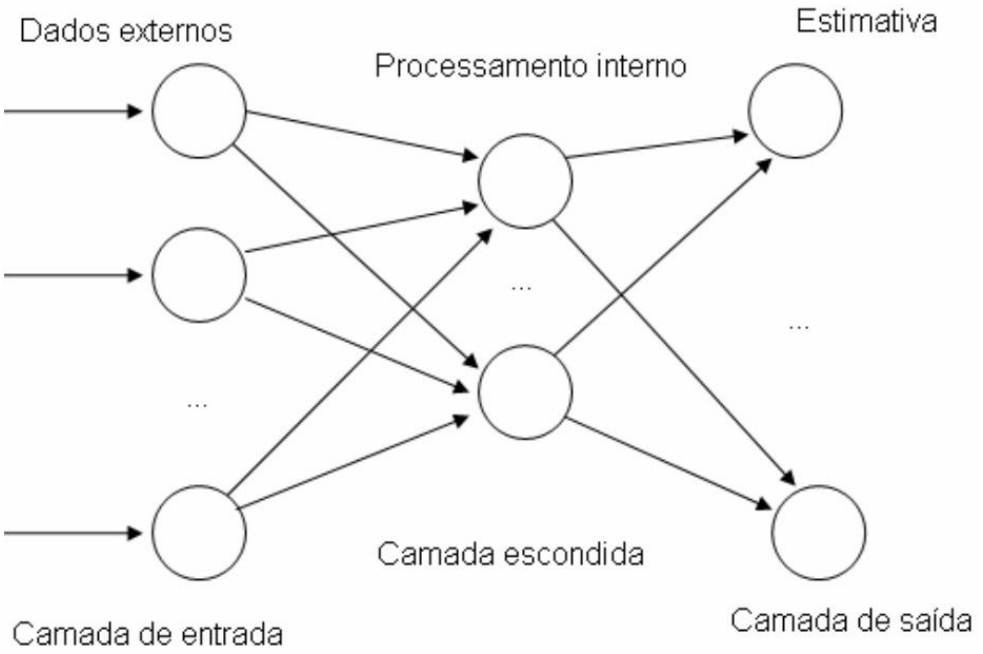

Figura 13 - Arquitetura de uma rede neural simples.

Fonte: Adaptado de GOLDSMITH, PASSOS, 2003 p. 176.

Segundo Kranz (2004), a tecnologia das redes neurais possui fundamentalmente duas técnicas. A primeira, denominada aprendizagem supervisionada, envolve o treinamento da rede, tendo por base um conjunto de dados pré-estabelecido. Nesta técnica, as diversas entradas na rede serão pesadas de maneira distinta, a partir da experiência da aprendizagem. Esse procedimento é repetido, até que se obtenha um ponto ótimo global. A segunda técnica, por sua vez, é a aprendizagem não supervisionada. Por não haver supervisão, os padrões serão detectados enquanto os dados são passados através do programa. Como não há uma pesagem prévia, os padrões serão detectados como um subproduto do processo.

Os RNAs possuem diversas vantagens, uma que se pode destacar está atrelada ao fato de o método apresentar resultados satisfatórios mesmo quando utilizado em áreas complexas, com entradas imprecisas ou incompletas (SILVA

2008). Outra vantagem da RNA reside em sua estrutura, que provê uma situação de tolerância às falhas devido ao armazenamento distribuído das informações na rede. (STAHNKE. 2008).

De acordo com Santos (2008), a desvantagem encontrada na utilização de uma RNA relaciona-se ao resultado final encontrado, de forma que não é possível definir como o mesmo foi encontrado, justamente porque a camada onde ocorre o processamento está escondida. KRANZ (2004) afirma que há outra desvantagem das RNAs, sendo esta ligada à alimentação dos dados, pois representações de dados distintos podem produzir resultados diferentes.

Na sequência será apresentado um dos algoritmos mais comuns dentre aqueles que podem ser utilizados em uma rede neural, o back propagation.

3.2.1 Back propagation

O algoritmo de back propagation, também conhecido como algoritmo de retro propagação do erro, é um exemplo de algoritmo de aprendizado supervisionado. Ele possui como objetivo principal, minimizar a função de erro que ocorre entre a saída gerada pela rede neural e a saída realmente desejada pelo usuário. (GOLDSCHMIDT, PASSOS. 2003).

De acordo com Silva (2008), o treinamento necessário na utilização deste algoritmo consiste em duas etapas de processamento distintas. Uma chamada de processamento para frente, *forward*, e outra denominada processamento para trás, *backward*. Em meio ao processamento para frente, um padrão é exposto à camada de entrada da rede. A atividade resultante percorre a rede, buscando camada por

camada, até que seja produzida pela camada de saída uma resposta. No processamento chamado para trás, é realizada uma comparação da saída encontrada, com a saída desejada para o padrão requerido. No caso do resultado ser incorreto, o erro é calculado, sendo este então propagado a partir das camadas de saída até a camada de entrada. Nesse caminho o processo modifica os pesos das conexões entre as unidades das camadas internas levando em consideração a propagação do erro.

Esse processo repete-se para cada linha no conjunto do treinamento. A cada iteração que atravessa todas as linhas que compõem o conjunto de treinamento dá-se o nome de *epoch*. O conjunto de treinamento será utilizado repetidas vezes, a fim de que a taxa de erro não sofra mais diminuição. Quando este ponto é alcançado, diz-se que a rede neural é considerada como um treinamento para encontrar um padrão dentro do conjunto de teste (TWO CROWS CORPORATION 1999).

Uma característica a ser destacada encontrada nas redes neurais, é o fato de que para que elas possuam bons resultados, se faz necessária uma extensa quantidade de treinamento. Isso significa que o processo precisa de muitos dados, e bastante tempo para que obtenha capacidade suficiente, a menos que o problema a ser resolvido seja muito simples. No entanto, após longo treinamento, uma RNA consegue proporcionar resultados muito rapidamente. (TWO CROWS CORPORATION. 1999).

No intuito de ilustrar a capacidade de melhora no desempenho de uma RNA através de massivos treinamentos é mostrado o gráfico 1. Nele pode-se perceber que à medida que são realizados mais treinamentos, com uma maior quantidade de dados, é menor a taxa de erro apresentada. De forma que quanto mais a RNA passar por treinamentos longos, com muitos dados, mais preparada ela está para realizar as tarefas com dados válidos. Cabe ressaltar que existe um ponto onde a rede neural atinge seu ápice em relação à diminuição dos erros apresentados,

quando essa taxa é obtida, ela está realmente pronta para a utilização com máximo desempenho.

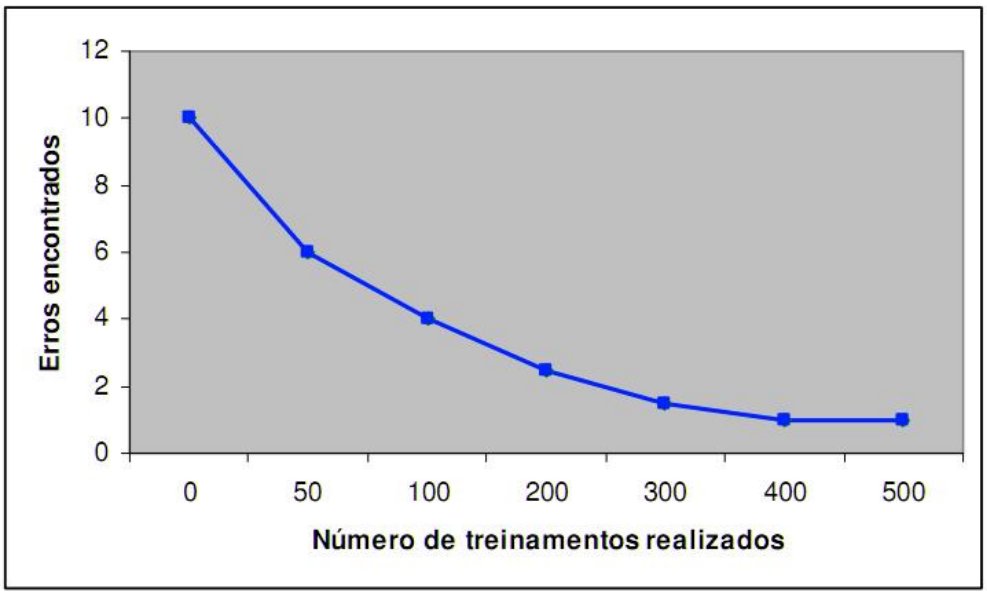

Gráfico 1 - Diminuição da taxa de erro de RNAs utilizando algoritmos de treinamento.

Fonte: Adaptado de TWO CROWS CORPORATION, 1999 p. 13.

3.3 ALGORITMOS GENÉTICOS

De acordo com Goldschmidt e Passos (2005), são conhecidas como algoritmos genéticos, as técnicas que procuram encontrar boas soluções para problemas complexos por meio da evolução de soluções codificadas em cromossomos artificiais. O processo tem uma abordagem adaptativa, já que as soluções existentes a cada instante influenciam na busca por futuras soluções. O paralelismo que é encontrado no processo deve-se ao fato de que as mais **diversas** soluções são consideradas a cada momento pelos algoritmos genéticos.

São algoritmos que buscam simular o processo de seleção natural proposto por Charles Darwin em 1859. Segundo Darwin, a seleção natural é um processo que privilegia aqueles organismos que melhor se adaptam ao ambiente em que estão expostos. Desta forma, quanto mais adaptado ao meio ambiente que se encontra, maior é a chance de que este organismo sobreviva, e transmita aos seus sucessores suas características através dos cromossomos. Os Algoritmos Genéticos (AG) baseiam-se nessa mesma teoria para desenvolver seus modelos. Diversos modelos são estudados, porém apenas aqueles que se mostram mais habilitados para encontrar a solução desejada serão desenvolvidos (AMARAL. 2001).

De acordo com Two Crows Corporation (1999), os algoritmos genéticos recebem esse nome por seguirem padrões encontrados na evolução biológica, onde os membros de uma mesma geração competem para transmitir suas características para a próxima geração, até o ponto onde seja encontrada a geração mais próxima da perfeição. A informação que é transmitida às novas gerações de organismos está contida nos “cromossomos”, que possuem os parâmetros necessários para a criação do modelo.

De uma forma simplificada, o aprendizado genético se dá a partir da criação de uma população inicial, constituída com base em regras criadas de maneira aleatória. Cada uma dessas regras pode ser representada como uma cadeia de bits.

Para exemplificar, supondo-se que as amostras obtidas em um determinado conjunto de treinamento sejam descritas através de duas variáveis booleanas, A1 e A2, e que existam duas classes, C1 e C2. Com esse ambiente, a seguinte regra “Se A1 e não A2 então C2” poderia ser codificada como a sequência de bits “100”, onde só os dois primeiros caracteres da esquerda representam os atributos A1 e A2 e o último representa a classe. Da mesma forma, a regra “Se não A1 e não A2 então C1”, é codificada como “001”. Se um determinado atributo possui K valores, então K

bits deverão ser utilizados para codificar os valores que este possuir. (HAN, KAMBER. 2001).

Como a teoria dos algoritmos genéticos orienta cada um dos indivíduos componentes da população gerada, acaba por representar uma solução para o problema. Isso acontece pois os indivíduos são formados por cromossomos, sendo estes, por sua vez, cadeias de bits que apresentam uma solução. A formação dessa cadeia de bits é realizada através das regras comentadas anteriormente. Com essa ideia central, a solução que se obtém através dos algoritmos genéticos é produzir, por meio das regras específicas e necessárias, uma grande quantidade de indivíduos, formando uma população. Essa população cresce a fim de que se obtenha a maior variedade possível de soluções para o problema. Caso não haja uma solução para o problema, a população é novamente gerada, mas agora pelo processo de reprodução. Esse processo de reprodução possui três módulos: mutação, reprodução e cruzamento.

Na figura 14 apresenta-se uma estrutura simplificada de um algoritmo genético. Dentro dele cada uma das iterações equivale à execução das operações básicas como: seleção do indivíduo, avaliação do seu cromossomo, verificação se a solução do problema proposto foi encontrada e criação de uma nova população se necessário for.

Figura 14 - Estrutura básica de um Algoritmo Genético.

Fonte: Adaptado de COX, 2009, p. 244.

4 PROPOSTA DE TRABALHO

Este capítulo abrange os tópicos de trabalhos relacionados e a proposta de trabalho desenvolvida. Serão apresentadas algumas ideias semelhantes à proposta, uma visão de seu funcionamento, definição e objetivos. Também serão apresentadas algumas diferenças entre esses trabalhos e o projeto proposto.

4.1 TRABALHOS RELACIONADOS

Existem outros trabalhos que possuem foco semelhante ao trabalho apresentado, onde um sistema que lê um determinado texto gera uma representação gráfica das partes consideradas pelo seu algoritmo as mais importantes. Esse resultado gráfico também apresenta as ligações pertinentes entre os termos gerando um grafo com informação relevante. Existem diversos exemplos neste segmento de reconhecimento de palavras e posterior montagem destas em um grafo ordenado. Uma das ferramentas mais conhecidas para mineração de palavras e representação gráfica é o SOBEK.

4.1.1 Ferramenta SOBEK

A proposta básica do SOBEK (LORENZATTI, 2007) é a mineração de textos, para resultar em uma forma gráfica dos termos relevantes, buscando facilitar o entendimento dos assuntos tratados. A ferramenta utiliza estatística para, a partir do conteúdo informado, estabelecer uma base de conhecimento que pode ser posteriormente aprimorada. Essa base tem como componentes, os principais conceitos obtidos nos textos passados como parâmetro. Com essa base de dados,

são gerados grafos que apresentam os conceitos e seus respectivos relacionamentos. (SCHENKER, 2003).

O SOBEK foi construído utilizando a linguagem de programação Java. Devido à necessidade de gerar os grafos, foi utilizada uma API dedicada, a *Interactive Graph Drawing* (ERLINGSSON; KRISHNAMOORTHY, 1996). Sua construção foi projetada para realizar mineração de textos de duas formas, inserindo textos diretamente, ou utilizando uma base de conceitos. Uma tela inicial do SOBEK é apresentada na figura.

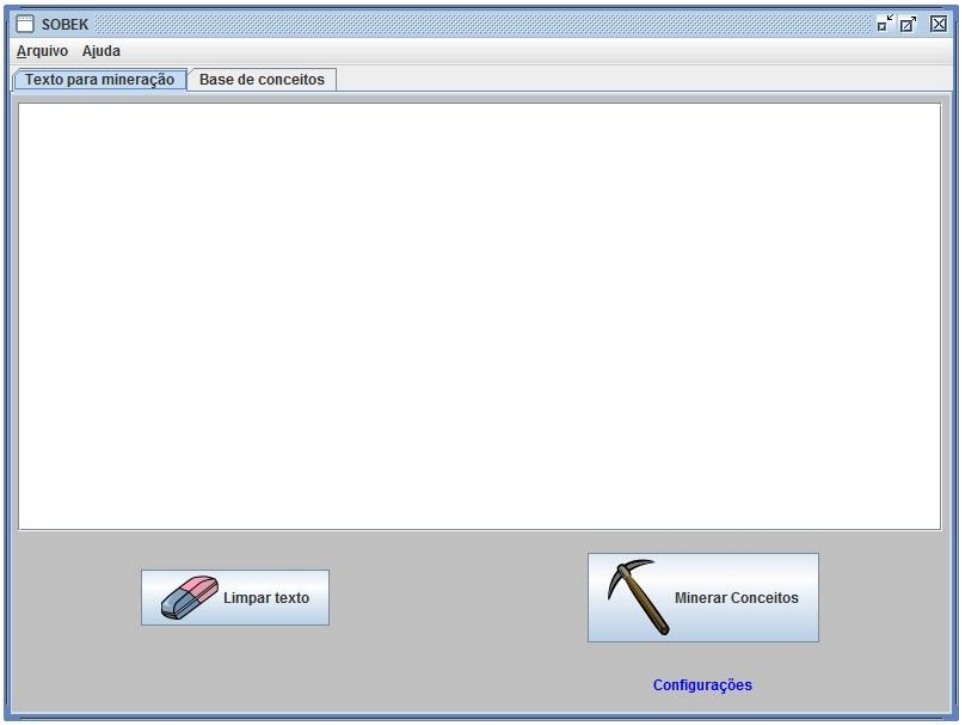

Figura 15 - Interface da ferramenta SOBEK.

Fonte: LORENZATTI, 2007

A ferramenta de mineração de dados SOBEK foi construída para trabalhar com textos sem imagens ou formatações, ou seja, o texto em seu formato puro (LORENZATTI, 2007). Existem duas formas de uso da ferramenta: uma é minerando

textos diretamente e a outra funciona a partir do cadastro de uma base de conceitos para que, a partir destes, seja gerado o grafo.

4.1.2 Geração de mapas conceituais a partir de textos

Existem diversas propostas de trabalho que se baseiam no parse de palavras de algum texto a fim de gerar suas representações gráficas. Devido ao fato da representação gráfica do texto ser um mapa conceitual, será comentado o trabalho de Pérez (2005).

O trabalho de Pérez (2005) utiliza como ferramenta de parser o software PALAVRAS, desenvolvido para o português por Eckhard Bick (Bick, 2000 apud Pérez), realiza as etapas de tokenização, processamento léxico-morfológico, e a fase de análise sintática. A seleção de estruturas sintáticas é extraída em triplas formadas por argumento-relação-argumento.

O trabalho resultou em bons mapas conceituais em forma textual. A autora utilizou a ferramenta CmapTools (IHCMConcept, 2011) para exibir os resultados. Essa ferramenta é bastante conhecida e difundida pelas pessoas que se utilizam de mapas conceituais.

4.2 PROPOSTA DA FERRAMENTA

O intuito deste trabalho é elaborar, ao final do projeto, um software que consiga gerar, a partir de um texto técnico em português, um mapa conceitual que ligue conceitos pertinentes do tema contido no texto. A etapa mais importante desse processo está na fase de leitura do texto, "*parse*" das palavras encontradas e

mineração destas, obtendo, a partir disso, os conceitos que serão utilizados como base do mapa conceitual.

Embora existam trabalhos com focos semelhantes, onde um sistema que lê um determinado texto e, após isso, gera uma representação gráfica das partes consideradas pelo seu algoritmo as mais importantes. O trabalho em questão buscará realizar uma solução completa, de leitura compreensão do texto através de algoritmos de *text mining* e posteriormente a geração de um mapa conceitual relevante baseado nas informações passadas.

Destaca-se, no contexto do projeto, a necessidade de utilizar listas de *stopwords* que serão úteis para filtrar aquelas palavras não relevantes ao contexto, uma solução diferenciada do ponto de vista gráfico. Comparando a proposta com os trabalhos relacionados, diferentemente do SOBEK, a ferramenta tem como objetivo apresentar um mapa conceitual dos termos relevantes encontrados no texto e não um simples grafo. Diferente daquilo que foi descrito por Pérez (2005), a ferramenta irá proporcionar todo o processo desde a captura textual até a exibição do mapa conceitual extraído, não sendo necessária uma intervenção do usuário para elaborar o mapa.

Embora a premissa do desenvolvimento da ferramenta seja que utilizando apenas textos informados, seja possível apresentar um mapa conceitual condizente, o software irá permitir alterações posteriores no mapa. Entre essas mudanças permitidas constam a adição de novas arestas e vértices, bem como ajuste de proporções, ligações e definições pré-estabelecidas. Como não se pode esquecer que o intuito base é o cunho educacional de apoio ao estudo, dessa maneira a liberdade de ajuste e mudança a partir daquilo que é gerado é bem vinda. Permitindo que mesmo nas oportunidades onde o software não consiga gerar um bom resultado, seja possível modificá-lo para melhorar sua resposta final.

5 FERRAMENTA MAPA EXTRATOR

Com o objetivo de executar algumas etapas descritas neste trabalho, o autor criou o projeto MapaExtrator. Esse projeto, que tem como premissas ser uma ferramenta de simples utilização, e de código aberto, foi desenvolvido ao longo do período do trabalho de conclusão, sendo inteiramente idealizado na plataforma de tecnologia JAVA 6. A escolha da tecnologia JAVA foi natural devido ao prévio conhecimento do autor, e do orientador na mesma. Além disso, como o projeto visa uma facilidade de utilização e execução, o JAVA permite criar uma aplicação multiplataforma, podendo ser executada nos mais diversos sistemas operacionais. Outro ponto a favor é a grande comunidade ativa de desenvolvedores desta tecnologia, pessoas que, interessadas no projeto, poderão colaborar com futuras melhorias e modificações.

Outra premissa deste projeto é idealizar uma aplicação que não necessite de uma máquina de alto desempenho para funcionar, nos testes efetuados, até mesmo uma máquina com de mais de dez anos de idade e configuração defasada executou o MapaExtrator sem problemas. Como seu intuito é colaborar com a educação de alunos e ajudá-los a compreender melhor a utilização de mapas conceituais nos seus estudos, o aplicativo será mantido com uma licença de código aberto *General Public Licence* ou simplesmente GPL (FREE SOFTWARE FOUNDATION, 2007). Desta forma além de permitir o uso irrestrito, também mantém se a ideia de colaboração por parte de algum usuário que assim o queira modificar podendo até mesmo ser utilizado como apoio didático em aulas de programação, onde novas melhorias podem ser realizadas.

Para organização do projeto, além de centralizar os esforços de desenvolvimento em apenas um local, foi criado um projeto no portal *http://sourceforge.net* com o nome de MapaExtrator. Assim, qualquer pessoa interessada pode entrar em contato, acompanhar o projeto ou mesmo fazer o

download do aplicativo acessando o site gerado apenas para o projeto. A página inicial do projeto MapaExtrator pode ser acessada através do link *http://mapaextrator.sf.net/*. O projeto atualmente conta com constante atualização de seu código fonte, o qual pode ser acompanhado através da área de desenvolvimento que permite também o download do código fonte utilizado.

5.1 DESCRIÇÃO DO PROJETO MAPA EXTRATOR

Como o projeto ainda continua em desenvolvimento e sofrendo constantes melhorias, uma solução simplificada para explicar seu funcionamento, é valer-se de uma forma simplificada de seu diagrama de classes. Para que a partir deste todo, seja explicado de forma simples e concisa o necessário a respeito do funcionamento do projeto. Essa visão geral do MapaExtrator utilizando ponto de vista de desenvolvimento pode ser acompanhada já que apresenta as principais classes geradas no processo e suas ligações.

5.1.1 Classe MapaExtrator

Para o desenvolvimento do projeto foi idealizada como classe principal a de nome MapaExtrator. A partir dela partem as requisições às demais componentes do software e nela se encontra grande parte do processo lógico desenvolvido. Sendo essa classe uma extensão de um JFrame ela também tem a responsabilidade de controlar a janela que é exibida ao usuário. Podemos ter uma ideia visual da utilização da classe MapaExtrator verificando, na figura 15, a quantidade de ligações entre ela e outras classes componentes do projeto.

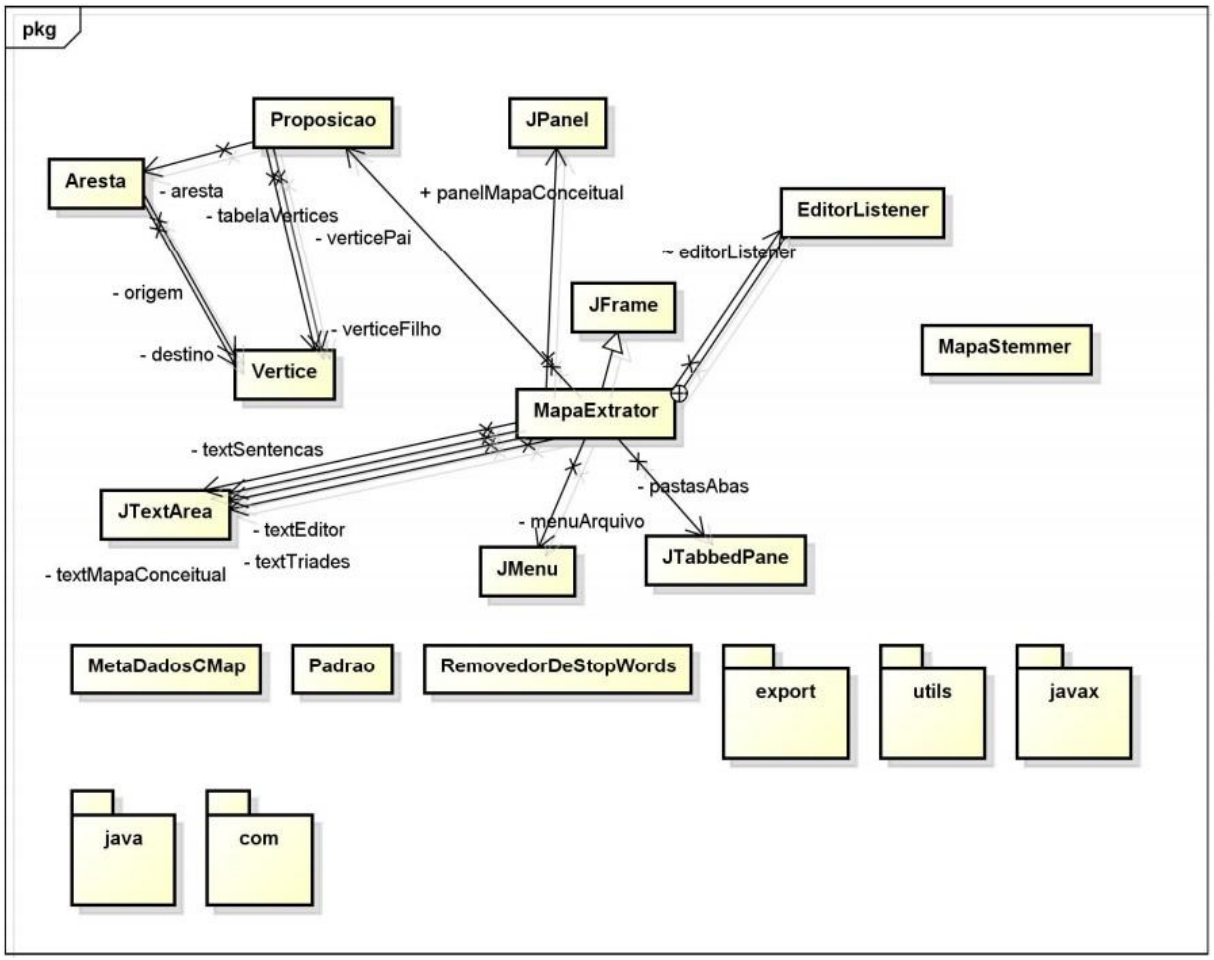

Figura 16 - Diagrama de classes simplificado do MapaExtrator.

Fonte: Autor

5.1.2 Classe Proposição

Conforme comentado anteriormente proposição é o nome dado ao conjunto de dois vértices ligados por meio de uma aresta. Essa mesma abordagem foi repetida na criação de sua classe em Java. Sendo que esta possui como filhas duas instâncias da classe de definição de vértice e uma instância da classe aresta. Esse conjunto, três componentes é diversas vezes relacionado no código através do

nome tríade. A partir do conjunto dessas tríades, que serão validadas e identificadas pode ser gerado o mapa conceitual.

5.1.2.1 Classe Vértice

Com o intuito de ser a mais simples representação de um vértice em formato de objeto Java, as instâncias dessa classe possuem como atributos os seguintes valores: Id, nome, coordenada X, coordenada Y, largura, altura e estilo. Isso será usado tanto para o reconhecimento de cada vértice através de seu identificares únicos, como o estilo e de que maneira cada item deste tipo estará disposto na tela.

5.1.2.2 Classe Aresta

Da mesma maneira que a classe vértice, a instância da classe aresta possui o mais simples formato representativo. El a se vale dos seguintes atributos: id, objeto pai, nome, origem, destino e estilo. Como se pode observar existe a figura de um objeto pai, que irá definir a localização dessa aresta, além das indicações de origem e destino, que no caso serão objetos do tipo vértice que essa aresta estará unindo.

5.1.3 Classe RemovedorDeStopWords

Esta classe como seu próprio nome enfatiza é utilizada na fase prévia de analise do texto inserido, fazendo uma varredura neste e removendo dele aquelas palavras que não irão afetar o entendimento da mensagem informada. Essa limpeza de palavras com baixo valor no significado do texto irá facilitar o processo de *matching* realizado posteriormente.

5.1.4 Classe Padrão

Essa é a classe que contém a essência do *parser* de significado das sentenças. É por meio dela que o processo de *pattern matching* acontece, fazendo com que sentenças sejam decompostas em pequenas partes e validadas em cada um dos padrões estabelecidos. Esse passo tem como objetivo identificar algum comportamento comum entre as sentenças de indicação relevantes em um texto.

Seu funcionamento inicia-se com o carregamento antecipado dos padrões conhecidos cadastrados no sistema. Entende-se como padrão neste momento um objeto composto formado por um conjunto de atributos entre eles: uma lista de itens para serem encontrados (*match*), e uma ou mais listas de saídas geradas a partir do formato de dado encontrado.

Exemplificando sua usabilidade, será explicado como definir um padrão simples, que encontre uma afirmação igualmente simples. Supondo a necessidade de um padrão que identifique uma afirmação composta por “é um” ou tendo no restante da sentença o sujeito e seu predicado, devemos proceder da seguinte forma: deve ser gerado um novo padrão, no primeiro item deste, informamos o formato esperado da sentença, que no caso será: “#1”, “é”, “um”, “#2”. O que se lê deste primeiro parâmetro é que aquilo definido por extenso serão os itens buscados, no caso as palavras “é” e “um” nesta sequência. Enquanto que os itens definidos numericamente precedidos por “#” serão compostos pelas palavras restantes na sentença, obtidas anterior e posteriormente ao *match* “é um”. Ao buscar o padrão recém criado na seguinte frase: “O MapaExtrator é um software”, o resultado alcançado é o seguinte: acontece o casamento da definição “é um” referindo assim “O MapaExtrator” como item do sujeito (#1), o restante (#2) é o predicado “software”.

A questão de encontrar um padrão definido é apenas o começo do processo, já que tão importante quando identificar é associar qual será o próximo passo a ser realizado. No exemplo simples apresentado, o caminho a ser usado pelas palavras encontradas será o de sua composição original, representado da seguinte forma: ”#1”, “é um”, ”#2”. Isso define uma tríade já finalizada que será posteriormente transformada em proposição, tendo como vértices os itens numerados e usando de aresta as palavras de ligação “é um”. Cabe ressaltar que anteriormente ao processo de geração da proposição, a sentença passa pela etapa de remoção de *stop words* que neste caso removerá o artigo inicial.

A fim de ter uma visualização real de como uma definição de padrão acontece dentro do software, abaixo será apresentado um padrão que busca o exemplo proposto. Os dois últimos parâmetros informados como *null* abrigariam novas possibilidades de conceitos a serem encontrados a partir do *match* inicial, se necessário for.

```
Padrao pSer1075 = new Padrao(
      new String [] { "#1", "é", "um", "#2"},
      new String [] { "#1", "é um", "#2" },
      null,
      null );
```

O resultado final, os conceitos gerados com esse padrão, ainda seguindo nosso exemplo, seria composto pela tríade: “mapaextrator”, “é um”, “software”. Com esse formato normalizado, ele pode ser passado ao processo de geração de proposição, que por sua vez, invoca o método gerador de mapas.

5.1.5 Classe MapaStemmer

Embora ainda não com sua funcionalidade total, essa classe ganhou forma com a finalidade de identificar palavras que, embora possuam diferenças em sua grafia, mantenham um mesmo significado geral. A classe provê o processo de stemming conforme explicado anteriormente. Devido ao processo de identificação de radicais ainda não estar completamente produzido, foi definido que ela não seria utilizada por enquanto.

5.1.6 Classe MetaDadosCMap

Para prover uma flexibilidade ao software, foi planejada a opção de exportar os mapas conceituais obtidos para o formato Concept Mapping Extensible Language ou CXL (IHMConcept, 2011). Esse formato, baseado em XML, é aceito como formato de importação de dados para a ferramenta de manipulação de mapas conceituais CMapTools (IHMConcept, 2011). Essa ferramenta é bastante conhecida e difundida pelas pessoas que se utilizam de mapas conceituais. O formato é bastante complexo, mas inicialmente foi codificado um método que abrange as mais básicas necessidades, como a geração de vértices e arestas apenas. Foi deixada de lado por ora a questão de agrupamento e de controle avançado de estilo que o formato poderá prover no futuro.

5.1.7 Outras classes

As demais classes apresentadas, como JFrame, JTextArea e outras que foram suprimidas do diagrama englobam o contexto da interface ao usuário,

provendo as janelas, botões e áreas de edição. Existem também as classes de utilidades, para geração de XML, conversão de formatos e controle de utilização com mouse e teclado. Todas as classes de utilidades ficaram no pacote nomeado *util*, enquanto as referentes à exportação do arquivo CXL encontram-se no pacote *export*. Por sua vez as classes centrais do projeto estão no pacote padrão, sendo esse definido pelo Java com o nome *src*.

5.2 USO DO MAPA EXTRATOR

Ao se executar o arquivo do MapaExtrator, uma nova janela aparece com espaço de livre edição. Conforme a figura 16, o aplicativo apresenta uma divisão de telas por meio de abas, além de botões, funcionalidades essas que serão detalhadas a seguir. Como se pode observar, o programa tem semelhanças à outros aplicativos de computador com botões para ações, menus de ajuda e uma grande área para a digitação dos textos.

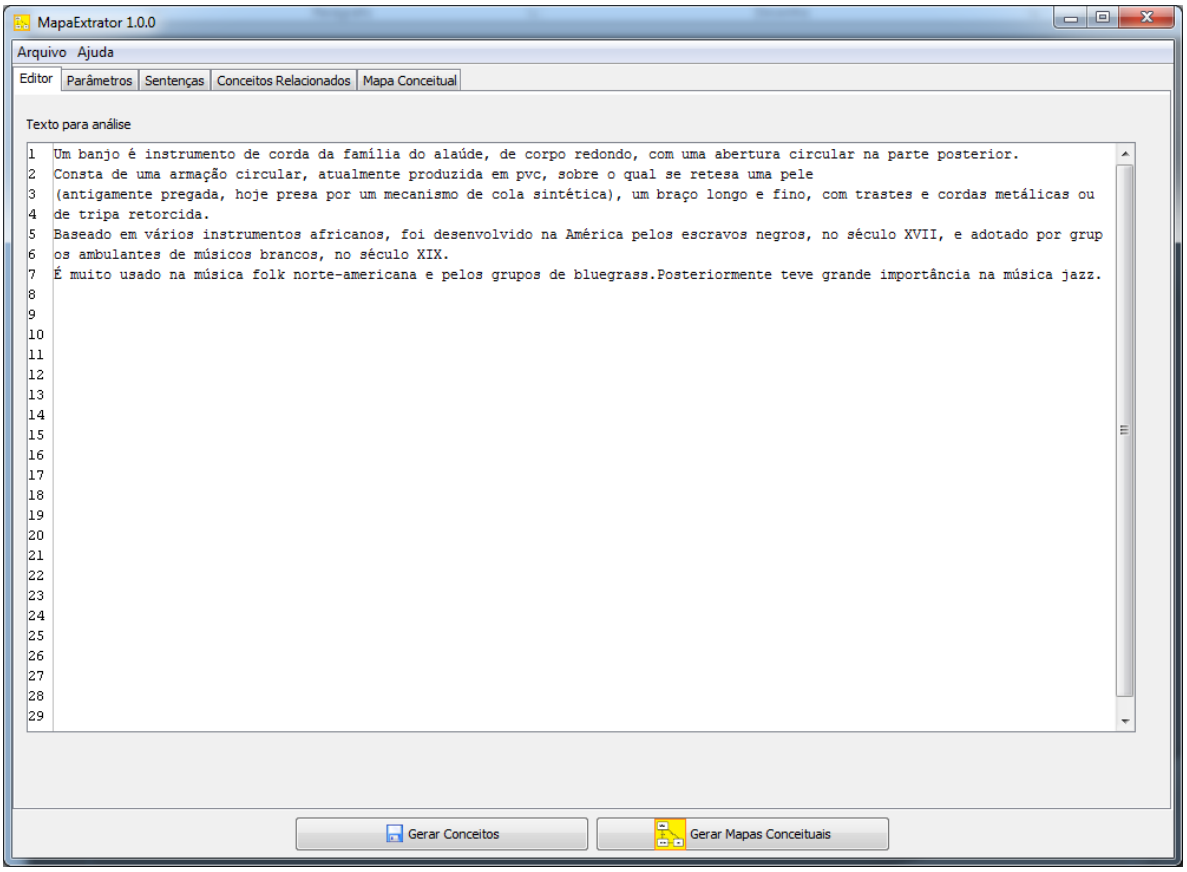

Figura 17 - Imagem inicial do MapaExtrator já com algum texto que será utilizado.

Fonte: Autor

5.2.1 Inserção de texto para a geração de conceitos

Como exemplo inicial do funcionamento do aplicativo e de suas capacidades de verificação de sentenças, na figura 17 pode ser visto que foi inserida uma única frase. O software, partindo desse texto informado, separa as sentenças obtidas para enfim poder realizar uma varredura de padrões, observando assim, os conceitos que poderiam ser equiparados. Para que o usuário chegue nessa etapa, basta inserir na aba “Editor” do software seu texto e, em seguida, clicar no botão “Gerar Conceitos”.

Nessa etapa as sentenças são separadas e em cada uma delas uma lista de *patterns* previamente estabelecida é verificada, proporcionando assim um *pattern matching. Esse match* tem como objetivo básico encontrar situações de interpretação de conceitos relacionados que podem formar uma proposição no mapa conceitual. Utilizando dessa informação, é apresentado ao usuário o resultado dos possíveis conceitos relacionados àquela sentença. Se for encontrado algum conceito válido, isto é, algum *pattern* conseguiu estabelecer uma relação com a sentença seu resultado final é mostrado na aba "Conceitos Relacionados”. Essa exibição de resultados pode ser observada na figura 18.

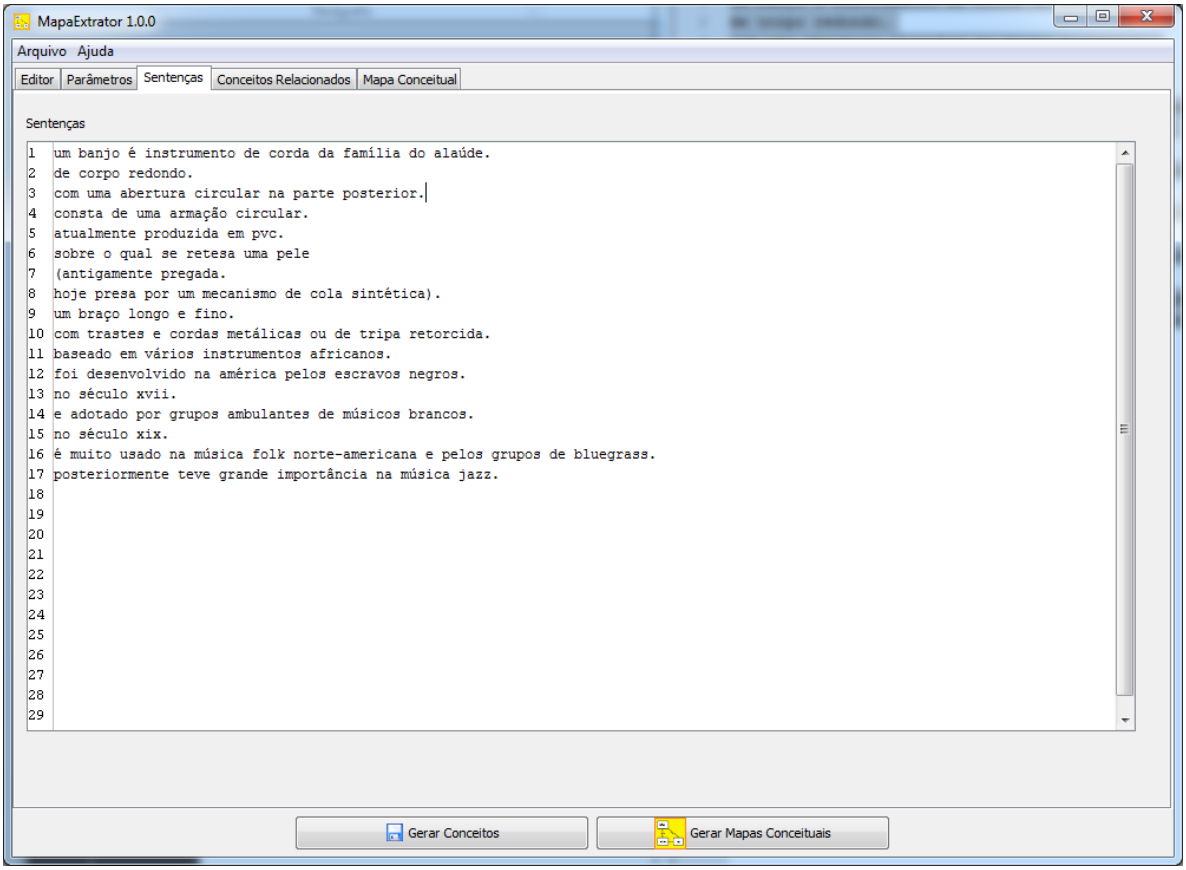

Figura 18 - Tela com as sentenças informadas reconhecidas.

Fonte: Autor

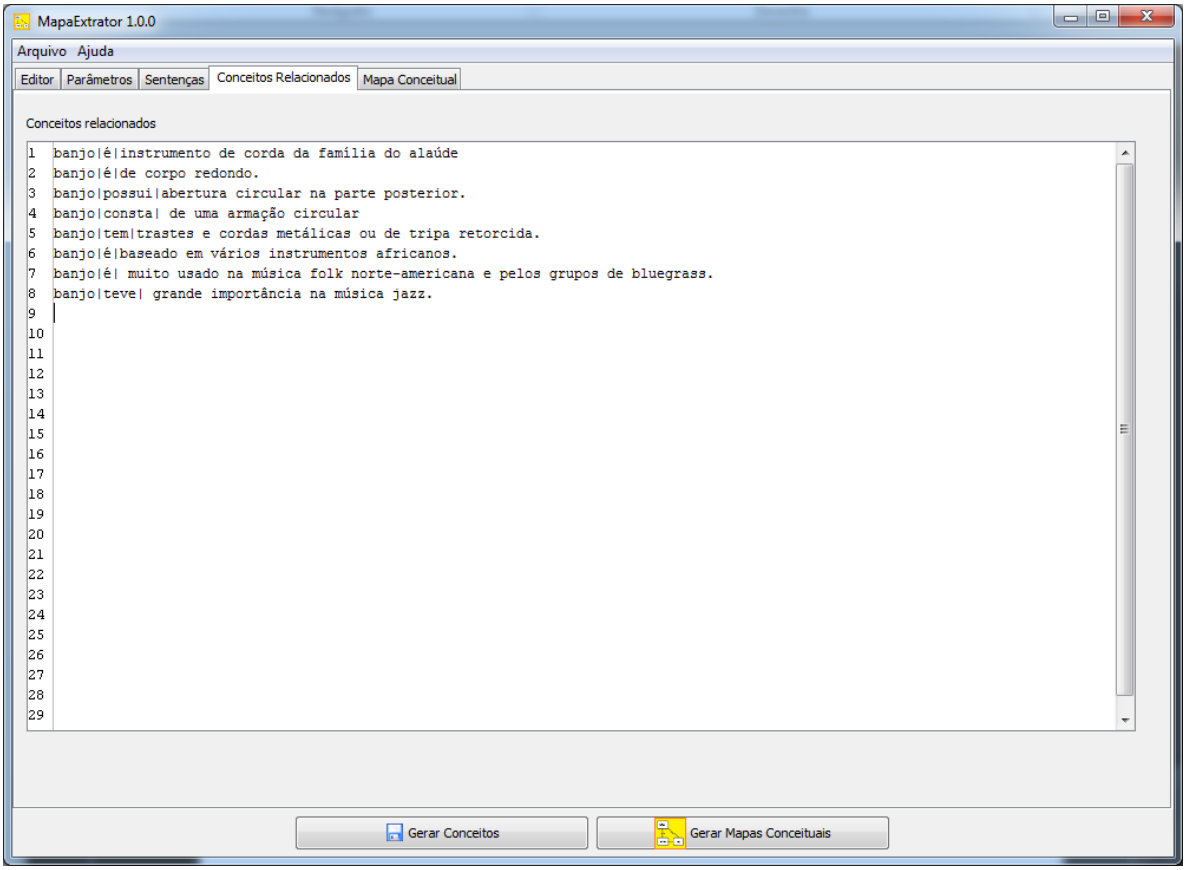

Figura 19 - Tela onde os conceitos relacionados obtidos são mostrados.

Fonte: Autor

5.2.2 Geração do Mapa Conceitual

Após encontrar os conceitos relacionados, o software tem a capacidade de filtrar as informações obtidas a fim de gerar um mapa conceitual adequado. O mapa conceitual apresenta-se na forma de um grafo ordenado, criado a partir de uma API gráfica escrita em JAVA chamada JGraphX da companhia JGraph Ltd. (2010). Utilizando essa API conseguimos um bom desempenho visual, com uma relativa simplicidade de desenvolvimento, o que contribui para a melhoria contínua do aplicativo.

Seguindo com o exemplo apresentado anteriormente, se o usuário após gerar os conceitos relacionados, clicar no botão “Gerar Mapas Conceituais” o software apresenta sua ultima aba com a imagem do mapa conceitual, representando os conceitos encontrados. Na figura 19 podemos observar o mapa conceitual obtido. Embora o intuito do software seja prover um mapa conceitual adequado imediatamente após a análise do texto, esse mapa apresentado é editável ao nível de ordenação na tela. O usuário é livre para arrastar os objetos presentes, como vértices e arestas a fim de ajustar ao seu gosto, ou mesmo realizar alguma tarefa de adequação do mapa gerado.

Dando mais liberdade ao usuário foi disponibilizada também a opção de inserir manualmente novas formas no mapa conceitual, podendo o usuário, a qualquer momento modificar os itens do mapa, bem como inserir novos ou apagar itens que não corresponderem à expectativa. Suas mudanças vão desde a estética mudança de fonte, cor ou tamanho, à total reordenação e modificação do mapa obtido.

Embora sejam possíveis essas modificações manuais é importante lembrar que através do aplicativo, os mapas conceituais podem ser atualizados sempre que necessário. Para que isso aconteça basta apenas realizar a devida mudança nos textos e após isso gerar os conceitos relacionados. Com esses conceitos atualizados armazenados, um novo clique no botão gerador de mapa faz com que o antigo mapa seja apagado, dando lugar ao novo grafo que possui as representações dos últimos conceitos obtidos. No caso de um mapa ainda não salvo, o programa oferece a gravação do arquivo atual, evitando a perda do trabalho já realizado.

Conforme o texto for sendo modificado, ele pode acabar caindo em mais *patterns* que invariavelmente representam mais conceitos relacionados o que aumenta gradativamente a complexidade do mapa a ser gerado.

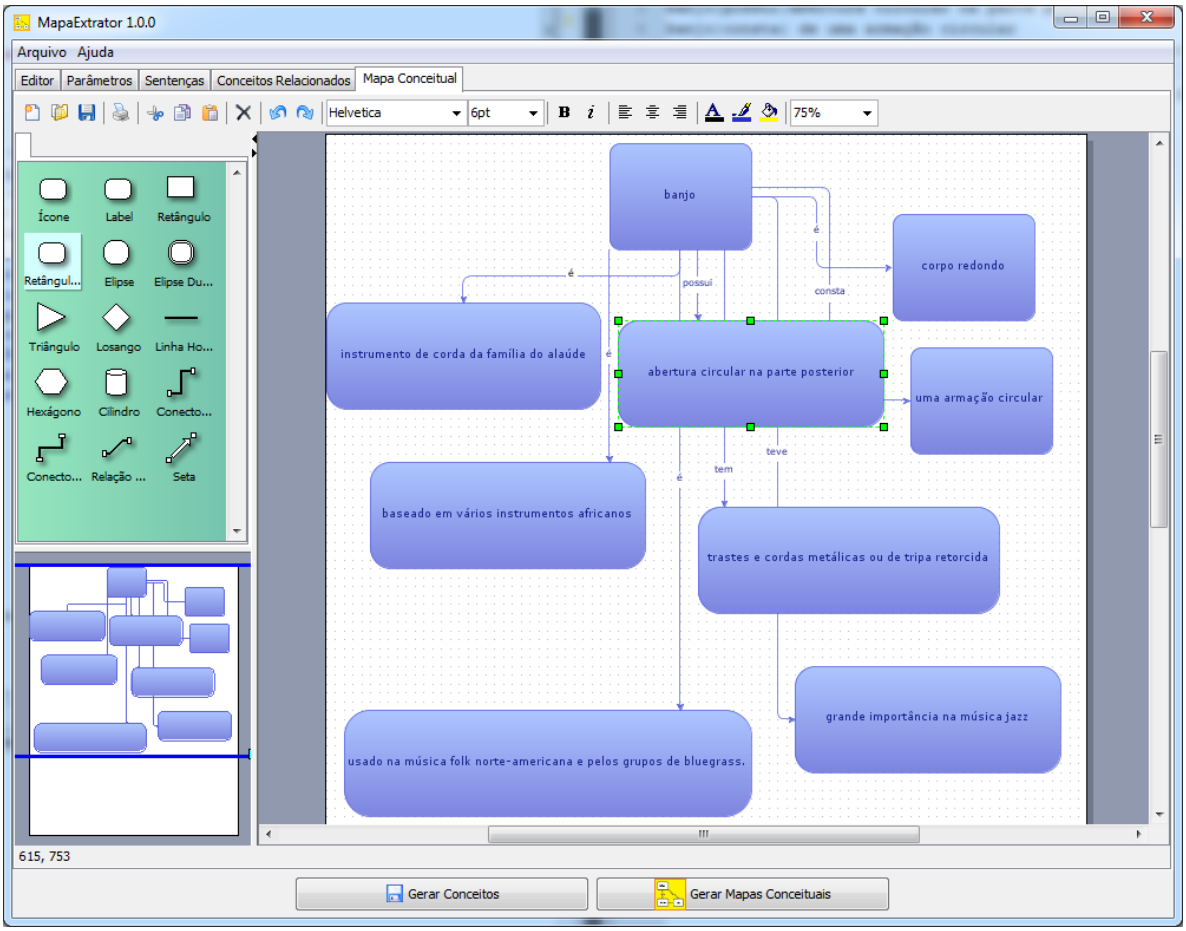

Figura 20 - Mapa conceitual obtido a partir de uma frase que gerou um conceito.

Fonte: Autor

CONSIDERAÇÕES FINAIS

Este trabalho apresentou o estudo bibliográfico para a criação de uma ferramenta de geração de mapas conceituais a partir de textos, tendo como aspecto principal a facilidade de uso.

O estudo bibliográfico realizado mostrou que o uso de mapas conceituais no apoio da aprendizagem em diferentes contextos educacionais, pode ser uma ferramenta interessante. Dessa forma, surge a necessidade de uma ferramenta que possa gerar modelos de mapas conceituais a partir de textos. Essa ferramenta ajudaria professores e alunos a gerar novos mapas baseados em textos educacionais, esses mapas embora talvez não representem a melhor forma de representação gráfica possível, seria um caminho para que os alunos pudessem completar ou modificar aquilo que acharem necessário.

Indo ao encontro à necessidade apresentada, foi iniciado o projeto MapaExtrator, que obteve como resultado um software em Java capaz de identificar padrões previamente definidos em textos. A partir dessas definições ele foi capaz de apresentar mapas conceituais que podem ser modificados, melhorados e salvos em diversos formatos. Um desses formatos, o CXL é também um formato aceito pelo programa CMapTools, uma ferramenta de mapas conceituais muito conhecida e utilizada.

O que pôde ser observado, nas ferramentas de geração de grafos a partir de texto analisadas, foi toda a complexidade que compõe o processo de parse de informações e geração dos grafos. Sendo essa uma das partes mais complexas do projeto, o estudo dos dados informados a fim de encontrar padrões e relevância. Foi e continuará sendo no decorrer das melhorias na ferramenta, um dos maiores desafios no projeto.

Existem algumas diferenças entre as propostas do MapaExtrator e do SOBEK no que diz respeito à capacidade de geração de um mapa conceitual mais organizado e hierárquico, bem como poder-se utilizar de uma API gráfica mais robusta que permitiu novas opções de utilização além da capacidade de exportação em diversos formatos. Contudo, a facilidade de uso do SOBEK foi inspiração para o trabalho.

Um dos aspectos importantes projeto MapaExtrator é a intenção de criar uma ferramenta gratuita, que possa colaborar no desenvolvimento educacional dos alunos que a utilizarem, na geração de novas formas de entendimento para matérias conhecidas através mapas conceituais.

Dentro dessa ideia de colaboração cabe lembrar que o projeto surgiu como um software livre e de código aberto, permitindo que os próprios usuários ou outros interessados possam realizar o download do código-fonte e trabalhar em melhorias e novas funcionalidades. Algumas das melhorias propostas incluem a geração de uma tela de criação de novos padrões, para que estes possam ser inseridos pelo usuário. Outra adição interessante é inserir, durante o processo, uma validação das palavras encontradas por meio de um dicionário de português, dando a opção de ajustar aquilo que por ventura esteja escrito de maneira equivocada.

Assim como nosso idioma português vive em constante modificação, moldando-se de acordo com a necessidade ao passar do tempo, o projeto MapaExtrator ainda possui um longo caminho a percorrer. Esse caminho trilhado lado a lado com as futuras mudanças da língua portuguesa terá como objetivo sempre melhorar sua capacidade de interpretação de texto e compreensão de contexto. Sabe-se que um trabalho com processamento de linguagem natural perfeito seria o estado da arte da codificação, no entanto a vontade é de chegar sempre o mais longe possível com as ferramentas que possam ser utilizadas.

REFERÊNCIAS BIBLIOGRÁFICAS

AMARAL, Fernanda Cristina Naliato. **Data Mining**: Técnicas e aplicações para o marketing direto. São Paulo, SP: Editora Berkeley, 2001.

BARION, Eliana Cristina Nogueira; LAGO, Decio. **Revista de Ciências Exatas e Tecnologia – Mineração de Textos.** Valinhos, SP: 2008. Vol. III, Nº. 3. Anhanguera Educacional S.A.

BUNESCU, Razvan; MOONEY, Raymon J.. **Knowledge from Text Using Information Extraction**. Austin, TX: 1995. University of Texas at Austin. Department of Computer Sciences.

ERLINGSSON, U.; KRISHNAMOORTHY Mukkai. **Interactive Graph Drawing**. Disponível em: <http://www.cs.rpi.edu/research/groups/pb/graphdraw/> Acesso em Out. 2010

FARIA, Wilson de. **Mapas conceituais:** aplicações ao ensino, currículo e avaliação. São Paulo: E.P.U, 1995.

FELDMAN, Ronen; SANGER, James. **The Text Mining Handbook: advanced approaches in analyzing unstructured data**. New York, NY: 2007. Cambridge University Press.

FRANK, Eibe; WITTEN, Ian H.. **Data mining**: Pratical machine learning tools and techniques with Java implementations. San Diego, CA: Morgan Kaufmann, 2000.

FREE SOFTWARE FOUNDATION INC. **GNU General Public Licence.** Disponível em: <http://www.gnu.org/licenses/gpl.html> Acesso em 10 Mar 2011. Publicado em 29 Jun. 2007.

GOLDSCHMIDT, Ronaldo; PASSOS, Emmanuel. **Data Mining:** Um Guia Prático: Conceitos, Técnicas, Ferramentas, Orientações e Aplicações. Rio de Janeiro, RJ Elsevier, 2005.

GOMES, Igor Ruiz; MONTEIRO, Leda de Oliveira; OLIVEIRA, Thiago. **Etapas do Processo de Mineração de Textos – uma abordagem aplicada a textos em Português do Brasil**. Belém, PA: 2006. Artigo. Anais do XXVI Congresso da SBC. Workshop de Computações e Aplicações.

GONÇALVES, Eduardo Corrêa. **Extração de árvores de decisão com a ferramenta de Data Mining Weka.** Artigo disponível em: < http://www.devmedia.com.br/articles/viewcomp.asp?comp=3388> Aceso em 05 Nov. 2010

IHMConcept. **CmapTools.** Disponível em < http://cmap.ihmc.us/download/ > Acesso em 10 Set. 2010

JERÔNIMO, Paulo Marcelo. **Estudo sobre Data Mining. Data Warehouse. Cases – Data Warehouse**. Novo Hamburgo, RS: Monografia (Bacharelado em Ciência da Computação) – Institutos de ciências exatas e tecnológicas, Universidade Feevale, 2001.

JGRAPX LTD. **JGraphX - Java Graph Visualization Library.** Disponível em: < http://www.jgraph.com/jgraph.html > Acesso em 17 Ago. 2010.

LOH, Stanley. **Text Mining por Stanley Log**. Blog disponível em <http://miningtext.blogspot.com/2008/11/listas-de-stopwords-stoplist-portugues.html>. Acesso em 10 Nov. 2010. Publicado em 26 Nov. 2008.

LORENZATTI, Alexandre. **SOBEK: uma Ferramenta de Mineração de Textos** Caxias do Sul, RS: Monografia (Bacharelado em Ciência da Computação) – Departamento de Informática, Universidade de Caxias do Sul, 2007.

NOVAK, Joseph D. **Learning, creating, and using knowledge**: concept maps as facilitative tools in schools and corporations. New Jersey: L. Erlbaum Associates, 1998.

NOVAK, Joseph D.; GOWIN, D. Bob. **Learning how to learn**. Cambridge: Cambridge University Press, 2002.

ONTORIA, Antonio et al. **Mapas conceptuais: uma técnica para aprender**. Lisboa, Portugal: ASA, 1999.

PÉREZ, Cláudia Camerini Corrêa; VIEIRA Renata. **Mapas Conceituais: geração e avaliação.** São Leopoldo, RS: 2005. XXV Congresso da Sociedade Brasileira de Computação.

PRADO, Hércules Antônio; OLIVEIRA, José Palazzo Moreira; FERNEDA, Edilson; WIVES, Leandro Krug; SILVA, Edilberto Magalhães; LOH, Stanley. **Text Mining in the Context of Business Intelligence**. Encyclopedia of Information Science and Technology. Hershey: Idea Group, 2005.

SANTOS, Daiana Pereira. **Mineração em notas fiscais de entrada de empresa calçadista**. Novo Hamburgo, RS: 2008. Monografia (Bacharelado em

Ciência da Computação) – Instituto de Ciências Exatas e Tecnológicas, Universidade Feevale.

SHOLOM, Weiss M.. **Text Mining: predictive methods for analyzing unstructured information.** New York, NY: Springer, 2005.

SILVA, Cláudio Aurélio. **Descoberta de conhecimento em uma base de dados de uma academia.** Novo Hamburgo, RS: 2008. Monografia (Bacharelado em Ciência da Computação) – Instituto de Ciências Exatas e Tecnológicas, Universidade Feevale, 2008.

SILVA, Edilberto Magalhães. **Descoberta de Conhecimento com o uso de Text Mining: Cruzando o Abismo de Moore**. Brasília, DF: 2002. Dissertação (Pós-graduação em Gestão do Conhecimento e da Tecnologia). Universidade Católica de Brasília.

STAHNKE, Fernando Rafael. **Uso de Data Mining no mercado financeiro**. Novo Hamburgo, RS: 2008. Monografia (Bacharelado em Ciência da Computação) Instituto de Ciências Exatas e Tecnológicas, Universidade Feevale.

TONIAZZO, Lucas Hernani Giovenardi. **Módulo de recuperação de conhecimento para auxílio na tomada de decisão em um sistema de ouvidoria**. Novo Hamburgo, RS: 2005. Monografia (Bacharelado em Ciência da Computação) – Instituto de Ciências Exatas e Tecnológicas, Universidade Feevale.

TWO CROWS CORPORATION. **Introduction to Data Mining and Knowledge Discovery**. Potomac, MD: 1999.