MARIA ANGELA MOSCALEWSKI ROVEREDO DOS SANTOS

EXTRAINDO REGRAS DE ASSOCIAÇÃO A PARTIR DE TEXTOS

Dissertação apresentada ao Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática Aplicada.

Área de Concentração: *Sistemas Inteligentes*

Orientador: Prof. Dr. Alex Alves Freitas

CURITIBA

2002

Santos, Maria Angela Moscalewski Roveredo dos
Extraindo regras de associação a partir de textos. Curitiba, 2002. 51 p.

Dissertação (Mestrado)– Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática Aplicada.

1. Regras de associação 2. Mineração de textos 3. Recuperação de informações I.Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de Tecnologia. Programa de Pós-Graduação em Informática Aplicada II-t

A Deus,
que sempre ajuda a
quem se ajuda.

À minha mãe,
Therezinha Moscalewski Roveredo, exemplo pessoal
e profissional, sabedoria e generosidade.
Que me deu forças para partir, mas que partiu
antes de minha chegada,
deixando uma profunda e imensa saudade.

Ao meu pai,
Ledi Roveredo,
exemplo de força, amor e fé.
Ofereço esta dissertação.

Agradecimentos

O meu profundo agradecimento ao Doutor Alex Alves Freitas pelas excelentes contribuições científicas e orientações recebidas, em especial, pelo respeito à minha construção intelectual, na qual pude revelar minhas idéias, meus ideais, minhas crenças e minhas esperanças.

O meu agradecimento é extensivo aos Doutores Celso Antônio Alves Kaestner e Júlio Cesar Nievola. Sou grata pela compreensão e condições de trabalho, o que permitiu a continuidade desta dissertação.

Agradeço, também, à Pontifícia Universidade Católica do Paraná, por intermédio de seu Programa de Pós-Graduação em Informática Aplicada, pela minha formação acadêmica e pelo reencontro com minha área de atuação.

Ao profissional de informática e amigo Marcos Afonso Debur, o meu agradecimento às contribuições recebidas.

À Doutora Marta Morais da Costa e à Mestra Maria Antonia Moscalewski Schuartz, o meu reconhecimento pela palavra amiga, pelo carinho e pela amizade nos momentos de alegria e de dor.

Serei eternamente grata à minha família e amigos verdadeiros pela compreensão de minhas ausências, pelo constante incentivo, pelo companheirismo, pelo amor e pela generosa amizade.

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

O objetivo desse projeto é extrair automaticamente regras de associação entre "itens de informação" (palavras) contidos em documentos, aonde cada documento será considerado uma "transação". Para que as regras descobertas sejam mais relevantes para o usuário, os "itens de informação" ocorrendo nas regras são limitados a palavras de determinado tipo, em particular substantivos. Além disso, o sistema utiliza a medida de *Tf-Idf* (*Term Frequency – Inverse Document Frequency*), da área de recuperação de informações, para estimar quais palavras são mais interessantes para o usuário. O sistema foi utilizado para extração de regras de associação a partir de documentos da base *Tipster* [Tipster 94], uma base padrão na literatura de recuperação de informações. Os resultados revelaram a necessidade de uma medida de interesse das regras descobertas, o que resultou em uma proposta de uso do *WordNet* para avaliação do interesse das regras descobertas.

Palavras-Chave: Regras de associação, Mineração de textos, Recuperação de informações

Abstract

The goal of this work is to automatically extract association rules from "information items" (words) contained in documents, where each document will be considered as a “transaction”. In order to improve the relevance of the discovered rules, the "information items" occurring in the rules are limited to a certain kind of words, in particular nouns. Furthermore, the system uses the *Tf-Idf* (Term Frequency – Inverse Document Frequency) measure, from the information retrieval area, to estimate which words are more interesting for the user. The system was used to extract association rules from documents of the *Tipster* [Tipster 94] base, a standard document base in the literature on information retrieval. The results revealed the need for a measure of interest of the discovered rules, which has led to a proposal for using *WordNet* to evaluate the interestingness of the discovered rules.

Keywords: Association rule, Text mining, Information retrieval.

Capítulo 1

Introdução

Data Mining tem sido reconhecida como uma nova área de pesquisa interdisciplinar, envolvendo aprendizado de máquina, estatística, banco de dados e outras áreas [Berson & Smith 97]. Tal área pode ser definida como "descoberta eficiente de regras interessantes em grandes volumes de dados" [Srikant & Agrawal 95].

Esse trabalho aborda uma variante de *data mining*, chamada *text mining* (mineração de textos). Essa é uma área de vital importância, tendo em vista o acúmulo rápido e contínuo da quantidade de texto armazenada eletronicamente.

A descoberta (semi-) automática de regras que capturem interrelacionamentos entre itens de informação em um texto pode ser bastante importante para a tomada de decisões estratégicas.

A extração de regras de associação é uma técnica de *data mining* que gera regras do tipo "Se X Então Y" a partir de um banco de dados de transações, onde X e Y são conjuntos de ítens que co-ocorrem em várias transações. Por exemplo, a regra "Se batata_frita Então refrigerante" pode ser automaticamente descoberta em um banco de dados de transações de um supermercado, onde cada transação corresponde à compra realizada por um determinado cliente em um determinado momento. Essas regras podem ser bastante úteis para tomada de decisão. No exemplo acima, elas podem ser usadas, por exemplo, para aumentar as vendas de refrigerantes, colocando esses produtos em prateleiras próximas às prateleiras das batatas fritas.

Este trabalho tem como objetivo extrair automaticamente regras de associação entre "itens de informação" (palavras) contidos em documentos. Nesse caso, um conjunto de

documentos será considerado como um "banco de dados de transações", onde cada documento será considerado uma "transação".

Para que as regras descobertas sejam mais interessantes para o usuário, os "itens de informação" ocorrendo nas regras são limitados a palavras de determinado tipo, em particular substantivos. Além disso, o sistema utiliza a medida de *Tf-Idf* (*Term Frequency – Inverse Document Frequency*), da área de recuperação de informações, para estimar quais palavras são mais interessantes para o usuário.

Este trabalho está dividido da seguinte maneira: Capítulo 2 apresenta a revisão da literatura dos principais conceitos envolvidos na descoberta de regras de associação, bem como em *text mining*. O Capítulo 3 descreve o método proposto para descoberta de regras de associação em textos. O Capítulo 4 apresenta resultados computacionais do método proposto. O Capítulo 5 discute trabalhos relacionados. Finalmente, o Capítulo 6 conclui o trabalho.

Capítulo 2

Revisão da Literatura

2.1. Descoberta de Regras de Associação

Segundo [Freitas & Lavington 98], na sua forma original, esta tarefa é definida por um tipo especial de dados, chamados dados de *basket* (cesta de supermercado), onde uma tupla consiste em um conjunto de atributos binários chamados itens. Cada tupla corresponde a uma transação do usuário, aonde um item tem valor verdadeiro ou falso dependendo ou não do item correspondente ter sido adquirido pelo usuário naquela transação.

Esse tipo de dado é normalmente coletado através da tecnologia de código de barra, cujo exemplo típico é um scanner de supermercado.

Segundo [Agrawal et al 93], muitas organizações tem coletadas grandes quantidades do tipo de dado de *basket*. Esses conjuntos de dados são normalmente armazenados em memória terciária e são vagarosamente migrados para os sistemas de base de dados. Uma das principais razões para o limitado sucesso dos sistemas de base de dados nesta área é que tais sistemas não fornecem a funcionalidade necessária para um usuário interessado beneficiar-se deste tipo de informação.

Um exemplo de regra de associação seria a seguinte afirmação: ' 90% das transações que adquirem pão e manteiga também adquirem leite'.

Essa regra pode ser representada na forma: **se** pão, manteiga **então** leite.

O antecedente desta regra consiste em 'pão' e 'manteiga' e o conseqüente consiste somente em 'leite'.

O número 90% é o fator de confiança da regra.

2.1.1. Conceitos básicos

Uma regra de associação é um relacionamento na forma X => Y, onde X e Y são conjuntos de itens e $X \cap Y = \varnothing$. Para cada regra de associação é computado um fator de suporte 'Sup' e um fator de confiança 'Conf'.

Sup é definido como a razão do número de tuplas que satisfaçam tanto X quanto Y sobre o número total de tuplas, isto é, $Sup = |X \cup Y| / N$, onde N corresponde ao número total de tuplas.

Conf é definido como a razão do número de tuplas que satisfaçam tanto X quanto Y sobre o número de tuplas que satisfazem X, isto é, $Conf = |X \cup Y| / |X|$.

A tarefa de descoberta de regras de associação consiste na extração de todas as regras com Sup e Conf maior que ou igual ao Sup e Conf mínimos (denotados SupMín e ConfMín) especificados por um usuário.

A descoberta de regras de associação é geralmente executada em duas etapas.

Na primeira etapa, um algoritmo determina todos os conjuntos de itens que apresentam Sup maior que ou igual ao SupMín especificado pelo usuário. Tais conjuntos são chamados de conjuntos de itens freqüentes (*frequent itemsets*), às vezes chamados conjuntos de itens grandes (*large itemsets*).

Na segunda etapa, para cada conjunto de itens freqüente, todas as possíveis regras candidatas são geradas e testadas quanto à ConfMín. Uma regra candidata é gerada extraindo-se um subconjunto de itens do conjunto de itens freqüentes para ser o antecedente da regra e usando-se os itens restantes no conjunto de itens freqüentes para ser o conseqüente da regra. Somente as regras candidatas com Conf maior que ou igual à ConfMín especificada pelo usuário são incluídas na saída do algoritmo.

Formalmente, dado um conjunto de itens freqüentes $Z = I_1, I_2.....I_k$, k >=2, o algoritmo da segunda etapa gera todas as regras que usem os itens do conjunto $I_1, I_2,....., I_k$. O antecedente de cada uma dessas regras será um subconjunto X de Z tal que X possui pelo menos 1 item, e o conseqüente Y será o conjunto de itens Z - X. Por exemplo, considere a geração de uma regra com um único item no conseqüente. Para gerar uma regra $X \Rightarrow I_j$ com fator de confiança c, onde $X = I_1 I_2.....I_{j-1}, I_{j+1}.... I_k$, e $Y = I_j$, obtenha o suporte de $X \cup Y$ e divida-o pelo suporte de X. O resultado dessa divisão é c, o grau de confiança da regra. Se c é

maior que ou igual a ConfMín, então a regra é incluída na saída do algoritmo; caso contrário a regra é descartada.

Uma solução simples para a primeira etapa (identificação dos conjuntos de itens freqüentes) é formar todos os conjuntos de itens e obter seus respectivos suportes em uma única passagem (ou "varredura"- scan) pelos dados. No entanto, essa solução não é razoável em um ambiente computacional. Se existirem 'm' itens na base de dados, haverão 2^m conjuntos de itens possíveis, aonde 'm' pode ser facilmente maior que 1.000.

Um método mais eficiente proposto por [Agrawal et al 93] é relatado sucintamente a seguir:

- Usa-se um procedimento de estimação cuidadosamente regulado para determinar quais conjuntos de itens podem ser medidos (ter seu suporte contado) em uma passagem pelos dados. Este procedimento atinge um meio termo entre o número de passagens pelos dados e o número de conjuntos de itens que são medidos em uma passagem;
- Incorpora-se o gerenciamento de *buffer* para lidar com o fato de que todos os conjuntos de itens que precisam ser medidos na passagem podem não se caber na memória principal, mesmo depois de serem podados.

[Agrawal et al 93] testaram a eficácia do método descrito acima aplicando-o em dados de venda obtidos de uma grande companhia. Para este conjunto de dados, o método exibiu excelente performance. O procedimento de estimação exibiu alta precisão e as técnicas de poda foram capazes de eliminar uma fração muito grande de conjuntos de itens sem ser necessário medir o suporte daqueles conjuntos de itens.

2.1.2. Regras de Associação Hierárquicas

Em muitos casos taxonomias de itens estão disponíveis, ou seja, itens hierarquicamente superiores aos itens abordados são conhecidos. Na Figura 2.1 é demonstrado um exemplo de uma taxonomia. Esta taxonomia informa que *Jacket* é um tipo de *Outerwear*, *Ski Pants* é um tipo de *Outerwear, Outerwear* é um tipo de *Clothes*, etc.

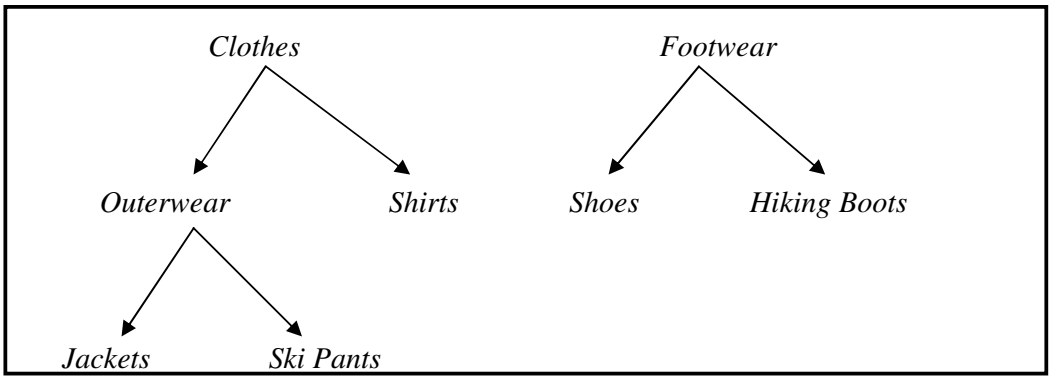

Figura 2.1: Exemplo de uma Taxonomia [Srikant & Agrawal 95]

Segundo [Srikant & Agrawal 95] as regras com itens em um nível mais profundo de taxonomia tendem a não possuir suporte mínimo, e regras com itens em um nível mais elevado de taxonomia tendem a não possuir confiança mínima. Assim, quando lidando com itens hierárquicos, um algoritmo deve considerar a possibilidade de descobrir regras com itens em diferentes níveis da hierarquia, a fim de atingir o suporte e confiança mínimos especificados pelo usuário. Portanto, num certo sentido, as taxonomias podem ser usadas para podarem regras sem interesse ou redundantes.

Cabe ressaltar que, quando calculado o suporte de regras de associação hierárquica, o suporte de um item não é igual a soma dos suportes de seus itens filhos na taxonomia, visto que diversos itens filhos poderiam estar presentes em uma única transação.

2.2. Conceitos Básicos de Processamento de Linguagem Natural

2.2.1. Classes Gramaticais (*Parts of Speech*)

As palavras que possuem comportamento sintático e semântico similares e típicos podem ser agrupadas em uma mesma classe, originando as chamadas categorias sintáticas ou gramaticais, mais comumente denominadas *parts of speech* (*POS*). As três principais são substantivo, verbo e adjetivo. Os substantivos referem-se a pessoas, animais, conceitos e coisas. O verbo é usado para expressar a ação numa sentença e os adjetivos, por sua vez, propriedades dos substantivos.

Cabe ressaltar que existem softwares de domínio público que identificam automaticamente a *POS* de cada palavra em uma sentença [Manning & Schutze 99]. Esse tipo de software foi utilizado na realização deste trabalho, como será visto posteriormente.

Tendo em vista que neste trabalho os itens das regras de associação são substantivos, essa classe gramatical é descrita em mais detalhes a seguir.

Os substantivos (*nouns*) em geral referem-se a entidades genéricas, por exemplo: *dog, tree, person, hat, speech, idea, philosophy*. Porém, substantivos também podem se referir a nomes próprios (*proper nouns*), tais como nomes de pessoas, cidades, etc.

Analisando as inflexões comuns do substantivo (*number, gender* e *case*), reconhece-se para o Inglês uma única inflexão: o plural dos substantivos. Na sua grande maioria, o plural do substantivo é caracterizado pelo acréscimo do sufixo *-s* (*dog: dogs, hat: hats*), porém há várias exceções. Por exemplo, o plural pode ser formado por "*es*", como em *speech: speeches* e, ainda, destacam-se os termos irregulares (*woman: women, child: children, etc.*).

Destaca-se, porém, o fato de que em algumas linguagens o substantivo pode exercer diferentes funções (sujeito, objeto, etc.) em uma sentença, inflexão esta chamada de *case*. Por exemplo, em Latim 'filho' é *filius* quando substantivo e *filium* quando objeto de um verbo. Em Inglês não há inflexão real do tipo *case*, porém identifica-se como sendo uma instância de *case* o que sistematicamente é reconhecido por *genitive*, ou seja, o possessivo. Por exemplo: *woman's house* indica que *woman* possui *house*. O *genitive* é indicado por *'* ou *'s* chamado de *clitic*.

No contexto de *Text Mining* pode ser desejável tratar todas as inflexões de um substantivo como um único termo. Isso pode ser realizado com um algoritmo de *stemming*, conforme explicado brevemente na seção 2.2.6.2.

2.2.2. Tokenização

Tokenização consiste em identificar *tokens*, ou palavras, em um texto. Uma regra prática para identificar palavras, baseada em noções puramente gráficas, sugere que estas são definidas como "uma string com caracteres alfanuméricos contíguos sem espaços, podendo também incluir hífens e apóstrofos, mas nenhuma outra marca de pontuação".

Dentro do espírito dessa noção de palavra gráfica, o "espaço em branco", mais precisamente um espaço ou *tab* ou início de uma nova linha, é o recurso mais utilizado em inglês para separar palavras, apesar desse sinal não ser necessariamente seguro.

Alguns problemas específicos são discutidos a seguir [Manning & Schutze 99]:

a) Ambigüidade do Ponto Final ("."): as palavras podem ter anexada a elas uma pontuação representada por vírgula, ponto-e-vírgula e ponto final. À primeira vista, apresenta-se fácil o reconhecimento da pontuação. Porém o caso de ponto final é problemático. Apesar de que muitas vezes um ponto marca o fim de uma sentença, às vezes um ponto faz parte de abreviaturas tais como *etc*. ou *Calif*. Presume-se que esses pontos das abreviaturas fazem parte da palavra com a qual se apresenta. Por exemplo: distingue-se *Wash*. (que é a abreviatura do estado de Washington) da forma capitalizada do verbo *wash*. Percebe-se também que quando uma abreviatura como *etc*. aparece no fim de uma sentença, apresenta-se apenas um ponto com ambas as funções de abreviatura e final de frase ao mesmo tempo.

b) Ocorrência de apóstrofos ('): Nesse caso é difícil saber como resgatar informações do tipo *I'll* ou *isn't*. Conforme definição anterior, apresenta-se como uma "palavra gráfica", mas há uma forte intuição que realmente tem-se duas palavras como as contrações de *I will* e *is not*. Assim, alguns sistemas separam tais contrações em duas palavras enquanto outros não o fazem.

c) Hifenização: Muitas expressões escritas de forma hifenizada são claramente tratadas como uma palavra única, como por exemplo *e-mail*, *co-operate* ou *A-1-plus*. Existem casos também como *non-lawyer*, *pro-Arab* e *so-called* aonde os hífens são léxicos. Eles são comumente inseridos antes ou após pequenas palavras de caráter formativo, algumas vezes com o propósito de separar seqüências de vogais.

d) Espaço em branco: Algumas vezes um espaço em branco não indica uma quebra na palavra. Por exemplo, caso seja decidido tratar *database* como sendo uma palavra, uma outra maneira de tratá-la como sendo uma palavra é escrevendo-a na forma *data base*. Destacam-se os números de telefone como sendo casos mais comuns dessa condição, onde um número como *9365 1873* é considerado uma única palavra. Há também vários nomes constituídos de múltiplas partes, como *New York* e *San Francisco,* que são reconhecidos como uma única palavra. Especialmente torna-

se difícil o caso em que este problema interage com hifenização, como demonstrado na expressão a seguir: "*the New York-New Haven railroad*". Neste caso, o hífen não expressa agrupamento com as "palavras gráficas" imediatamente adjacentes - o tratamento de *York-New* como unidade semântica seria um enorme equívoco.

2.2.3. Morfologia, Sintaxe, Semântica e Pragmática

Morfologia é o estudo da formação dos vocábulos. É uma parte da Gramática que estuda as palavras quanto à estrutura e formação, quanto às flexões e quanto à classificação.

Sintaxe é o estudo das regularidades e equívocos na ordenação das palavras e estruturação das frases. Nas *noun phrases*, o substantivo (*noun*) é o elemento central que determina o caráter sintático da frase.

Semântica é essencialmente o estudo do significado das palavras. A semântica pode ser dividida em duas partes, o estudo do significado individual das palavras (ou semântica léxica) e o estudo de como o significado individual das palavras podem combinar com o significado das sentenças.

Uma forma de aperfeiçoar a compreensão de semântica léxica é estudar como os significados de uma palavra estão relacionados com cada uma das outras; organizando, por exemplo, as palavras em uma hierarquia léxica. Aqui destaca-se o sistema *WordNet* [Fellbaum 98] que será discutido posteriormente.

Pragmática é o estudo de como conhecimento sobre o mundo e convenções de linguagem interagem com o significado literal de sentenças.

2.2.4. Análise de Discurso

A análise de discurso consiste em elucidar relacionamentos entre sentenças em um texto. O problema central na análise de discurso é a resolução de relações anafóricas. Essas relações ocorrem quando diferentes termos ou '*noun phrases*' se referem à mesma pessoa ou coisa. Analisando um exemplo [Manning & Schutze 99]: "*Mary helped Peter get out of the cab. He thanked her* ", Nesse exemplo, "*Peter"* e "*He*" referem-se à mesma pessoa. A solução desse tipo de relação é muito importante para a tarefa de extração de informação, a qual será discutida na seção 2.3.

Para que a execução dessa tarefa tenha êxito, a correta identificação das relações anafóricas é crucial para manter o rastro dos participantes. Por exemplo, considere as duas sentenças a seguir: *"Hurricane Hugo destroyed 20,000 Florida homes. At an estimated cost of one billion dollars, the disaster has been the most costly in the state's history."* Se for identificado que *Hurricane Hugo* e *disaster* referem-se à mesma entidade nessas duas sentenças (um mini-discurso), poderemos fornecer a resposta *Hugo* para a pergunta: "*Which hurricane caused more than a billion dollars worth of damage?*".

Análise de discurso é parte da Pragmática.

Relações anafóricas são um fenômeno pragmático de difícil resolução por envolverem conhecimento sobre o mundo real. Por exemplo, para resolver as relações do mini-discurso anterior, é necessário possuir o conhecimento de que *hurricanes* são *disasters*.

Muitos problemas de pragmática não recebem a devida atenção da área de processamento de linguagem natural estatístico porque é muito complexa a modelagem de conhecimento do mundo real nos moldes estatísticos [Manning & Schutze 99].

2.2.5. Colocações (ou expressões compostas)

Colocações (expressões compostas) são agrupamentos de palavras onde o significado do todo é a soma dos significados das partes mais algum componente semântico adicional não previsto pelas partes. Como exemplo pode-se citar: cabelo branco, pele branca e vinho branco aonde o branco do cabelo é cinza, o branco da pele é rosado e o branco do vinho é amarelo.

Um caso extremo de colocação são as frases idiomáticas, onde o relacionamento entre o significado das palavras e o significado da frase é completamente adaptado, opaco ou "simplesmente aceito". Como exemplo pode-se citar: *'carriage return'* indicando o caracter delimitador de final de linha, porém, não condizendo com o significado original da palavra.

2.2.6. Pré-processamento de Texto para *Text Mining*

2.2.6.1. Remoção de *Stop Words*

Stop Words são palavras que ocorrem freqüentemente em textos. Uma vez que elas são muito comuns, sua presença não contribui significativamente para a determinação do

conteúdo do documento. Logo, elas podem ser removidas do documento, para fins de *Text Mining*. Exemplos são: "*do*", "*can*", "*will*", etc.

Neste trabalho utilizou-se de uma lista de *Stop Words* [Stop word list] com 641 elementos. O algoritmo varre todo o texto em questão à procura das palavras contidas nesta lista, removendo-as ao encontrá-las, eliminando a possibilidade de tais palavras serem submetidas aos demais procedimentos descritos nesta seção 2.2.6.

2.2.6.2. Algoritmo de *Stemming*

Stemming consiste em converter cada palavra para seu "radical" ("*stem*"), isto é, uma forma neutra com respeito a *tag-of-speech* e inflexões verbais plurais. Por exemplo, as palavras "*learning*" e "*learned*" são ambas convertidas para o *stem* "*learn*" [Porter 97].

O algoritmo de *stemming* utilizado neste trabalho é o algoritmo de Porter [Porter 86], o qual varre uma string numa série de passos descritos a seguir:

- Primeiro passo: remove o plural, incluindo casos especiais tais como *"sses" "ies"*;
- Segundo passo: une padrões com alguns sufixos tais como:
  "ational" -> "ate", "tional" -> "tion", "enci" -> "ence", "anci" -> "ance", "iser" -> "ize", "abli" -> "able", "alli" -> "al", "entli" -> "ent", "eli" -> "e", "ousli" -> "ous", "ization" -> "ize", "isation" -> "ize", "ation" -> "ate", "ator" -> "ate", "alism" -> "al", "iveness" -> "ive", "fullness" -> "ful", "ousness" -> "ous", "aliti" -> "al", "iviti" -> "ive", "biliti" -> "ble"
  *Nessas transformações, removem-se os sufixos e os substituem por suas "*raízes*".*
- Terceiro passo: ocorre a manipulação das transformações necessárias para algumas palavras especiais conforme apresentadas a seguir:
  "icate" -> "ic", "ative" -> "", "alize" -> "al", "alise" -> "al", "iciti" -> "ic", "ical" -> "ic", "ful" -> "", "ness" -> ""
- Quarto passo: a palavra analisada é checada perante mais sufixos, no caso da palavra ser composta incluindo:
  "al", "ance", "ence", "er", "ic", "able", "ible", "ant", "ement", "ment", "ent", "sion", "tion", "ou", "ism", "ate", "iti", "ous", "ive", "ize", "ise"

- Quinto e último passo: checa se a palavra termina em vogal, fixando-a apropriadamente. Este algoritmo particular serve também para analisar e separar afixos e alguns prefixos simples, tais como:*"kilo", "micro", "milli", "intra", "ultra", "mega", "nano", "pico", and "pseudo"*.

2.2.6.3. *Part of Speech Tagger (POS)*

Conforme já descrito previamente na seção 2.2.1, visa a obtenção dos *nouns* (substantivos) do arquivo texto que está sendo processado. O software *Brill POS Tagger* [Brill POS] é um software de domínio público e está sendo útil para a realização do trabalho. O exemplo a seguir ilustra a entrada e saída do software:

Input: Mr. Red have a red ball

Output: Mr/NNP ./. Red/NNP have/VBP a/DT red/JJ ball/NN

Utilizando-se este software, realiza-se a categorização sintática e semântica das palavras, obtendo-se os *nouns* que representam a classe gramatical usada nesse trabalho. A classe gramatical dos *nouns* é representada pelas seguintes variantes que devem ser consideradas para as etapas posteriores:

- NN: *Noun, singular or mass*
- NNS: *Noun, plural*
- NNP: *Proper noun, singular*
- NNPS: *Proper noun, plural*

2.3. Extração de Informações

2.3.1. Definição Geral de Extração de Informação

Na extração de informação, varremos o texto buscando tipos específicos de eventos tais como desastres naturais, ataques terroristas ou aquisições corporativas. A tarefa é identificar os participantes do evento e outra informação específica que caracterize este evento (por exemplo, o preço promocional de uma fusão corporativa ou o tipo de arma usada em um ataque terrorista).

2.3.2. Fases de um sistema de Extração de Informação

Apesar de existirem muitas variações de sistema para sistema, segue abaixo as principais funções executadas em um sistema de extração de informação [Cardie 97].

a) Cada texto de entrada é primeiramente dividido em sentenças e palavras, numa etapa de tokenização e *tagging* (colocação de etiquetas). Muitos sistemas etiquetam cada palavra com a respectiva classe gramatical - ou *part of speech (POS)* e, possivelmente, classes semânticas neste ponto do processo;

b) A fase de análise de sentença compreende um ou mais estágios de análises sintáticas ou *parsing* que, juntas, identificam grupos de substantivos, grupos de verbos, expressões preposicionais e outras estruturas simples. Em alguns sistemas, o *parser* também localiza, num nível superficial, sujeitos e objetos diretos e identifica conjunções e outras *expressões* complexas. Em algum ponto, antes, durante ou depois dos passos principais da análise sintática, o sistema de extração de informação também procura e etiqueta entidades semânticas relevantes ao tópico de extração;

c) A fase de extração é o primeiro componente do sistema que é específico para o domínio de aplicação. Durante a fase de extração, o sistema identifica relações entre entidades relevantes no texto;

d) O trabalho principal na fase de *merging* é a resolução co-referenciada ou resolução anafórica: O sistema examina cada entidade encontrada no texto e determina se tal entidade se refere a uma entidade já existente ou se ela é nova e deve ser adicionada ao nível de discurso do sistema que representa o texto;

e) As inferências a nível de discurso feitas durante a fase anterior, isto é, de *merging*, auxiliam a fase de geração de gabaritos, a qual determina o número de eventos distintos no texto, mapeia os itens individuais de informação extraídos de cada evento e produz gabaritos de saída. Inferências sobre o domínio de aplicação específico podem também ocorrer durante a geração de gabaritos.

Para essa última fase pode-se usar, por exemplo, *gabaritos* de expressões com classes gramaticais (*part of speech*) pré-definidas, tais como <nome nome>, <nome preposição nome>, <adjetivo nome>, etc, veja por exemplo [Feldman et al 98].

2.4. Identificação e categorização semântica de nomes próprios

2.4.1. Visão Geral

Para analisar uma instância de um nome próprio deve-se [McDonald 96]:

1. Delimitar a seqüência de palavras que compõem o nome, isto é, identificar seus limites. No sistema *Sparser*, por exemplo, o algoritmo de delimitação simplesmente agrupa qualquer seqüência contígua de palavras com iniciais maiúsculas. "*&*" é considerado como estendendo a seqüência, e pontos (".") são considerados como sinalizadores de final, a menos que eles sejam parte de uma abreviação;
2. Classificar o resultado baseado em cada tipo individual dos nomes, por exemplo, nomes de cidades, países, organizações, nomes de pessoas, etc.
3. Registrar o nome e a individualidade denotada por ele no modelo de discurso como nossa interpretação do significado obtido.

Nomes próprios exibem uma enorme diversidade, porém apresentam sempre uma estrutura sistemática e composta reconhecida pela gramática. Isso abre a possibilidade para identificação de nomes próprios usando evidências internas ou externas, conforme explicado a seguir.

2.4.2. Diferença entre Evidência Interna e Evidência Externa para identificação de nomes próprios

Evidência interna provém de uma seqüência de palavras que compõem o nome [McDonald 96]. Tem-se aqui um critério definitivo quando da presença de "termos corporativos" conhecidos para indicar companhias. Como exemplo citam-se "*Ltd.*," e "*G.m.b.H*." Também existe o critério heurístico, como em abreviaturas ou primeiros nomes conhecidos, os quais geralmente indicam pessoas (por exemplo, "*Mr*.").

Por outro lado, a evidência externa provém do contexto no qual o nome está inserido. A base para esta evidência é a observação óbvia que nomes são justamente a maneira pela

qual referenciam-se indivíduos de tipos específicos e que tais tipos têm propriedades características e participam de eventos também característicos. Por exemplo, "*Osamu Nagayama, deputy president of Chugai...*". Nessa sentença, a palavra "*president*" é uma evidência externa de que "*Osamu Nagayama*" é um nome de pessoa.

Uma motivação para o uso de evidência externa é que as listas predefinidas de palavra comumente usada como evidência interna não costumam ser completas. E, em muitas situações, a evidência externa supera a evidência interna.

2.5. *WordNet*

WordNet é um banco de dados léxico eletrônico [Miller 98]. Ele contém informações sobre palavras, expressões compostas (*phrasal verbs*, colocações, frases idiomáticas, etc.). Ele separa suas entradas de acordo com categorias sintáticas: substantivo, verbo, adjetivo e advérbio. Dentro de cada categoria, várias relações semânticas entre palavras e expressões compostas são armazenadas. Como exemplo, descreve-se a seguir, sucintamente, as relações de hiperônimo e holônimo.

Um hiperônimo é uma palavra mais genérica, por exemplo, animal é um hiperônimo de gato. Um hipônimo é uma palavra com significado mais específico: gato é um hipônimo de animal. (Se w^1 é um hiperônimo de w^2, então w^2 é um hipônimo de w^1).

A condição de ser parte do todo é chamada merônimo. Por exemplo, folha é um merônimo de árvore. Por sua vez, o todo correspondente a uma parte é chamado holônimo.

Capítulo 3

Método Proposto

Neste trabalho um algoritmo de regras de associação será utilizado para descobrir associações entre itens que freqüentemente ocorrem em um mesmo documento. Conforme mencionado na Introdução, cada documento será considerado uma transação.

Cabe ressaltar que os arquivos utilizados nesse trabalho provêm da base de dados chamada *Tipster* [Tipster 94].

3.1. Pré-processamento de Texto para *Text Mining*

Inicialmente o usuário escolhe um subconjunto de documentos da base *Tipster*, a partir dos quais deseja-se extrair regras de associação. Cada documento está armazenado em um arquivo separado, e cada documento corresponde a uma transação da base de dados fornecida como entrada para o algoritmo de associação.

Para cada arquivo (documento) o sistema primeiro extrai todas as palavras que ocorrem naquele documento. A seguir, o conjunto de palavras de cada documento é submetido a um pré-processamento consistindo de três passos, a saber:

a) remoção de "*stop words*";

b) execução de um algoritmo de *stemming*, para extrair o radical de cada palavra (removendo sufixos, terminações de gênero e número, etc.);

c) execução de um algoritmo "*Part of Speech Tagger*" para identificação da classe gramatical de cada palavra (substantivo, verbo, etc.).

Esses três passos foram discutidos na subseção 2.2.6.

Após realizada a seqüência de procedimentos descrita acima para todos os arquivos que compõem o banco de dados, será obtido o primeiro arquivo resultante da aplicação do sistema, chamado pesquisa.txt, onde cada registro consiste de dois campos:

a) nome do arquivo componente da base de dados *Tipster*;
b) palavra extraída do documento.

Note que o nome do arquivo é repetido quantas vezes forem necessárias para acompanhar cada palavra válida obtida neste arquivo, sendo ambos separados pela vírgula como delimitador. Um exemplo é mostrado na Tabela 3.1.

Tabela 3.1: Exemplo de arquivo pesquisa.txt

WSJ8_008.008,spokeswoman
WSJ8_008.008,compani
WSJ8_008.008,comment
WSJ8_008.008,specul
WSJ8_008.008,rumor
WSJ8_008.010,southmark
WSJ8_008.010,unit
WSJ8_008.010,bui
WSJ8_008.010,chunk
WSJ8_008.010,parent
WSJ8_008.012,nabisco
WSJ8_008.012,partner
WSJ8_008.012,author
WSJ8_008.012,staff
WSJ8_008.012,group
WSJ8_008.012,nabisco
WSJ8_008.014,amp
WSJ8_008.014,stratton

Tabela 3.1: Exemplo de arquivo pesquisa.txt (continuação)

WSJ8_008.014,senn
WSJ8_008.016,questech
WSJ8_008.016,presid
WSJ8_008.016,appoint
WSJ8_008.016,salvatori

3.2. Obtenção das regras de associação

3.2.1. Ordenação dos termos que comporão a base de dados

A base de dados que servirá como base para a obtenção das regras de associação será gerada a partir dos dados gravados no arquivo pesquisa.txt. Porém, a base de dados terá seus termos ordenados de forma decrescente de valor do *Tf-Idf* (*term frequency – inverse document frequency*). Em outras palavras, os termos que se encontrarem no topo da lista após a ordenação são os termos que possuem maior peso, determinado através do cálculo do *Tf-Idf*. Esses são considerados os melhores termos para serem usados na obtenção das regras de associação.

Note que, em geral, a medida de *Tf-Idf* é usada para computar o peso (ou relevância) de um termo "t" em um determinado documento "d". Neste trabalho, deseja-se computar um único peso para cada termo "t", levando em conta todos os valores de *Tf-Idf*(t, d) em todos os documentos "d". Assim, o peso atribuído a um termo "t" corresponderá ao somatório de todos os *Tf-Idf*´s calculados para esse termo, em todos os documentos "d" em que "t" estiver presente. Portanto, utilizando-se da fórmula para obtenção do *Tf-Idf*, tem-se que:

$$\boxed{\textit{Tf-Idf}(t, d) = \textit{Tf}(t, d) * \textit{Idf}(t)}$$

$$\boxed{\textit{Tf-Idf}(t, d) = \textit{Tf}(t, d) * \log (N / n)}$$

onde:

Tf-Idf(t, d) é o *Tf-Idf* de um termo "t" em um documento "d".

A freqüência de um termo "t" em um documento"d" é dada por:

Tf(t, d) = número de vezes que o termo"t" ocorre em um documento "d".

No cálculo do *Idf*(t), "N" é o número total de documentos e "n" é o número de documentos onde o termo "t" ocorre pelo menos uma vez.

Sumarizando, o sistema computa uma única medida de *Tf-Idf* para cada termo "t", e quanto maior o valor dessa medida, maior é a relevância (ou interesse) estimada para aquele termo. Assim, apenas os termos com maiores valores de *Tf-Idf* serão usados como entrada para o algoritmo de associação. Isso permite uma considerável redução no número de termos a serem fornecidos como "itens de informação" para o algoritmo de associação, o que tende a reduzir bastante o tempo de processamento daquele algoritmo.

3.2.2. Preparação da base de dados conforme o padrão de arquivo de entrada para o *Weka*

Nesse trabalho utilizou-se um software de domínio público chamado *Weka* (*Waikato Environment for Knowledge Analysis*) que contém implementações de vários métodos de mineração de dados, para realização de diversas tarefas, incluindo a tarefa de associação, que é o foco deste trabalho. Os módulos de tarefas disponíveis no *Weka* e que serão úteis para aplicação do sistema proposto neste trabalho são os de *Preprocess* e *Associate*, este último utilizando o algoritmo *Apriori* para a descoberta de regras de associação [Witten & Frank 99].

A base de dados a ser submetida ao algoritmo *Apriori* do *Weka* para obtenção de regras de associação tem como base o arquivo pesquisa.txt após ordenação, conforme descrito no item 3.2.1, além do que há que submetê-la a uma padronização exigida pelo software *Weka*. Assim:

Weka aceita somente arquivos em formato *arff*, logo há que se converter para o formato exigido.

Um arquivo no formato *arff* consiste de duas partes. A primeira parte contém uma lista de todos os atributos (um atributo por linha), com os valores dos atributos entre "{" e "}" e separados por vírgulas. A segunda parte consiste das instâncias ou registros a serem minerados (um registro por linha). Cada instância contém uma lista de itens, sendo que cada item corresponde a uma palavra extraída do documento correspondente, no caso deste trabalho. Os itens (atributos) são também separados por vírgulas. Cabe ressaltar que o

software *Weka* requer que a ausência de um item em um registro seja denotada pelo símbolo "?". A presença de um item em um registro pode ser denotada por qualquer valor, sendo que o valor "1" foi adotado neste trabalho.

Após concluída a etapa anterior, basta carregar o arquivo já alterado num editor de texto e adicionar os seguintes *flags*:

- @relation <nome da base de dados em uso>: Esse *flag* deve ser incluído na primeira linha do arquivo *.arff*.
- @attribute <informação sobre o atributo>: Esse *flag* deve ser incluído para cada atributo (item) na primeira parte do arquivo *.arff*.
- @data <sem nenhum parâmetro na mesma linha, somente indicando o início da listagem dos registros>

Somente para ilustração, na Tabela 3.2 é apresentada uma amostragem de uma base de dados computada a partir de uma base de documentos real. Nesta tabela o arquivo contém 50 atributos (itens). Note que cada atributo possui apenas um valor (indicado entre "{" e "}"), a saber o valor "1" como mencionado anteriormente. Isso é resultado do fato de trabalharmos com itens binários. Assim, em uma dada transação, um dado item ou está presente (valor "1") ou não está presente, o que é indicado pelo *flag* "?" (*valor ausente*), como pode ser observado nas linhas da Tabela 3.2 após o *flag* "@data". Na Tabela 3.2 são mostradas apenas 10 linhas após o *flag* "@data" por simplicidade, mas naturalmente a base de dados é bem maior.

Tabela 3.2: Exemplo de arquivo associacao.arff

@relation associacao
@attribute american { 1 }
@attribute amp { 1 }
@attribute analyst { 1 }
@attribute bank { 1 }
@attribute bid { 1 }
@attribute bond { 1 }
@attribute bush { 1 }
@attribute busi { 1 }

Tabela 3.2: Exemplo de arquivo associacao.arff (continuação)

@attribute cent { 1 }
@attribute compani { 1 }
@attribute comput { 1 }
@attribute corp { 1 }
@attribute court { 1 }
@attribute dai { 1 }
@attribute democrat { 1 }
@attribute dollar { 1 }
@attribute firm { 1 }
@attribute fund { 1 }
@attribute govern { 1 }
@attribute group { 1 }
@attribute industri { 1 }
@attribute investor { 1 }
@attribute issu { 1 }
@attribute japan { 1 }
@attribute manag { 1 }
@attribute market { 1 }
@attribute month { 1 }
@attribute peopl { 1 }
@attribute plan { 1 }
@attribute point { 1 }
@attribute presid { 1 }
@attribute price { 1 }
@attribute product { 1 }
@attribute quarter { 1 }
@attribute rate { 1 }
@attribute rjr { 1 }
@attribute sale { 1 }
@attribute secur { 1 }

Tabela 3.2: Exemplo de arquivo associacao.arff (continuação)

@attribute share { 1 }
@attribute soviet { 1 }
@attribute state { 1 }
@attribute stock { 1 }
@attribute system { 1 }
@attribute takeov { 1 }
@attribute tax { 1 }
@attribute time { 1 }
@attribute trade { 1 }
@attribute week { 1 }
@attribute year { 1 }
@attribute york { 1 }
@data
1,?,?,?,?,?,1,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,1,?,?,?,?,?,?,?,?,1,?,?,?,?,?,?,?,1,?,?,?,?,?,?,1,?,1,?
?,?,?,?,?,?,?,1,?,?,?,?,?,?,?,?,?,?,1,1,?,?,?,?,1,?,?,?,1,?,?,1,?,?,?,?,1,?,?,?,?,?,?,?,?,?,?,?,?,1,?
?,1,?,?,?,?,?,?,?,?,1,?,1,?
?,?,?,?,?,?,?,?,?,1,?,1,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,1,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,1,?
?,?,?,?,?,?,?,?,?,1,?,1,?,?,?,?,?,?,?,1,?,?,?,?,?,?,?,?,?,?,1,?,1,?,?,?,?,?,?,?,?,?,?,?,?,?,?,1,?,?
?,?,?,?,?,?,?,?,?,?,?,1,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,1,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,1,?
?,?,?,1,1,1,?,?,?,1,?,?,?,?,?,?,?,?,?,?,?,?,1,?,?,?,1,?,?,?,?,?,?,?,?,?,?,1,?,1,?,?,1,?,?,?,?,1,?,1,?
?,?,?,?,?,?,?,1,1,1,1,1,?,?,?,?,?,?,?,?,1,?,1,?,?,?,?,?,?,?,?,?,?,1,?,?,1,?,1,?,?,?,1,?,1,?,1,?,1,?
?,?,?,?,?,?,?,?,1,1,?,1,?,?,?,?,?,1,?,?,?,?,?,?,1,?,?,?,?,?,?,?,?,1,?,?,1,?,1,?,?,?,?,?,1,?,?,?,1,?
?,?,1,1,1,?,?,1,1,1,?,?,?,?,?,?,?,?,?,1,?,?,?,?,1,1,?,?,1,?,?,?,?,?,?,?,?,?,?,1,?,?,1,?,?,?,?,?,1,?,1,?

3.2.3. Execução do Algoritmo *Apriori* do *Weka*

Uma vez executados os passos de pré-processamento descritos nos itens 3.2.1 e 3.2.2, os dados são submetidos ao algoritmo *Apriori* do *Weka*. O sistema permite trabalhar o *Apriori* com parâmetros *defaults* ou com parâmetros definidos pelo usuário.

As opções para alteração dos parâmetros por parte do usuário são listadas na Tabela 3.3.

Note que, embora a especificação da confiança mínima das regras descobertas seja feita de forma direta, através da opção "-C", a especificação do suporte mínimo das regras descobertas é feita de forma indireta, através dos parâmetros "-D", "-U" e "-M". O sistema inicialmente tenta extrair regras de associação com suporte igual ao especificado pela opção "-U". Se o número de regras descobertas com aquele suporte for maior ou igual ao número de regras especificado na opção "-N", o algoritmo termina, e reporta as N primeiras regras descobertas ao usuário. Caso contrário o sistema decrementa o valor do suporte pelo valor especificado na opção "-D", e tenta descobrir mais regras, agora com um menor valor de suporte. Esse processo é iterativamente repetido, até que o valor do suporte atinge o limite mínimo especificado na opção "-M" ou até que o número máximo N de regras tenha sido descoberto.

Tabela 3.3: Opções do *Apriori*

Opção	Função
-T	especificar nome do arquivo de treinamento
-N	especificar o número máximo de regras a serem descobertas pelo algoritmo *Apriori* do *Weka*
-C	especificar confiança mínima das regras descobertas
-D	especificar a variação para decréscimo do suporte mínimo, do limite superior até o limite inferior
-U	especificar limite superior para suporte mínimo das regras descobertas
-M	especificar limite inferior para suporte mínimo das regras descobertas

A título de exemplo, a Tabela 3.4 mostra os parâmetros usados para processar uma base de dados com 50 atributos (palavras) e com 2434 registros (documentos). Uma amostragem dessa base de dados foi apresentada na Tabela 3.2. O resultado do processamento dessa base de dados, com os parâmetros mostrados na Tabela 3.4, é mostrado na Tabela 3.5. Note que, apesar do parâmetro –N especificar que desejávamos 50 regras descobertas, o algoritmo *Apriori* encontrou apenas 18 regras de associação com suporte e confiança maiores ou iguais aos mínimos especificados.

Tabela 3.4: Parâmetros definidos pelo usuário

-N 50 (50 regras a serem descobertas);
-T 0 (não é arquivo de treinamento);
-C 0.75 (confiança mínima de 75%);
-D 0.05 (5% de variação para decréscimo do suporte mínimo);
-U 1.0 (100% como limite superior para suporte mínimo);
-M 0.1 (10% como limite inferior para suporte mínimo).

Tabela 3.5 - associacao.arff (arquivo de saída)

=== Run information ===
Scheme: weka.associations.Apriori -N 50 -T 0 -C 0.75 -D 0.05 -U 1.0 -M 0.1 -S -1.0
Relation: associacao
Instances: 2434
Attributes: 50
american
amp
analyst
bank
bid
bond
bush
busi
cent
compani
comput
corp
court
dai
democrat

Tabela 3.5 - associacao.arff (arquivo de saída) – (continuação)

dollar
firm
fund
govern
group
industri
investor
issu
japan
manag
market
month
peopl
plan
point
presid
price
product
quarter
rate
rjr
sale
secur
share
soviet
state
stock
system
takeov
tax

Tabela 3.5 - associacao.arff (arquivo de saída) – (continuação)

time
trade
week
year
york
=== Associator model (full training set) ===
Apriori
=======
Minimum support: 0.05
Minimum metric <confidence>: 0.75
Number of cycles performed: 17
Generated sets of large itemsets:
Size of set of large itemsets L(1): 34
Size of set of large itemsets L(2): 60
Size of set of large itemsets L(3): 10
Best rules found:
1. cent=1 442 ==> share=1 393 conf:(0.89)
2. share=1 year=1 510 ==> compani=1 434 conf:(0.85)
3. share=1 trade=1 506 ==> stock=1 429 conf:(0.85)
4. share=1 stock=1 570 ==> compani=1 476 conf:(0.84)
5. stock=1 trade=1 514 ==> share=1 429 conf:(0.83)

Tabela 3.5 - associacao.arff (arquivo de saída) – (continuação)

6. trade=1 york=1 450 ==> stock=1 371 conf:(0.82)
7. share=1 trade=1 506 ==> compani=1 414 conf:(0.82)
8. compani=1 stock=1 588 ==> share=1 476 conf:(0.81)
9. stock=1 york=1 461 ==> trade=1 371 conf:(0.8)
10. stock=1 trade=1 514 ==> compani=1 410 conf:(0.8)
11. share=1 904 ==> compani=1 716 conf:(0.79)
12. sale=1 year=1 482 ==> compani=1 380 conf:(0.79)
13. corp=1 year=1 561 ==> compani=1 439 conf:(0.78)
14. month=1 709 ==> year=1 553 conf:(0.78)
15. stock=1 768 ==> compani=1 588 conf:(0.77)
16. time=1 697 ==> year=1 530 conf:(0.76)
17. compani=1 sale=1 504 ==> year=1 380 conf:(0.75)
18. share=1 stock=1 570 ==> trade=1 429 conf:(0.75)

Capítulo 4

Resultados Computacionais

A seguir são descritos resultados computacionais de dois experimentos realizados com um subconjunto de documentos da base *Tipster* [Tipster 94]. O primeiro conjunto de experimentos, denominados experimentos preliminares, foi útil para mostrar que o sistema proposto está funcionando corretamente e para se ter uma idéia do tempo computacional requerido pelo sistema, enquanto o segundo conjunto de experimentos avaliou a influência de um dos parâmetros do algoritmo de associação do *Weka* no número de regras descobertas.

4.1. Experimentos Preliminares

O objetivo destes experimentos preliminares foi medir a variação no número de regras descobertas e na confiança média das regras para diferentes bases de documentos. Mais precisamente, utilizou-se cinco conjuntos de documentos da base *Tipster* [Tipster 94]. Todos esses 5 conjuntos contêm documentos referentes a notas e artigos retirados do *Wall Street Journal*, enfatizando assuntos administrativos, econômicos, tecnológicos, de relações públicas e marketing em estabelecimentos comerciais, industriais e bancários. Cada conjunto de documentos contém documentos de um ano diferente, a saber: 1988 a 1992.

O número total de experimentos será igual a 5 (cinco), isto é, um experimento para cada ano – de 1988 a 1992 inclusive.

Os parâmetros para tais experimentos estão definidos a seguir:

- -N 50 (no máximo 50 regras a serem descobertas);
- -C 0.75 (confiança mínima de 75%);
- -D 0.05 (5% de variação para decréscimo do suporte mínimo);
- -U 1.0 (100% como limite superior para suporte mínimo);
- -M 0.1 (10% como limite inferior para suporte mínimo).

O número de regras descobertas e a respectiva confiança média de tais regras, encontram-se listados anualmente na Tabela 4.1:

Tabela 4.1: Nº de Regras descobertas e Conf. Média Anuais

Ano	Nº Regras	Confiança Média
1988	18	80,4%
1989	50	80,5%
1990	43	80,1%
1991	11	80,9%
1992	50	80,5%

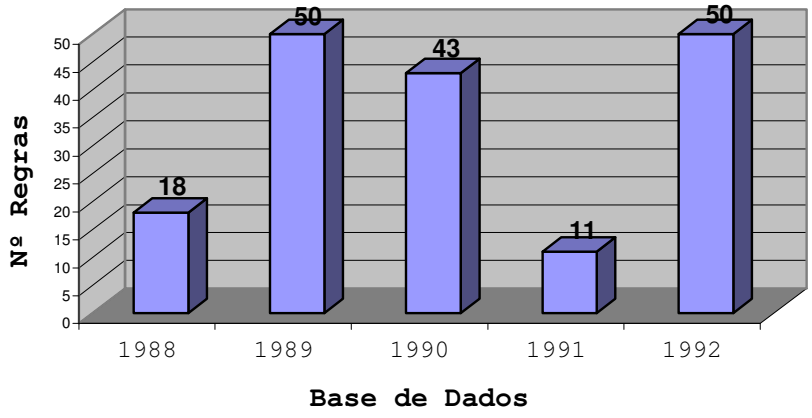

Figura 4.1– Nº de Regras Descobertas para cada conjunto de documentos

Examinando-se a Tabela 4.1, pode-se notar que houve uma considerável variação no número de regras descobertas entre os 5 conjuntos de documentos, conforme mostrado na Figura 4.1. Porém, a variação na confiança média das regras descobertas foi praticamente negligível.

Também é relevante relatar-se o tempo computacional em cada etapa envolvida no processo, bem como o tempo computacional total. As etapas envolvidas no processo são:

1ª etapa: Leitura e Pré-Processamento dos arquivos .txt da base de documentos *Tipster*. Lembre-se que em cada arquivo (documento), deve-se não apenas extrair suas palavras, mas também submetê-las aos passos de remoção se "*stop words"*, *stemming* e identificação de classes gramaticais, conforme descrito anteriormente;

2ª etapa: cálculo dos valores de *Tf-Idf* dos termos e geração dos arquivos *.arff* (padrão do arquivo de entrada para o *Weka*);

3ª etapa: processamento do algoritmo *Apriori* do *Weka*

O tempo computacional do processo está apresentado na Tabela 4.2: Em cada célula da tabela, o respectivo tempo computacional é mostrado no formato horas: minutos: segundos.

Tabela 4.2: Tempo Computacional do Processo

Tempo Computacional				
Etapas da Geração do Arquivo *.arff*			**Etapa de processamento do algoritmo *Apriori***	**Tempo Computacional Total**
Ano	**1ª etapa**	**2ª etapa**	**3ª etapa**	
1988	00:13:15	04:55:28	00:00:08	05:08:51
1989	00:21:02	06:16:34	00:00:10	06:37:46

Tabela 4.2: Tempo Computacional do Processo (continuação)

1990	00:15:42	06:09:48	00:00:09	06:25:39
1991	00:14:25	04:54:29	00:00:09	05:09:03
1992	00:23:37	06:17:04	00:00:09	06:40:50
MÉDIA	00:17:36	05:42:41	00:00:09	06:00:26

A Tabela 4.3, que destaca o fator "tamanho" dos arquivos processados, tem por objetivo clarificar as diferenças no tempo computacional gasto para o conjunto de documentos de cada ano, conforme apresentado anteriormente na Tabela 4.2.

Tabela 4.3: Tamanho dos arquivos

Arquivos Originais e Obtidos na etapa de Pré-processamento			
Ano	**Arquivos Originais***	**Arquivos Obtidos****	
	Tamanho (bytes)	**Quantidade**	**Tamanho (bytes)**
1988	7.103.534	2.434	2.500.971
1989	8.438.145	2.577	2.989.014
1990	8.206.331	2.456	2.792.831
1991	8.077.544	2.404	2.789.904
1992	8.810.064	2.479	3.173.698
Média	8.127.124	2.470	2.849.284

Arquivos Originais*: obtidos da base *Tipster* [Tipster 94]

Arquivos Obtidos**: resultados obtidos na etapa de pré-processamento de texto (1 arquivo obtido = 1 transação)

4.2. Experimentos Avaliando Influência da Variação do Suporte Mínimo no Número de Regras Descobertas

Um segundo experimento foi realizado com o objetivo de medir a variação no número de regras descobertas com base na redução gradativa do limite inferior do SupMín (suporte mínimo) para diferentes bases de documentos. Assim, este experimento envolve uma avaliação da capacidade de expansão (“*scalability”)* do sistema com relação à redução do suporte mínimo das regras descobertas.

Também para esse experimento, utilizou-se os mesmos cinco conjuntos de documentos da base *Tipster* [Tipster 94] já relatados no experimento anterior.

Assim, o número total de experimentos será igual a 5 (cinco), isto é, um experimento para cada ano – de 1988 a 1992 inclusive.

Os parâmetros para tais experimentos estão definidos a seguir:

- -N **variável para cada ano** (nº de regras a serem descobertas). Utilizou-se um número tal que permitisse ao algoritmo *Apriori* do *Weka* apresentar todas as regras descobertas conforme o valor do limite inferior que estivesse sendo utilizado no momento;
- -C 0.75 (confiança mínima de 75%);
- -D 0.05 (5% de variação para decréscimo do suporte mínimo);
- -U 1.0 (100% como limite superior para suporte mínimo);
- -M **assumiu 5 (cinco) variações para cada ano** (valor percentual, entre 0 e 100 exclusive, como limite inferior para suporte mínimo). Uma variação por vez neste parâmetro fez com que o número de vezes que ocorreu o processamento do algoritmo *Apriori* do *Weka* fosse igual ao número de variações para este parâmetro.

As variações do parâmetro correspondente ao limite inferior (L.I.) para suporte mínimo encontram-se listadas na Tabela 4.4:

Tabela 4.4: Valores assumidos pelo L.I. do SupMín

Nº da variação	**Valor assumido pelo L.I. do suporte mínimo**
1	0.20
2	0.15
3	0.1
4	0.05
5	0.01

O número de regras descobertas conforme as variações assumidas pelo limite inferior (L.I.) do suporte mínimo encontram-se listados, para cada ano da base de documentos, na Tabela 4.5:

Tabela 4.5: Nº de Regras descobertas para cada base anual de documentos

Nº de Regras descobertas						
Ano	**L.I.= 0.20**	**L.I.= 0.15**	**L.I.= 0.1**	**L.I.= 0.05**	**L.I.= 0.01**	**Instâncias**
1988	1	4	18	154	3.154	2.434
1989	2	7	53	457	14.750	2.577
1990	2	11	43	294	8.584	2.456
1991	0	3	11	165	4.334	2.404
1992	3	10	69	544	15.434	2.479

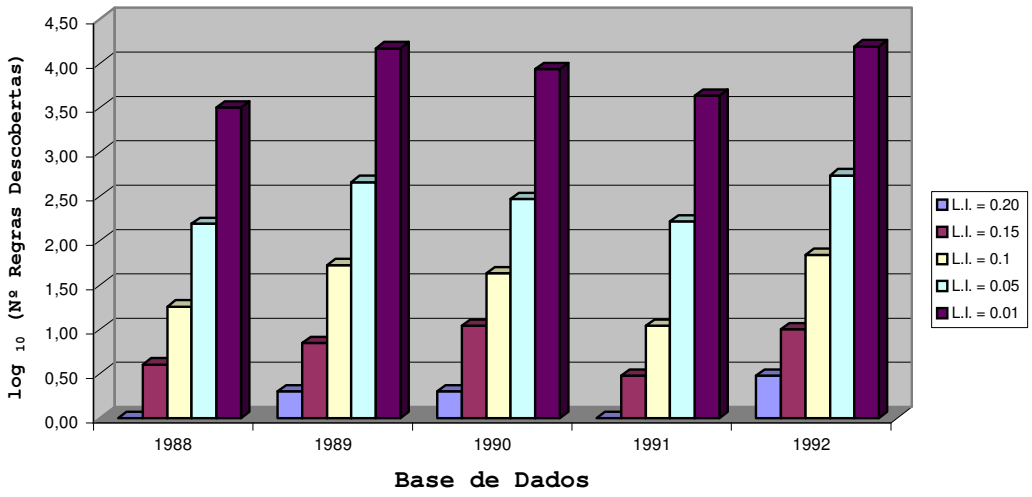

Figura 4.2: Nº de Regras Descobertas para cada conjunto de documentos

Examinando-se a Tabela 4.5, pode-se notar que houve uma considerável variação no número de regras descobertas segundo cada valor assumido pelo limite inferior do SupMín para os 5 conjuntos de documentos, conforme mostrado na Figura 4.2. Ressalta-se que, para melhoria na qualidade do gráfico obtido (na Figura 4.2), o número de regras descobertas está expresso em logaritmo na base 10.

Note que, conforme esperado (isso é uma característica de algoritmos de associação em geral), o número de regras descobertas cresce rapidamente com a redução do Suporte Mínimo. Em particular, quando o limite inferior do suporte mínimo é reduzido de 0,1 para 0,01 (uma redução de uma ordem de magnitude), o número de regras descobertas cresce em duas ordens de magnitude. Por exemplo, para o conjunto de documentos de 1989, o limite inferior de suporte mínimo igual a 0,1 resultou na descoberta de 53 regras, enquanto que um limite inferior de 0,01 resultou em 14.750 regras.

Também é importante relatar-se o tempo computacional envolvido para geração dos resultados apresentados na Tabela 4.5.

O tempo computacional do processo está apresentado na Tabela 4.6: Em cada célula da tabela, o respectivo tempo computacional é mostrado no formato horas: minutos: segundos. Nessa tabela, o tempo computacional é o tempo gasto com a execução do algoritmo de associação do *Weka* apenas, sem incluir o tempo gasto com pré-processamento.

Tabela 4.6: Tempo Computacional do Processo

Tempo Computacional					
Ano	**L.I. = 0.20**	**L.I. = 0.15**	**L.I. = 0.1**	**L.I. = 0.05**	**L.I. = 0.01**
1988	00:00:04	00:00:05	00:00:09	00:00:15	00:01:05
1989	00:00:05	00:00:07	00:00:10	00:00:21	00:01:09
1990	00:00:04	00:00:06	00:00:09	00:00:17	00:01:10
1991	00:00:04	00:00:06	00:00:09	00:00:15	00:01:12
1992	00:00:05	00:00:07	00:00:10	00:00:19	00:01:15

Note que, apesar do tempo computacional aumentar rapidamente com a redução do suporte mínimo (o que é, naturalmente, uma conseqüência do aumento no número de regras descobertas), em geral o tempo computacional não foi muito excessivo nestes experimentos.

Deve-se lembrar que, na prática, em geral não seria muito útil retornar ao usuário milhares de regras, o que acontece quando o limite inferior do suporte mínimo é ajustado para 0,01. Assim, desde que se use um limite inferior de suporte mínimo que resulte em um número não muito alto de regras (digamos, menos de 100 regras), o sistema parece ser relativamente rápido.

4.3. Regras Descobertas

Para ilustrar as regras de associação descobertas pelo sistema, apresenta-se nesta seção as regras descobertas para um dado ano, a dizer, o ano de 1992, oriundas da base de dados *Tipster* [Tipster 94].

Para cada uma das regras será relatada a regra em si (em seu formato **se - então**), o valor do *Tf-Idf* médio da regra obtido somando-se o *Tf-Idf* de todos os itens da regra e dividindo-se esse total pelo número de itens da regra, o valor do suporte da regra e o valor da confiança da regra. As regras (no formato **se – então**) e seus respectivos valores do *Tf-Idf* médio, suporte e confiança apresentam-se na Tabela 4.7.

Os valores dos parâmetros usados para obter as regras da Tabela 4.7 são:

- -N 10.000 (no máximo10.000 regras a serem descobertas);

- -T 0 (não é arquivo de treinamento);
- -C 0.75 (confiança mínima de 75%);
- -D 0.05 (5% de variação para decréscimo do suporte mínimo);
- -U 1.0 (100% como limite superior para suporte mínimo);
- -M 0.15 (15% como limite inferior para suporte mínimo).

Cabe ressaltar que várias regras representam associações pouco interessantes entre o antecedente e o conseqüente; como por exemplo a **regra 1** da Tabela 4.7, representando uma associação não-surpreendente entre "mês" e "ano".

Assim, esses resultados sugerem que uma importante direção de pesquisa futura consiste em desenvolver medidas de interesse de regras que promovam a descoberta de regras mais interessantes, representando associações mais surpreendentes para o usuário.

Tabela 4.7: Regras de Associação Descobertas

Nº da Regra	**Descrição da Regra**	**Valor do *Tf-Idf* médio da Regra**	**Valor do Suporte da Regra**	**Valor da Confiança da Regra**
1	**se** (*month*) **então** (*year*)	891,71	0,24	0,82
2	**se** (*execut*) **então** (*year*)	882,63	0,22	0,82
3	**se** (*presid*) **então** (*year*)	951,83	0,28	0,81
4	**se** (*amp* E *corp*) **então** (*compani*)	900,36	0,20	0,80
5	**se** (*price*) **então** (*market*)	1.459,77	0,20	0,79
6	**se** (*share*) **então** (*compani*)	1.287,40	0,25	0,79
7	**se** (*trade*) **então** (*year*)	1.099,29	0,21	0,76
8	**se** (*amp* E *market*) **então** (*compani*)	1.116,08	0,24	0,76
9	**se** (*sale*) **então** (*compani*)	1.045,04	0,22	0,75
10	**se** (*corp*) **então** (*compani*)	963,45	0,30	0,75

4.4. Medida de Interesse de Regras

Conforme citado na seção anterior, várias regras representam associações pouco interessantes entre o antecedente e o conseqüente; como por exemplo a **regra 1** da Tabela 4.7, representando uma associação não-surpreendente entre "mês" e "ano".

Assim, esses resultados sugeriram que fosse utilizado um mecanismo para medir o interesse das regras descobertas. Esta medida foi feita através da computação das distâncias entre cada par de palavras P_1 e P_2, onde P_1 é uma palavra do antecedente da regra e P_2 é uma palavra do conseqüente da regra de associação descoberta. Utilizou-se o software de domínio público *WordNet* [Miller 98] para a obtenção das distâncias entre os pares de palavras das dez regras descobertas no tópico 4.3 deste trabalho.

O procedimento para a obtenção das distâncias está brevemente relatado a seguir. Como dado de entrada ao software, digita-se a palavra para a qual se deseja conhecer o respectivo posicionamento na hierarquia do *WordNet*. A pesquisa neste trabalho somente refere-se a substantivos (*nouns)*. Para encontrar-se o respectivo posicionamento da palavra na hierarquia, seleciona-se a opção exibição de seus Hiperônimos, isto é, a palavra em questão **sendo – parte - de** palavras com nível superior hierárquico.

O procedimento relatado no parágrafo anterior repete-se para cada palavra pertencente às dez regras de associação descobertas do tópico 4.3 deste trabalho.

Após terem sidos descobertos todos os níveis superiores às palavras pertencentes às regras, pode-se exibir um diagrama representando a hierarquia encontrada segundo o software *WordNet*, conforme Figura 4.3.

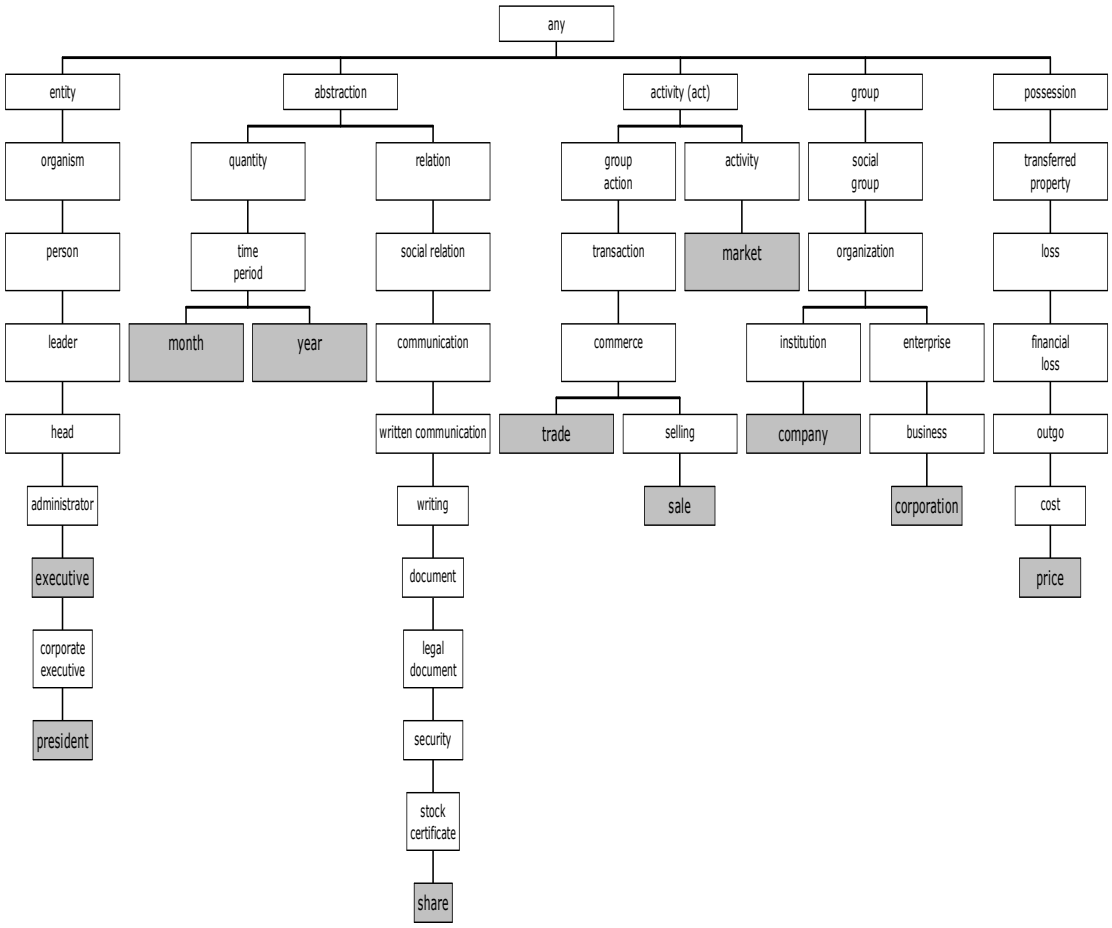

Figura 4.3: Diagrama dos hiperônimos encontrados no *WordNet* para palavras pertencentes às regras de associação descobertas

A distância entre cada par de palavras foi obtida calculando-se o número de arestas conectando duas palavras, P_1 e P_2, respectivamente do antecedente e do conseqüente de cada regra de associação descoberta, ou seja, inicialmente identifica-se a primeira palavra ancestral comum a P_1 e P_2, na hierarquia de hiperônimos. Seja P_a esse ancestral. Então soma-se o número de arestas de P_1 a P_a com o número de arestas de P_a a P_2. O resultado dessa soma é o número de arestas conectando P_1 a P_2. Quanto maior essa medida de distância, mais "interessante" a regra em questão. Assim obteve-se a Tabela 4.8, que apresenta as distâncias calculadas para todas as dez regras de associação descobertas na seção 4.3 deste trabalho.

Tabela 4.8: Distâncias das palavras pertencentes às Regras de Associação Descobertas

Nº da Regra	**Descrição da Regra**	**Distância em número de arestas**
1	**se** (*month*) **então** (*year*)	2
2	**se** (*execut*) **então** (*year*)	11
3	**se** (*presid*) **então** (*year*)	13
4	**se** (*amp* E *corp*) **então** (*compani*)	5
5	**se** (*price*) **então** (*market*)	10
6	**se** (*share*) **então** (*compani*)	16
7	**se** (*trade*) **então** (*year*)	9
8	**se** (*amp* E *market*) **então** (*compani*)	8
9	**se** (*sale*) **então** (*compani*)	11
10	**se** (*corp*) **então** (*compani*)	5

Ressalta-se que existem vários casos em que o resultado obtido através do *WordNet* foi satisfatório, mas também existem casos em que o resultado obtido não foi considerado satisfatório devido ao *WordNet* não tratar relacionamentos temáticos. Por exemplo, na **regra 1** da Tabela 4.8, é satisfatória a distância encontrada entre as palavras "*month"* e "*year"*, mostrando que ambas estão bastante próximas na hierarquia do *WordNet.* Essa pequena distância sugere, corretamente, que a regra não é interessante, por envolver um óbvio relacionamento entre palavras próximas na hierarquia de hiperônimos do *WordNet*. Porém, cita-se o exemplo da **regra 5** da Tabela 4.8, aonde a distância encontrada entre as palavras "*price"* e "*market"* é relativamente grande, sugerindo erroneamente que a regra seria interessante. Na verdade, visto do ponto de vista do relacionamento temático entre ambas, a regra representa um relacionamento relativamente óbvio, e portanto pouco interessante. Logo, pode-se destacar um resultado considerado satisfatório do *WordNet* como o apresentado na **regra 1** e um resultado não satisfatório do *WordNet* como o apresentado na **regra 5**.

Capítulo 5

Trabalhos Relacionados

A seguir são descritos trabalhos relacionados à descoberta de regras de associação em textos. Cabe ressaltar que, na pesquisa bibliográfica realizada pela autora, não foi encontrado nenhum trabalho utilizando a medida de *Tf-Idf* para selecionar as palavras (itens) mais relevantes para serem fornecidas como entrada a um algoritmo de associação. Assim, os trabalhos descritos a seguir são relacionados a essa dissertação em um alto nível de abstração, tendo como similaridade a extração de conhecimento a partir de texto.

5.1. Trabalhos gerais sobre a Descoberta de Conhecimento em Bases de Dados tipo Texto (KDT)

Segundo [Feldman & Dagan 95], para aplicar métodos de *data mining* à análise de textos devem ser consideradas as diversas limitações da tecnologia de processamento de textos, possibilitando a definição de estruturas o mais simples possível que possam ser extraídas dos textos quase que de uma forma automática e com um custo razoável. Em [Feldman & Dagan 95] é proposto um método que faz uso de um paradigma de categorização de texto que percebe partes do texto com conceitos significativos e que são organizados numa estrutura hierárquica. Também é sugerido que tal percepção relativamente simples seja rica o suficiente para fins de *data mining*, facilitando a sumarização de dados e a exploração de padrões interessantes. Destacam-se três componentes básicos no método proposto naquele trabalho: a definição do conceito de hierarquia, a categorização de textos através dos conceitos de hierarquia e a comparação das distribuições de encontrar padrões inesperados.

Para apresentar informação interessante para o usuário, é sugerida a quantificação de níveis de interesse de alguns dados através da comparação desse dado em relação a um dado fornecido por um modelo conhecido.

5.2. Mineração de Texto via Extração de Informações

Para [Feldman et al 99], dada uma coleção de documentos texto, a maioria dos métodos utilizados para que a mineração de texto possa executar operações de descoberta de conhecimento faz uso de rótulos (*labels*) associados com cada documento. Em alguns casos, esses rótulos são palavras-chave (*keywords*) que representam os resultados de processos não triviais do tipo rotulação de palavras chaves, em outros casos, tais rótulos (*labels*) não são nada mais do que uma lista de palavras interessantes contidas em documentos. Em [Feldman et al 99], é apresentado um método intermediário na qual a descoberta de conhecimento toma lugar em uma coleção focada de eventos e expressões extraídos de e rotulados em cada documento. Tais eventos agregados a entidades de alto nível de interesse são organizados numa taxonomia hierárquica e são usados em processos de descoberta de conhecimento.

Esse método foi implementado num sistema chamado *Textoscope*. O sistema *Textoscope* consiste de:

a) um módulo de recuperação de documentos, que converte documentos recuperados nos formatos originais para documentos no formato SGML usados pelo *Textoscope*;
b) um mecanismo de extração de informações que baseia-se em atributos gramaticais poderosos e ampliados por uma valiosa base de conhecimento;
c) uma ferramenta de criação taxonômica através da qual o usuário pode auxiliar na especificação de entidades de alto nível de interesse;
d) um conjunto de ferramentas de descoberta de conhecimento para reconhecer o resultado de documentos rotulados por eventos.

Os resultados dos experimentos realizados com o uso do sistema *Textoscope* confirmaram que a mineração de textos via extração de informações serve como uma técnica adequada para gerenciar conhecimento encapsulado em grandes coleções de documentos.

5.3. *TextVis*: Um Ambiente Visual Integrado para Mineração de Texto

Em [Feldman et al 98-b], *TextVis* é apresentado como um sistema visual de mineração de dados para coleções de documentos.

TextVis faz uso de um recurso multi-estratégia para a mineração de textos, habilitando esquemas complexos de análise a partir de componentes básicos promovidos pelo sistema.

O sistema de análise é constituído por ícones funcionais de arrasto de uma paleta de ferramentas contida na área de trabalho, conectados de acordo com a indicação do fluxo da informação.

O sistema oferece uma coleção de ferramentas básicas de análise, tais como:

a) conjuntos freqüentes;
b) associações;
c) distribuições;
d) correlações.

Os padrões descobertos estão presentes na interface visual para dar suporte ao usuário a operar os resultados e para acessar documentos associados.

TextVis é um sistema de mineração de texto que usa tecnologia baseada em agentes para acessar várias fontes de informação online, ferramentas de pré-processamento de texto para extrair informações relevantes dos documentos, uma variedade de algoritmos de mineração de dados e um conjunto de navegadores visuais para visualização dos resultados.

5.4. *Clasitex*: Uma Ferramenta para Descoberta de Conhecimento em Textos

Clasitex é apresentado em [Trinidad et al 98] como um sistema que descobre os tópicos mais importantes tratados em textos escritos em espanhol ou em inglês. Esse sistema trabalha baseado em árvores de conceitos e encontra:

a) os conceitos mais freqüentes no texto;

b) a relação entre esses conceitos, computando a co-ocorrência desses conceitos nas sentenças que se ajustam ao texto.

Além disso, o sistema *Clasitex* fornece um mapa de distribuição dos conceitos mais freqüentes no texto, sendo uma característica importante do sistema a manipulação de uma certa quantidade de conceitos em espanhol e em inglês num tempo de execução bastante aceitável.

5.5. Representação Contextual de Texto para Descoberta Não-supervisionada de Conhecimento em Textos

Em [Perrin & Petry 98], foi demonstrado que uma estrutura de texto e um conteúdo, ambos bastante úteis, podem ser extraídos sistematicamente através da análise léxica colocacional sem precisar de nenhum outro recurso extra de conhecimento. São descritos algoritmos que modelam extração autônoma de fatos relevantes de um texto e associam cada um deles com um rótulo obtido na extração automática do tópico de um discurso estruturado. As vantagens desse método são:

a) Métodos estatísticos de extração e reordenação são atraentes, principalmente por serem simples e serem independentes dos domínios de aplicação. Eles apenas dependem do conteúdo atual de um texto.
b) Não há uma divisão muito acentuada nos textos, o que reduz consideravelmente a complexidade na utilização de determinada informação. Porém aqui também não se deve desconsiderar o fato de textos serem narrativas contendo jargões, sentenças não muito bem estruturadas, etc. que são os reais desafios para gramática de Processamento em Linguagem Natural (PLN) e analisadores de textos sem restrições.

Algumas limitações existentes referem-se ao algoritmo de extração do conteúdo do texto não considerar a relação semântica entre as palavras contidas no texto. O algoritmo somente fornece um resultado resumido das palavras com associações mais relevantes aos usuários.

5.6. Extração de Termos Chaves num Domínio Específico

Expressões por si mesmas não possuem grande valor – suas propriedades ou atributos é que são importantes [Witten et al 99]. Quando atributos são referenciados, muitos deles vêm imediatamente à mente: o número de palavras numa expressão, o número de caracteres, a posição da expressão no documento, etc.

Porém, nos experimentos realizados por [Witten et al 99], somente dois atributos apareceram como discriminantes entre expressões – chaves e expressões - não-chaves. São eles:

a) o valor de *Tf-Idf* de uma expressão e
b) a distância da primeira aparição da expressão em relação ao início de um documento.

O valor de *Tf-Idf* de uma expressão é uma medida padrão em recuperação de informação conforme visto na seção 3.2.1 desta dissertação.

A distância de uma expressão em relação ao início de um documento é calculada como sendo o número de palavras que a precede em sua primeira aparição dividida pelo número de palavras do documento.

Conclusão

Este trabalho propôs um método para extrair automaticamente regras de associação entre "itens de informação" (palavras) contidas em documentos. O método utiliza um algoritmo para descoberta de regras de associação. Esse tipo de algoritmo tem a vantagem de descobrir regras no formato **se - então**, fáceis de serem interpretadas pelo usuário.

Porém algoritmos de regras de associação têm a desvantagem de exigirem um alto tempo de processamento, particularmente quando o número de itens é muito alto (o que é uma situação comum na prática, e em geral é o caso em *text mining*).

Para reduzir esse problema, além de usar de técnicas básicas de pré-processamento de textos em recuperação de informações (remoção de *stop words* e *stemming*), este trabalho propôs que itens de informação fossem formados apenas por substantivos e propôs um novo método de pré-processamento para identificação dos itens de informação mais relevantes. Esse método é baseado no cálculo do *Tf-Idf* (*Term Frequency – Inverse Document Frequency*) para cada palavra, ou item de informação. Apenas as palavras com maiores valores de *Tf-Idf* são fornecidas como entrada para o algoritmo de descoberta de regras de associação. Isso permite uma considerável redução no número de palavras fornecidas como entrada para o algoritmo de associação, o que tende a reduzir bastante o tempo de processamento daquele algoritmo.

O sistema proposto foi utilizado para extração de regras de associação a partir de documentos da base *Tipster* [Tipster 94], uma base de documentos freqüentemente usada para avaliação de algoritmos de mineração de textos na literatura.

Apesar do método de pré-processamento proposto ser eficaz na redução do tempo computacional e ainda assim permitir a descoberta de regras com alto fator de confiança, os resultados revelaram que algumas regras descobertas não representavam conhecimento interessante para o usuário, por refletirem um relacionamento óbvio entre palavras no antecedente e no conseqüente da regra. Isso levou à proposta de uma medida de interesse das regras descobertas, utilizando-se a hierarquia de hiperônimos do software *WordNet*. A medida

de interesse proposta revelou-se razoavelmente eficaz (mas certamente não-perfeita) na avaliação do grau de interesse de regras descobertas, em alguns experimentos preliminares.

Assim, algumas direções para pesquisa futura são sugeridas a seguir. Inicialmente, cabe ressaltar que a medida de interesse de regras baseada no *WordNet* foi calculada manualmente para umas poucas regras descobertas.

No futuro poder-se-á automatizar o cálculo dessa medida, o que permitirá a realização de experimentos mais extensos sobre sua eficácia.

Além disso, outras medidas de interesse de regras poderão ser investigadas no futuro, talvez utilizando-se outras informações disponíveis no *WordNet*.

Outra direção para pesquisas futuras seria estender o algoritmo de associação para descobrir regras de associações hierárquicas, lidando com itens de informações hierárquicos.

Referências Bibliográficas

[Agrawal et al 93] AGRAWAL, R., IMIELINSKI, T., SWAMI, A . Mining Association Rules between Sets of Items in Large Databases. *Proc. of the Int. Conf on Management of Data (SIGMOD-93)*, 207-216. Washington, DC, USA, 1993.

[Berson & Smith 97] BERSON, A . & SMITH, S.J. *Data Warehousing, Data Mining and OLAP*. Nova Iorque, USA. McGraw-Hill, 1997.

[Brill POS] BRILL, E. *Brill POS Tagger Output*. Disponível em: <http://nlp01.cs.ul.ie/cgi-bin/nlps_post_query2_dev> Acesso em: 04 de novembro de 2001.

[Cardie 97] CARDIE, C. Empirical Methods in Information Extraction. *AI Magazine*, 18 (4), 65 - 79. Winter, 1997.

[Feldman & Dagan 95] FELDMAN, R. & DAGAN, I. Knowledge Discovery in Textual Databases (KDT). In *Proc. of the 1st International Conference on Knowledge Discovery and Data Mining (KDD-95),* 112 – 117. AAAI / MIT Press: Menlo Park, CA, 1995.

[Feldman et al 98] FELDMAN, R., FRESKO, M., KINAR, Y., LINDELL, Y., LIPHSTAT, O ., RAJMAN, M., SCHLER, Y., ZAMIR, O . Text Mining at the Term Level. Lecture Notes in Artificial Intelligence (1510). (*Proc. of the PKDD*-98), 65-73. Springer-verlag, 1998.

[Feldman et al 98-b] FELDMAN, R., FRESKO, M., LANDAU, D., LINDELL, Y., LIPHSTAT, O., AUMANN, Y., ZAMIR, O. TextVis: An Integrated Visual Environment for Text Mining. Lecture Notes in Artificial Intelligence (1510). In *Proc. of the PKDD*-98, 56-64. Springer-Verlag, 1998.

[Feldman et al 99] FELDMAN, R., AUMANN, Y., FRESKO, M., LIPHSTAT, O., ROSENFELD, B., SCHLER, Y. Text Mining via Information Extraction. Lecture Notes in Artificial Intelligence (1704). In *Proc.of the PKDD-99,* 165-173. Springer-Verlag, 1999.

[Fellbaum 98] FELLBAUM, C. (Ed.) *WordNet: an Electronic Lexical Database*. MIT Press, 1998.

[Freitas & Lavington 98] FREITAS, A .A. & LAVINGTON, S.H. *Mining Very Large Databases with Parallel Processing*. Kluwer Academic Publishers, 1998.

[Manning & Schutze 99] MANNING, C.D. & SCHUTZE, H. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[McDonald 96] McDONALD, D.D.. Internal and External Evidence in the Identification and Semantic Categorization of Proper Names. In: Branimir Boguraev and James Pustelovsky (Eds.) *Corpus Processing for Lexical Acquisition*, 21-39. MIT Press, 1996.

[Miller 98] MILLER, G. Nouns in WordNet. In: Christiane Fellbaum (Ed.). *WordNet: an Electronic Lexical Database*, 23 - 46(1). MIT Press, 1998.

[Perrin & Petry 98] PERRIN, P. & PETRY, F. Contextual Text Representation for Unsupervised Knowledge Discovery in Texts. Lecture Notes in Artificial Intelligence (1394). In *Proc. of the PKDD*-98, 246-257. Springer-Verlag, 1998.

[Porter 86] PORTER's stemming algorithm. Disponível em: <http://softlab.od.ua/algo/text/stem.html > Acesso em: 04 de novembro de 2001.

[Porter 97] PORTER, M.F. An algorithm for suffix stripping. In *Readings in Information Retrieval*, 313-316. Morgan Kaufmann, 1997.

[Srikant & Agrawal 95] SRIKANT, R. & AGRAWAL, R. Mining Generalized Association Rules. *Proc. of the 21st Very Large Databases Conference*. Zurique, Suíça. 1995.

[Stop word list] SANDERSON, M. *Information Retrieval linguistic utilities.* Disponível em: <http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words> Acesso em: 04 de novembro de 2001.

[Tipster 94] MERCHANT, R. (Ed.) *Proc. of the Tipster Text Program - Phase I.* Morgan Kaufmann, 1994.

[Trinidad et al 98] TRINIDAD, J.F.M., MARTÍNEZ, B.B., ARENAS, A.G., SHULCLOPER, J.R. CLASITEX: A Tool for Knowledge Discovery from Texts. Lecture Notes in Artificial Intelligence (1510). In *Proc. of the PKDD*-98, 459-467. Springer-Verlag, 1998.

[Witten & Frank 99] WITTEN, I.H. & FRANK, E. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation*. San Francisco: Morgan Kaufmann, 2000.

[Witten et al 99] WITTEN, I.H., FRANK, E., PAYNTER, G.W. Domain-Specific Keyphrase Extraction. In *Proc. of the IJCAI-99 and ICML-99*, 668-673. 1999.