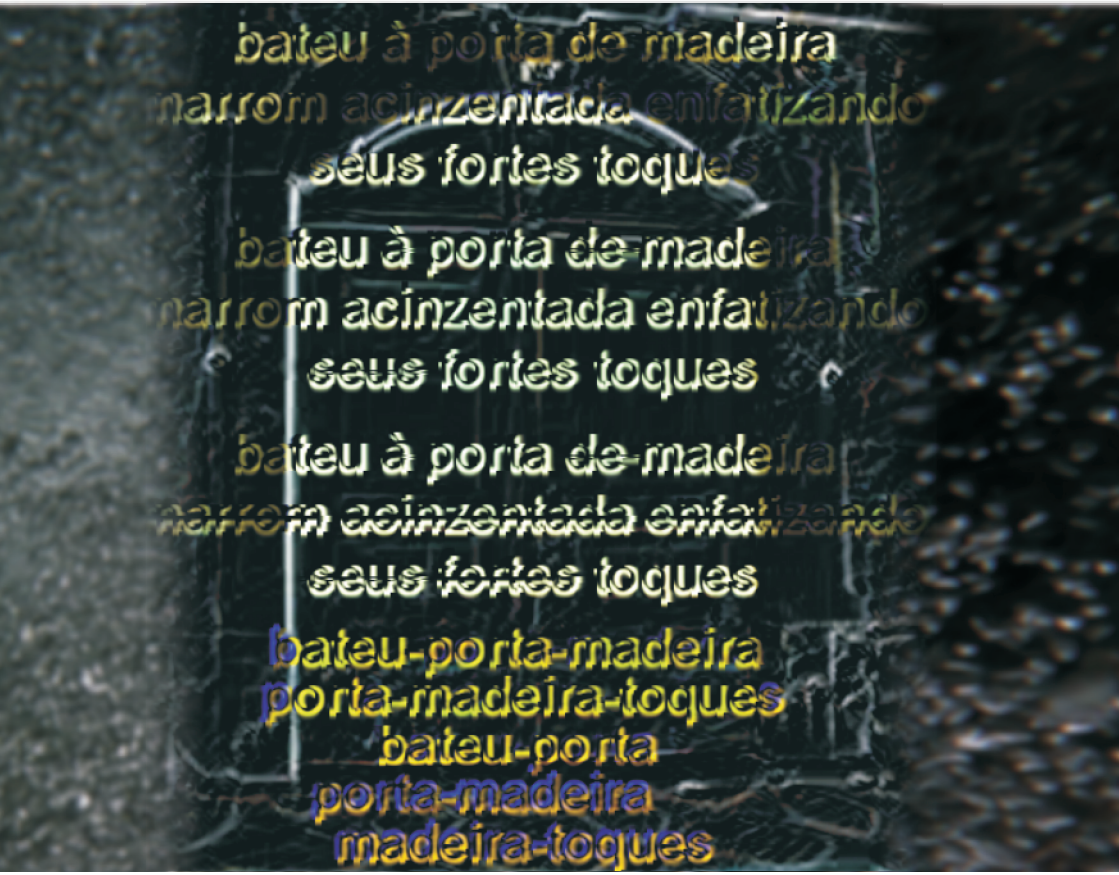
Boletim de Pesquisa e Desenvolvimento 23
ISSN 1677-9266
dezembro, 2010

Um modelo para a seleção de *n-gramas* significativos e não redundantes em tarefas de mineração de textos



Embrapa

ISSN 1677-9266
Dezembro, 2010

Empresa Brasileira de Pesquisa Agropecuária
Embrapa Informática Agropecuária
Ministério da Agricultura, Pecuária e Abastecimento

Boletim de Pesquisa e Desenvolvimento 23

Um modelo para a seleção de *n-gramas* significativos e não redundantes em tarefas de mineração de textos

Maria Fernanda Moura
Bruno Magalhães Nogueira
Merley da Silva Conrado
Fabiano Fernandes dos Santos
Solange Oliveira Rezende

Embrapa Informática Agropecuária
Campinas, SP
2010

Embrapa Informática Agropecuária
Av. André Tosello, 209 - Barão Geraldo
Caixa Postal 6041 - 13083-886 - Campinas, SP
Fone: (19) 3211-5700 - Fax: (19) 3211-5754
www.cnptia.embrapa.br
sac@cnptia.embrapa.br

1ª edição on-line 2010

Dados Internacionais de Catalogação na Publicação (CIP)
Embrapa Informática Agropecuária

Um modelo para a seleção de *n-gramas* significativos e não redundantes em tarefas de mineração de textos / Maria Fernanda Moura... [et al.]. - Campinas : Embrapa Informática Agropecuária, 2010.

37 p. : il. - (Boletim de pesquisa e desenvolvimento / Embrapa Informática Agropecuária , ISSN 1677- 9274 ; 23).

1. Atributo. 2. Mineração de texto. 3. Dados categorizados. I. Moura, Maria Fernanda. II. Titulo. II. Série.

CDD 006. 3 (21. ed.)

Sumário

Um modelo para a seleção de *n-gramas* significativos e não redundantes em tarefas de mineração de textos

Maria Fernanda Moura[1]
Bruno Magalhães Nogueira[2]
Merley da Silva Conrado[3]
Fabiano Fernandes dos Santos[4]
Solange Oliveira Rezende[5]

Resumo

Uma proposta completa para resolver o problema de selecionar automaticamente atributos não redundantes do tipo *n-gramas* é apresentada neste trabalho. Geralmente, o uso de *n-gramas* é um requisito para melhorar a interpretação subjetiva dos resultados em tarefas de mineração de textos, nesses casos, eles são estatisticamente gerados e selecionados. Após a seleção, em geral, há a presença de redundâncias, por exemplo, o termo "informática agropecuária" e seus componentes "informática" e "agropecuária". Assim, propõe-se um modelo que envolve a remoção de *stopwords* estatisticamente identificadas, uma seleção estatística eficiente para os atributos do tipo *n-grama* e a remoção das redundâncias apresentadas após a seleção. Observa-se, pelos resultados experimentais apresentados,

[1] *Doutora em Ciências de computação e matemática computacional, Pesquisadora da Embrapa Informática Agropecuária, fernanda@cnptia.embrapa.br*

[2] *Universidade de São Paulo*

[3] *Universidade de São Paulo*

[4] *Universidade de São Paulo, fabianof@icmc.usp.br*

[5] *Universidade de São Paulo, solange@icmc.usp.br*

sobre os atributos originais e os atributos sem as redundâncias, que, como esperado, após a eliminação das redundâncias não há perda de representatividade. Além disso, a redução no número de atributos é expressiva, o que pode significar ganhos em desempenho nas tarefas de extração de padrões, bem como na interpretabilidade subjetiva dos resultados. Deve-se salientar que o método proposto é útil a qualquer algoritmo de aprendizado de máquina aplicado a uma tarefa de mineração de textos, e, parece ser igualmente aplicável a textos em quaisquer línguas.

Termos para indexação: seleção de atributos, *n-gramas*, atributos redundantes, mineração de textos, dados categorizados

A model to select significant and non-redundant *n-grams* in texts

Abstract

A new and complete proposal to solve the automatic selection of non-redundant and significant n-grams attributes is presented in this paper. Generally, *n-grams* are used to facilitate the interpretation of the results in text mining tasks; and they are statistically extracted and selected from the texts. After the selection, there are always some kind of redundancy, as in “agriculture informatics” and its components “agriculture” and “informatic”. In this way, we propose a model which removes stopwords lists, which are statistically determined, as well as a statistical efficient method to select the *n-gram* attributes and to remove the presented redundancy. As expected, the experimental results show an equivalence between the attribute set with redundant elements and the attribute set without the redundancies. Besides, the reduction in the attribute set length is expressive, which can means some gain in the text mining tasks performance as well as in the results interpretation. It has to be highlighted that the proposed method is useful to any machine learning algorithm used in a text mining task, and it seems to be language independent.

Index terms: attribute selection, n-grams, redundant attribute, text mining, categorical data

1 Introdução

O processo de mineração de textos é dividido em cinco passos: identificação do problema, pré-processamento, extração de padrões, pós-processamento e uso do conhecimento. Frequentemente, o pré-processamento tende a ser considerado como um passo de menor importância, ou de menor interesse que os demais, devido à ausência de *glamour* técnico e excesso de tarefas manuais. Basicamente, este passo é responsável pela transformação da coleção de textos em uma forma útil aos algoritmos de aprendizado de máquina. Logo, envolve operações como o tratamento e limpeza dos dados e redução de atributos. Neste passo, as características mais representativas da coleção devem ser consideradas, procurando-se eliminar as características irrelevantes. Das dificuldades e desafios impostos neste passo, nada triviais, resultam a boa ou má qualidade dos dados analisados e, consequentemente, a eficiência dos algoritmos de aprendizado de máquina utilizados e grande parte da confiança nos resultados obtidos. Além disso, é importante considerar o conhecimento explícito carregado pelos atributos de um processo de mineração de textos, pois os resultados finais ou intermediários sempre podem ser subjetivamente analisados enriquecendo todo o processo (WEISS et al., 2005).

A probabilidade de considerar um atributo como um termo de domínio em uma coleção de textos pode ser entendida como uma propriedade qualitativa e discriminativa. Essa propriedade deve ser buscada tanto para palavras simples como compostas, como “inteligência” e “inteligência artificial”, na mesma coleção de textos. Embora muitos autores considerem o aumento de dimensionalidade mais prejudicial que benéfico, o aumento do número de atributos, quando se usam palavras simples e compostas parece facilitar a interpretação de resultados e, consequentemente, ser mais adequado aos usuários dos resultados em mineração de textos (MIAO et al., 2005; MOURA ; REZENDE, 2007). Em geral, quando se trabalha com palavras simples e compostas, ocorrem algumas redundâncias. Nesses casos, poder-se-ia descartar alguns desses atributos por meio de um filtro que lhes fosse adequado. Há poucas iniciativas na literatura para filtrar atributos entre *n-gramas*, exceto pelos trabalhos de Dejun e Maosong (2004) e Zhang e Zhu (2007), resumidos a seguir.

De acordo com Dejun e Maosong (2004) há um tipo de redundância resultante de características sobrepostas para caracteres *n-gramas* em chinês. Caracteres *n-gramas* são as composições de radicais ou partes de palavras, que para línguas orientais, como o chinês e japonês, muitas vezes correspondem a uma palavra ou a um conceito. Então, eles propuseram um modelo para evitar as redundâncias, sendo a sobreposição estimada e absorvida pelo modelo de atributos; e, eles acreditam que esse modelo seja independente de língua. Além disso, o modelo não utiliza rótulos (de classes) na seleção dos *n-gramas*, logo, não se restringe a tarefas de aprendizado supervisionado. Na proposta de Zhang e Zhu (2007), um comprimento de janela entre palavras que formam um *n-grama* é estimado em um modelo empírico, que procura por um valor discriminativo para esse tamanho, baseado na frequência dos *n-gramas* nos documentos e em relação à classe. Consequentemente esse modelo é mais adequado a aprendizado supervisionado, porém, é uma solução interessante para remover palavras entre os *n-gramas* que efetivamente interessam.

Neste trabalho é apresentada uma proposta para resolver o problema de selecionar automaticamente bons atributos do tipo *n-grama*, isto é, atributos estatisticamente significativos na coleção de textos e não redundantes. O modelo é útil para tarefas de aprendizado não supervisionado, supervisionado ou semi-supervisionado, pois depende exclusivamente de testes estatísticos sobre os atributos candidatos. Na próxima seção é apresentada a contextualização do problema, definindo os conceitos utilizados e explicando o que está sendo considerado redundância. Então, apresenta-se a metodologia proposta para selecionar os *n-gramas* e eliminar-lhes as redundâncias, bem como os algoritmos para implementar o processo, apresentado por Moura et al. (2008a). Na seção de experimentos e discussão, pode-se observar que a redução do espaço de atributos não sacrifica os resultados obtidos; o que é um bom indício de que os objetivos foram satisfeitos. Finalmente, são resumidas as conclusões e enumerados alguns trabalhos futuros.

2 Descrição do problema

Em um processo de mineração de textos escolhe-se a forma de representação dos atributos, após selecionar a coleção de textos com a qual se vai trabalhar (MOURA et al., 2008b). Supondo que se vá trabalhar exclusivamente com a coleção de textos como uma "*bag of words*", desconsiderando o contexto semântico e/ou ordem de ocorrência, utilizam-se as palavras presentes nos textos ou suas combinações. Essas palavras ou suas combinações são chamadas de *n-gramas*. Por exemplo, um *n-grama* pode ser o **unigrama** "plantio", ou o **bigrama** "plantio direto", ou o **trigrama** "plantio direto palha", ou outros. Ainda, podem-se reduzir as palavras a radicais, lemas ou outras formas simplificadas; por exemplo, removendo inflexões (aplicando um processo de *stemming*) os *n-gramas* do exemplo ficariam: **unigrama** "planti", **bigrama** "planti-diret" e **trigrama** "planti-diret-palh" . Seja qual a forma escolhida, necessariamente tem-se um enorme conjunto de atributos; desse conjunto podem ser descartados aqueles atributos que são pouco discriminativos e/ou redundantes, ou seja, aqueles que não são indispensáveis ao processo.

Para acompanhar o desenvolvimento das ideias apresentadas, será utilizada a seguinte coleção de cinco textos, completamente fictícios (Tabela 1).

Tabela 1. Coleção de cinco textos fictícios utilizados como exemplo.

Texto1: Textual data mining is data mining over textual data. Textual data are textual collections in several formats. Data mining has several definitions, but data are always data that are aimed at a special meaning and use.
Texto2: Textual data mining is used in business process. It is very useful for knowledge management.
Texto3: Textual data mining is a special kind of data mining. It works with textual data in textual collections. Data has to be transformed in useful knowledge.
Texto4: Textual data mining is a powerful tool. Many applications use textual data to discover market tendencies. Textual data can be magazine articles, internet chats, etc, every textual collection in a digital media. Data mining over textual data is the process which transforms data in useful information.
Texto5: Textual data mining is the subject of this text. We like to talk about data mining over textual data. Textual data are in textual collections. Data mining is the process of transforming data in useful information.

Embora essa coleção seja pequena e, aparentemente, gere poucos atributos (*n-gramas*), ela ilustra bem os problemas aqui tratados. Partindo desses textos, provavelmente nos seria útil obter o seguinte conjunto de atributos: "*textual data mining*", "*data mining*", "*textual data*", "*textual collections*", "*special meaning*", "*useful information*", "*useful knowledge*", "*data*" e "*use*". Outras combinações de palavras, ou mesmo apenas palavras simples, parecem não ser interessantes para discriminar o conjunto de textos. A questão é como chegar automaticamente a um conjunto de atributos enxuto e discriminativo.

Assim, o primeiro passo é como gerar os *n-gramas*, na próxima subseção discorre-se sobre modelos de geração de *n-gramas*, ou seja, como identificá-los nas coleções de textos, codificá-los e apresentá-los. O segundo passo é como selecionar os atributos, que técnicas podem ser usadas para decidir se o atributo é ou não mantido no processo de Mineração de Textos. O terceiro passo é decidir quais palavras compostas (*n-gramas*) são estatisticamente mais significativos na coleção. Então, supondo-se que se tenham atributos unigramas e *n-gramas*, isto é, com *n* maior ou igual a dois (2), selecionam-se os atributos a serem efetivamente utilizados. E, finalmente, na última subseção desta, mostra-se o que está sendo tratado como redundância neste ponto do pré-processamento das coleções de textos.

2.1 Modelo de geração de *n-gramas*

Ferramentas que geram as combinações de palavras presentes nos textos, geralmente, descartam as *stopwords* especificadas e depois aplicam algum processo de identificação das palavras de interesse, dependendo de como se definem os *tokens* para seus analisadores léxicos. As *stopwords* e os *tokens* são definidos pelo usuário da ferramenta. Por exemplo, para a ferramenta *N-gram Statistical Package* - NSP (BANERJEE; PEDERSEN, 2003), define-se um conjunto de expressões regulares que delimitam os *tokens* de interesse; já para a ferramenta PreText (MATSUBARA et al., 2003; SOARES et al., 2008) define-se qual a língua na qual os textos estão escritos, pois ela trabalha com radicalização inflexional (conhecida por *stemming*).

Para a coleção de textos do exemplo (Tabela 1), considerando-se a geração de unigramas, bigramas e trigramas com a ferramenta PreText, têm-

se: 32 unigramas, 51 bigramas e 63 trigramas. Todos esses *n-gramas* se encontram na Tabela 2 Cada valor apresentado nas células da tabela deve ser interpretado como ***tf:(df/#d):stem***. O valor ***tf*** (*term frequence*) corresponde à freqüência total de ocorrências do atributo na coleção. O valor ***df*** (*document frequence*) corresponde ao número de documentos nos quais o atributo é presente. O valor ***#d*** corresponde ao número total de docu-

Tabela 2. *N-gramas* formados partir dos textos do exemplo.

32 unigramas	1:(1/5):market 1:(1/5):article 1:(1/5):kind 2:(2/5):knowledge 1:(1/5):busi 1:(1/5):digit 1:(1/5):applic 2:(2/5):inform 1:(1/5):media 1:(1/5):discov 1:(1/5):text 3:(3/5):process 1:(1/5):tool 1:(1/5):magazin 1:(1/5):chat 3:(3/5):transform 1:(1/5):power 1:(1/5):definit 1:(1/5):tendency 4:(4/5):collect 1:(1/5):internet 1:(1/5):work 1:(1/5):format 11:(5/5):mine 1:(1/5):subject 1:(1/5):mean 1:(1/5):talk 17:(5/5):textual 1:(1/5):manag 1:(1/5):aim 2:(2/5):special 24:(5/5):data
51 bigramas	1:(1/5):process_knowledg 1:(1/5):data_data 1:(1/5):mine_subject 1:(1/5):discov_market 1:(1/5):digit_media 1:(1/5):data_process 1:(1/5):transform_knowledg 1:(1/5):mine_power 1:(1/5):special_mean 1:(1/5):definit_data 1:(1/5):mine_work 1:(1/5):articl_internet 1:(1/5):chat_textual 1:(1/5):mine_data 1:(1/5):data_discov 1:(1/5):internet_chat 1:(1/5):data_transform 1:(1/5):data_magazin 1:(1/5):text_talk 1:(1/5):subject_text 1:(1/5):data_aim 1:(1/5):collect_digit 1:(1/5):tendenc_textual 1:(1/5):collect_format 1:(1/5):market_tendenc 1:(1/5):format_data 2:(2/5):data_inform 1:(1/5):mine_busi 1:(1/5):applic_textual 2:(2/5):process_transform 1:(1/5):mine_definit 1:(1/5):work_textual 2:(2/5):transform_data 1:(1/5):busi_process 1:(1/5):magazin_articl 2:(2/5):collect_data 1:(1/5):media_data 1:(1/5):mine_special 3:(3/5):mine_textual 1:(1/5):power_tool 1:(1/5):tool_applic 4:(4/5):textual_collect 1:(1/5):aim_special 1:(1/5):knowledg_manag 5:(3/5):data_textual 1:(1/5):kind_data 1:(1/5):mine_process 11:(5/5):data_mine 1:(1/5):special_kind 1:(1/5):talk_data 13:(5/5):textual_data
63 trigramas	1:(1/5):textual_data_discov 1:(1/5):data_mine_special 1:(1/5):collect_data_mine 1:(1/5):mine_process_transform 1:(1/5):collect_digit_media 1:(1/5):articl_internet_chat 1:(1/5):data_mine_subject 1:(1/5):power_tool_applic 1:(1/5):data_mine_process 1:(1/5):magazin_articl_internet 1:(1/5):aim_special_mean 1:(1/5):work_textual_data 1:(1/5):data_data_aim 1:(1/5):mine_work_textual 1:(1/5):data_mine_power 1:(1/5):internet_chat_textual 1:(1/5):mine_subject_text 1:(1/5):text_talk_data 1:(1/5):data_aim_special 1:(1/5):subject_text_talk 1:(1/5):media_data_mine 1:(1/5):process_knowledg_manag 1:(1/5):talk_data_mine 1:(1/5):tendenc_textual_data 1:(1/5):collect_format_data 1:(1/5):digit_media_data 1:(1/5):definit_data_data 1:(1/5):chat_textual_collect 1:(1/5):data_mine_work 1:(1/5):data_process_transform 1:(1/5):data_mine_data 1:(1/5):discov_market_tendenc 1:(1/5):mine_busi_process 1:(1/5):data_mine_definit 1:(1/5):textual_collect_format 1:(1/5):mine_power_tool 1:(1/5):data_transform_knowledg 1:(1/5):special_kind_data 2:(2/5):data_textual_data 1:(1/5):kind_data_mine 1:(1/5):textual_data_magazin 2:(2/5):transform_data_inform 1:(1/5):mine_data_mine 1:(1/5):applic_textual_data 2:(2/5):process_transform_data 1:(1/5):collect_data_transform 1:(1/5):format_data_mine 2:(2/5):textual_collect_data 1:(1/5):data_magazin_articl 1:(1/5):data_mine_busi 3:(3/5):mine_textual_data 1:(1/5):market_tendenc_textua 1:(1/5):mine_definit_data 3:(3/5):data_mine_textual 1:(1/5):busi_process_knowledg 1:(1/5):textual_data_process 3:(3/5):data_textual_collect 1:(1/5):tool_applic_textual 1:(1/5):textual_collect_digit 5:(3/5):textual_data_textual 1:(1/5):mine_special_kind 1:(1/5):data_discov_market 5:(5/5):textual_data_mine

mentos na coleção. E, finalmente *stem* corresponde ao *n-grama* gerado. Geralmente, na área de recuperação de informações, os valores extraídos como atributos dos textos, ou seja, nesse caso os ***stems***, são denominados termos, por isso a medida ***tf*** corresponde a ***term frequence***.

No entanto, em todo este trabalho, sempre denominaremos os ***stems***, ou outras formas de obtenção dessas características dos textos, como **atributos**. Há dois bons motivos para isso: o primeiro é que termo corresponde a uma denominação mais ampla, dada a palavras ou expressões com significado semântico e associado a algum domínio de conhecimento; o segundo é que em mineração de textos estamos trabalhando com modelos de aprendizado de máquina e o papel desses "termos", nessa área, é denominado **atributo**.

Ainda, observando-se os *n-gramas* apresentados na Tabela 2 é evidente que nem todas as combinações levam a termos de interesse. Logo, grande parte dessas combinações deve ser cortada do conjunto de atributos.

2.2 Seleção de atributos unigramas

O número de atributos, mesmo após um cuidadoso processo de geração dos mesmos, inclusive removendo as inflexões (aplicando um processo de *stemming*), é exageradamente grande e, não possui representatividade em cada documento da coleção, levando a representações esparsas de suas ocorrências. Desta forma, outra tarefa de especial interesse nessa etapa é a **seleção de atributos unigramas**. O método mais comumente utilizado corresponde aos cortes de Luhn (1958). Para realizar esses cortes ordenam-se as frequências de ocorrência dos termos na coleção e, após obter o gráfico dessa curva, denominada curva de Zipf (1949), escolhem-se, subjetivamente, os pontos de corte, próximos aos pontos de inflexão da curva, considerando-se que palavras com freqüência de ocorrência muito baixa ou muito alta sejam estatisticamente irrelevantes, ou seja, escolhem-se os pontos de corte inferior (freqüência mais baixa) e corte superior (freqüência mais alta), como ilustrado na Figura 1.

Também, bastante comum é o método de Salton que usa os termos cujas frequências de ocorrência nos documentos da coleção estejam no intervalo

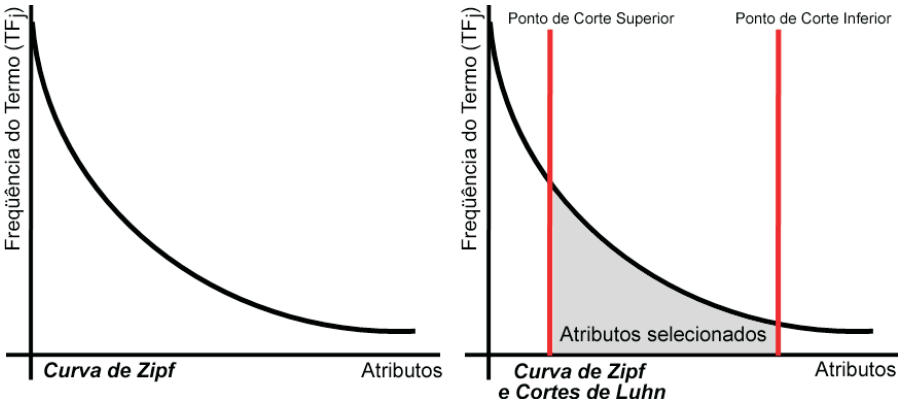


Figura 1. Cortes de Luhn sobre a curva de Zipf.

de um a dez por cento do total de documentos (SALTON et al., 1975). No trabalho de Nogueira et al. (2008) estão presentes algumas sugestões de filtros e a avaliação dos mesmos, dentre os quais alguns de representatividade da contribuição do atributo para a variância do conjunto total de atributos, também muito interessantes para unigramas. Filtros baseados em freqüência ou contribuição para a variância são mais confiáveis quando aplicados a conjuntos de atributos do tipo unigrama, pois tendem a favorecer termos com valores de frequências de ocorrências nem tão raras nem tão comuns. Experimentalmente, é fácil observar que para palavras compostas essa regra não vale, basta observar a Tabela 2. Normalmente, as palavras compostas com alta freqüência de ocorrência é que são os melhores candidatos a atributos, embora essa observação não garanta que eles sejam candidatos a termos de domínio ou colocações (MANNING; SCHÜTZE, 1999).

O melhor filtro sempre é o bom senso. Por exemplo, para os unigramas apresentados na Tabela 2, obtidos de apenas cinco documentos, se um desses filtros for aleatoriamente escolhido, podem-se perder atributos valiosos para a análise. No entanto, apenas a título de exemplo, serão escolhidos apenas os unigramas, ou todos os *n-gramas*, que aparecem em pelo menos dois dos textos da coleção, ou seja, com ***df*** maior ou igual a 2. Dessa forma, se esses unigramas forem utilizados em um agrupamento hierárquico aglomerativo, eles terão maior chance de contribuir para a for-

mação de um grupo. Com esse primeiro filtro, ***df≥2***, ficam-se com os nove unigramas mostrados na Tabela 3; e após aplicar esse filtro, pode-se gerar apenas os bigramas e trigramas ilustrados na Tabela 4.

Tabela 3. Unigramas filtrados com df > 1.

2:(2/5):special	2:(2/5):knowledg	2:(2/5):inform
3:(3/5):process	3:(3/5):transform	4:(4/5):collect
11:(5/5):mine	17:(5/5):textual	24:(5/5):data

Tabela 4. Bigramas e trigramas pós filtro dos unigramas (df>1).

19 bigramas	1:(1/5):process_knowledg 1:(1/5):data_transform 4:(4/5):textual_collect 1:(1/5):transform_knowledg 2:(2/5):process_transform 4:(4/5):collect_data 1:(1/5):mine_special 2:(2/5)a:transform_data 5:(4/5):mine_textual 1:(1/5):data_special 2:(2/5):mine_process 7:(4/5):data_textual 1:(1/5):data_data 2:(2/5):data_inform 11:(5/5):data_mine 1:(1/5):data_process 3:(2/5):mine_data 13:(5/5):textual_data 1:(1/5):special_data
24 trigramas	1:(1/5):mine_process_transform 1:(1/5):mine_data_data 3:(3/5):data_textual_data 1:(1/5):mine_special_data 1:(1/5):data_mine_special 3:(2/5):data_mine_data 1:(1/5):textual_data_process 1:(1/5):mine_process_knowledg 4:(4/5):data_textual_collect 1:(1/5):special_data_mine 2:(2/5):transform_data_inform 4:(4/5):textual_collect_data 1:(1/5):data_transform_knowledg 2:(2/5):process_transform_data 5:(4/5):mine_textual_data 1:(1/5):data_data_special 2:(2/5):data_mine_process 5:(4/5):data_mine_textual 1:(1/5):collect_data_transform 2:(2/5):mine_data_mine 5:(5/5):textual_data_mine 1:(1/5):data_process_transform 3:(3/5):collect_data_mine 7:(4/5):textual_data_textual

2.3 Seleção de atributos *n-gramas*

Para selecionar *n-gramas*, com *n*≥2, a regra de alta ou baixa freqüência nos textos não pode ser diretamente aplicada. Faz-se necessário verificar se as combinações encontradas são de fato significantes ou puramente casuais, por meio de testes estatísticos, por uma análise linguística ou terminológica, por comparação com algum vocabulário controlado, ou combinações de métodos (CONRADO et al., 2008). Qualquer teste que seja empregado estará sempre se referindo à coleção de textos disponível, logo, uma combinação de palavras (gramas) será considerada casual ou

importante na coleção de textos como um todo, e, não necessariamente para o idioma ou domínio de conhecimento considerado.

Esses *n-gramas* são candidatos a colocações, termos, expressões ou frases interessantes em uma coleção de textos e, podem ser estatisticamente indicados sem uma análise linguística rigorosa para, então, serem selecionados por especialistas do domínio (MANNING; SCHÜTZE, 1999). Esses candidatos serão automaticamente selecionados e considerados como os atributos mais relevantes no processo de Mineração de Textos.

De acordo com Banerjee e Pedersen (2003) e Manning e Schütze (1999), pode-se estabelecer hipóteses de dependência entre as combinações de *n-gramas* geradas e/ou *rankings* de dependência. Se as combinações são independentes, então o *n-grama* é considerado estatisticamente irrelevante na coleção de textos, caso contrário pode-se estabelecer seu *ranking* de relevância. Quando se obtém um *ranking* de todos os *n-gramas* considerados estatisticamente relevantes na coleção de textos é necessário decidir empiricamente um ponto de corte. Nesse caso, pode-se utilizar valores tabelados, em tabelas de distribuição de probabilidades, como os valores críticos de aceitação da hipótese de independência entre os *gramas* que formam o *n-grama*, dependendo de qual foi o estimador escolhido para o teste e estabelecendo-se níveis de significância. Ou, pode-se estabelecer um intervalo de confiança para a aceitação dos valores obtidos. Seja qual for o estimador escolhido, corte ou valor crítico utilizado, deve-se verificar se está adequado ao problema e à distribuição da freqüência observada dos dados.

Há várias possibilidades para os testes; por exemplo, a ferramenta NSP (BANERJEE; PEDERSEN, 2003) possui vários desses testes estatísticos implementados. No entanto, como se trata de um processo de mineração de textos, há que se utilizar um estimador que seja adequado a dados esparsos; pois, *n-gramas* podem ser muito raros ou muito frequentes em uma coleção, o que costuma causar a presença de frequências muito grandes e muito pequenas (inclusive próximas a zero) em vários casos de testes. Nem todo estimador tem um bom comportamento nesses casos, e, a literatura recomenda o uso de estimadores mais robustos ou mais confiáveis, como a razão do logaritmo natural da função estimada por máxima verossimilhança (MANNING; SCHÜTZE, 1999) ou estimadores que independem

da distribuição de frequências, com base na razão do produto cruzado para tabelas 2x2 (BISHOP et al., 1975).

Por exemplo, um bigrama genérico é w_1w_2, sendo w_1 a primeira palavra (grama) e w2 a segunda palavra (grama) na sua composição. A freqüência observada de w_1w_2 na coleção é f_{11}, a freqüência observada de w_1 na primeira posição em todos os bigramas obtidos da coleção é $f_{1.}$ e, a freqüência de w_2 na segunda posição entre todos os bigramas obtidos da coleção é $f_{.1}$. A freqüência de todos os bigramas na coleção é $f_{..}$. Para cada bigrama w_1w_2 tabulam-se todos esses valores como na Tabela 5.

Tabela 5. Frequências dos bigramas e suas partes.

	w_2	$!w_2$	
w_1	f_{11}	f_{12}	$f_{1.}$
$!w_1$	f_{12}	f_{22}	$f_{2.}$
	$f_{.1}$	$f_{.2}$	$f_{..}$

Para testar a hipótese de independência entre w_1 ocorrendo na primeira posição e w_2 ocorrendo na segunda posição do bigrama w_1w_2, usando o estimador Q de Yule, que é uma função de transformação da razão do produto cruzado (BISHOP et al., 1975). Deve-se então obter os valores cujos cálculos são ilustrados na Figura 2.

$$d = ((f_{..} - f_{1.}) - (f_{.1} - f_{11}))$$

$$\alpha = \frac{f_{11} * d}{(f_{1.} - f_{11}) * (f_{.1} - f_{11})}$$

$$Q_1 = (\alpha - 1)/(\alpha + 1)$$

$$v = \sqrt{\frac{1}{f_{11}} + \frac{1}{f_{1.} - f_{11}} + \frac{1}{f_{.1} - f_{11}} + \frac{1}{d}}$$

$$\sigma_{Q1} = \frac{1}{2} * (1 - Q_1^2) * v$$

$$IC_{Q1} = [Q_1 - 2 * \sigma_{Q1} + 2 * \sigma_{Q1}]$$

Figura 2. Valores alfa, *Q* e intervalo de confiança.

Para que w_1 e w_2 nessas posições sejam independentes, o valor α deve se aproximar de *1*, logo Q_1 deve se aproximar de *0*. O intervalo de confiança

da estimativa de *Q* é construído para cobrir 95% dos valores, dado que *Q* tem uma distribuição Normal (BISHOP et al., 1975). Assim, se $0 \in IC_{Q1}$ então o bigrama w_1w_2 não é candidato à colocação de domínio, termo ou frase interessante; pois esse bigrama tem uma distribuição homogênea na coleção de textos, logo, é estatisticamente insignificante.

Por exemplo, na Tabela 4 estão os bigramas e trigramas gerados após o corte de ***df>1***, para os unigramas do exemplo anterior, e a eliminação de algumas *stopwords* especialmente definidas (as corretas definições são colocadas com detalhes na seção 1.4). Para esses bigramas e trigramas, da Tabela 4, pode-se calcular os mais estatisticamente significantes e depois obter seus *rankings*. Por exemplo, para o bigrama "*data_transform*", da Tabela 4, tem-se os valores da Tabela 6, o que leva aos valores ilustrados na Figura 3. Logo, o bigrama "*data_transform*" pode ser desconsiderado do conjunto de atributos, dado que essa combinação é considerada casual segundo o teste estatístico realizado.

$$\alpha = \frac{1 \times 37}{2 \times 23} = 0.8043,\ Q_1 = \frac{\alpha - 1}{\alpha + 1} = -0.1084$$

$$v = \sqrt{\frac{1}{1} + \frac{1}{37} + \frac{1}{2} + \frac{1}{23}} = 1.2532$$

$$\sigma_Q = \frac{1}{2} * (1 - Q^2) * v = 1.2385$$

$$IC_Q = [-1.3469, 1.1301] \Rightarrow 0 \in IC_Q$$

Figura 3. Valores de alfa, Q e intervalo para o bigrama "*data-transform*"

Tabela 6. Frequências para o bigrama *data_transform* .

	transform	!transform	
data	1	23	24
!data	2	37	39
	3	60	63

Para testar a validade de um trigrama, como $w_1w_2w_3$, precisa-se de sua freqüência observada f_{123}, e, também se precisa da freqüência observada de w_1w_2, $f_{12.}$, sempre que esse bigrama é presente na primeira posição de todos os trigramas da coleção. Além disso, é necessária a freqüência observada do bigrama w_2w_3, $f_{.23}$, na segunda e terceira posições de todos

os trigramas da coleção. E, ainda, a freqüência observada de w_1, $f_{1..}$, na primeira posição de todos os trigramas da coleção e de w_3, $f_{..3}$, na terceira posição de todos os trigramas da coleção. Finalmente, precisa-se da freqüência observada de todos os trigramas da coleção $f_{...}$. Isto porque, é preciso realizar o teste de w_1 com w_2w_3 e w_1w_2 com w_3.

Assim, na Tabela 7 é ilustrada a freqüência de w_1, $f_{1..}$, na primeira posição em todos os trigramas da coleção e w_2w_3, $f_{.23}$, na segunda e terceira posições de todos os trigramas da coleção, além da freqüência de $w_1w_2w_3$, f_{123}, bem como seus complementos.

Tabela 7. Divisão do trigrama $w_1w_2w_3$ em w_1 e w_2w_3.

	w_2w_3	$!(w_2w_3)$	
w_1	f_{123}	$f_{1!(23)}$	$f_{1..}$
$!w_1$	$f_{(!1)23}$	$f_{(!1)!(23)}$	$f_{!(1)..}$
	$f_{.23}$	$f_{.!(23)}$	$f_{...}$

Para testar a hipótese de independência entre as ocorrências de w_1 na primeira posição entre todos os trigramas e w_2w_3 na segunda e terceira posições de todos os trigramas, calcula-se o estimador Q de Yule, como realizado no exemplo anterior. A seguir, na Tabela 8, os valores de freqüência de w_1w_2, $f_{12.}$, nas primeira e segunda posições de todos os trigramas e w_3, $f_{3..}$, na terceira posição de todos os trigramas são organizados. Então, calcula-se o Q de Yule e seu intervalo de confiança. O trigrama $w_1w_2w_3$ será atributo selecionado como significante na coleção se, e somente se, o valor zero estiver contido nos dois intervalos de Q de Yule calculados, ou seja, se nas duas tabelas onde o trigrama foi dividido ele for considerado não homogêneo (não casual) na coleção de textos.

Tabela 8. Divisão do trigrama $w_1w_2w_3$ em w_1w_2 e w_3.

	w_3	$!(w_3)$	
w_1w_2	f_{123}	$f_{12!(3)}$	$f_{12.}$
$!(w_1w_1)$	$f_{!(12)3}$	$f_{(!12)!(3)}$	$f_{!(12).}$
	$f_{..3}$	$f_{.!(23)}$	$f_{...}$

Por exemplo, para o trigrama “data_process_transform”, as frequências dos *n-gramas* que o compõe estão ilustradas nas Tabelas 9 e 10. A partir

Tabela 9. Trigrama *data_process_transform* dividido em *data* e *process_transform*.

	process_transform	! (process_transform)	
data	1	20	21
!(data)	1	36	37
	2	56	58

Tabela 10. Trigrama data_process_transform dividido em data_process e transform.

	transform	! transform	
data_process	1	0	1
!(data_process)	2	55	57
	3	55	58

dos valores apresentados calculam-se os valores da Figura 4. Logo, esse trigrama não é considerado importante como atributo, embora $0 \in ICQ_1$, no segundo teste foi constatada associação completa. A associação completa deve-se à freqüência zerada de *data_process* com outro unigrama na terceira posição. Para detalhes de como identificar associação completa veja (BISHOP et al., 1975).

$$\alpha = \frac{1 \times 36}{1 \times 20} = 1.8,\ Q_1 = \frac{\alpha - 1}{\alpha + 1} = 0.2857$$

$$v = \sqrt{\frac{1}{1} + \frac{1}{36} + \frac{1}{1} + \frac{1}{20}} = 1.4415$$

$$\sigma_{Q1} = \frac{1}{2} * (1 - Q_1^2) * v = 0.6619$$

α com divisor zero $\Rightarrow Q_2$ = indefinido

Figura 4. Cálculos de Q de Yule e intervalos

2.4 Atributos *n-gramas* e redundâncias

Quando se deseja trabalhar com atributos unigramas e *n-gramas*, n>1, faz-se necessário combinar as técnicas de seleção dos dois tipos de atributos. Pode-se simplesmente utilizar uma técnica para selecionar unigramas e

depois algum critério para selecionar os demais *n-gramas*. O problema é que também ao utilizar mais de um grama, novamente tem-se um grande aumento no número de atributos e talvez mais redundâncias, pois os *n-gramas* são formados pelas combinações dos unigramas apresentados e são menos frequentes que esses.

Para facilitar a ilustração do que está sendo considerado redundância separaram-se apenas os atributos do primeiro texto do exemplo da Tabela 1. Na Figura 5 são mostrados apenas os trigramas, bigramas e unigramas presentes nesse primeiro texto, que subjetivamente foram considerados úteis.

Na segunda coluna da Figura 5 estão as frequências observadas para cada *n-grama*. Os *n-gramas* foram ordenados da mais alta ordem (maior *n*) para a mais baixa (menor *n*) e pela freqüência de ocorrência dentro dessa primeira categoria. Na terceira coluna dessa tabela, o número de ocorrências foi subtraído do *n-grama* de mais alta ordem, trigrama, que contem o *n-grama* da linha. Finalmente, na quarta coluna, as frequências de cada unigrama foram subtraídas das dos bigramas, não removidos na primeira

n-gramas escolhidos subjetivamente	freqüência observada	subtrai ocorrência trigrama	subtrai ocorrência bigrama
textual data mining	1		
data mining	3	3-1=2	
textual data	3	3-1=2	
textual collections	1	1-0=1	
several formats	1	1-0=1	
several definitions	1	1-0=1	
several meaning	1	1-0=1	
special meaning	1	1-0=1	
data	7	7-1=6	6-2-2-1=1
mining	3	3-1=2	2-2=0
textual	4	4-1=3	3-2-1=0
collections	1	1-0=1	1-1=0
several	2	2-0=2	2-1-1=0
formats	1	1-0=1	1-1=0
definitions	1	1-0=1	1-1=0
aimed	1	1-0=1	1-0=1
special	1	1-0=1	1-1=0
meaning	1	1-0=1	1-1=0
use	1	1-0=1	1-0=1

Figura 5. Frequências observadas, redundâncias dos *n-gramas*.

subtração, nos quais eles ocorrem. Logo, os *n-gramas* cujas frequências foram zeradas nesse processo são descartados do conjunto de atributos; pois, suas frequências são combinações lineares das anteriores. Ou seja, esses *n-gramas* removidos representam redundâncias no conjunto de atributos inicial. Conseqüentemente, os *n-gramas* cujas frequências não foram zeradas são não redundantes e devem ser utilizados. Devido aos *n-gramas* de mais alta ordem serem menos frequentes que os de mais baixa ordem e formados pelos mesmos, opta-se por primeiro selecionar os de mais alta ordem e então eliminar as redundâncias.

3 Metodologia proposta

Assim, com base nas idéias desenvolvidas na seção anterior, é proposto um modelo de seleção de *n-gramas* significativos e não redundantes. Nesse modelo, ilustrado na Figura 6, faz-se a seleção de atributos unigramas e, então, adaptam-se algumas regras de seleção de atributos para selecionar também os *n-gramas*, n>1, e eliminar as redundâncias presentes entre eles.

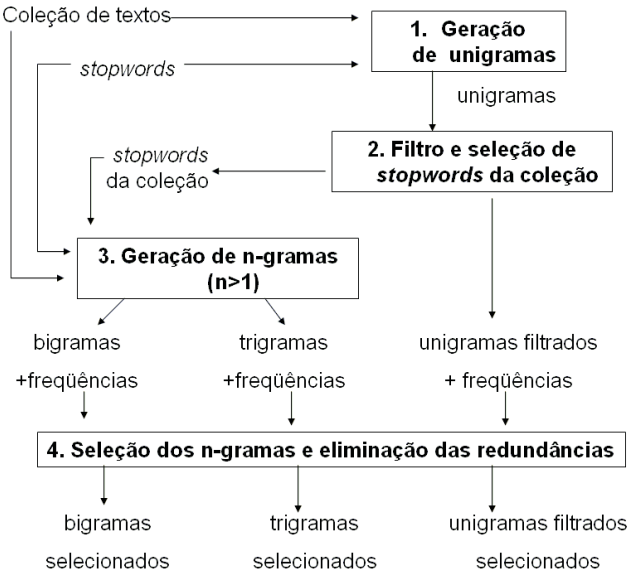


Figura 6. Ilustração do modelo de seleção de *n-gramas* significativos e não-redundantes

3.1 Passo 1 - geração de unigramas

O primeiro passo é gerar todos os possíveis unigramas da coleção de textos. Pode-se usar qualquer ferramenta que permita gerar os unigramas e contabilizar suas frequências, por exemplo, a PreText ou a NSP. Nesse passo, as entradas do processo são a coleção de textos e a lista de *stopwords* habituais. Essa lista de *stopwords* corresponde geralmente a artigos, interjeições, conjunções, preposições, etc; ou seja, palavras que se considera poder ser eliminadas da formação dos atributos. Logo, a saída deste processo são os unigramas gerados e contabilizados, pela sua freqüência de ocorrência na coleção de textos, medida **tf**.

3.2 Passo 2 - filtragem dos unigramas e seleção de *stopwords* da coleção

No segundo passo, escolhe-se um filtro para os unigramas gerados, geralmente um dos citados na seção anterior. Aplicado o filtro tem-se os unigramas filtrados e as *stopwords* da coleção. Os **unigramas filtrados** estão sendo tratados como aqueles que foram selecionados como atributos após a aplicação do filtro. Neste método, proposto para a seleção de *n-gramas*, as ***stopwords* da coleção** correspondem aos unigramas que foram descartados pelo filtro. Isso porque, vários autores consideram que *stopwords* devam ser obtidas dessa forma (SOUPICA; MINEAU, 2005).

3.3 Passo 3 - geração dos *n-gramas* com n>1

As *stopwords* da coleção precisam ser descartadas da construção dos *n-gramas* de mais alta ordem, porque podem vir a interferir na sua construção e seleção. Por exemplo, para um *n-grama* como “pastagem pesquisada melhorada Embrapa”, se as inflexões do verbo “pesquisar” tiverem sido descartadas, bem como o nome próprio “Embrapa”, o correto seria gerar apenas o bigrama “pastagem melhorada”.

Com base no exemplo utilizado na seção anterior, o trigrama “subject_ text_talk”, da Seleção de atributos unigramas, simplesmente deixa de existir após o corte dos unigramas com df<2, pois ficam como unigramas filtrados apenas os que possuem df>1. E, por exemplo, o trigrama “textu-

al_data_magazin" também deixa de existir após esse corte, mas o bigrama "textual_data" continua existindo; comparar a Seleção de atributos unigramas com a Seleção de atributos *n-gramas*. Pois, retirando-se as *stopwords* da coleção, geram-se os *n-gramas* de mais alta ordem formados apenas pelos unigramas estatisticamente mais discriminativos.

Assim, o processo de geração de *n-gramas* neste passo, recebe como entrada a coleção de textos, a lista de *stopwords* (habituais) e a lista de *stopwords* da coleção. Com isso, espera-se gerar apenas os *n-gramas*, n>1, que não contenham os unigramas descartados pelo filtro.

No exemplo anterior, especificam-se as *stopwords* da coleção como os unigramas descartados após o corte df>1 e todas as suas inflexões. Então, reaplicando a ferramenta PreText à coleção de documentos são gerados apenas os trigramas e bigramas relacionados na Seleção de atributos *n-gramas*.

Para observar um exemplo completo, totalmente fictício, supor os passos de extração dos *n-gramas* na frase da Figura 7. No primeiro retângulo, a frase está completa, nenhum tratamento lhe foi aplicado. No segundo retângulo, as *stopwords* habituais ("na", "de", "porque", "isso", "em", "menor", "e", "melhor", "para", "essa") e todas as inflexões foram removidas (aplicado um processo de *stemming* para a língua portuguesa (MATSUBARA et al., 2003)). No terceiro retângulo, algum filtro para a seleção de unigramas foi aplicado, eliminando-se os estatisticamente menos significantes na coleção de textos onde esse texto se insere. Com a seleção dos unigramas, a frase ficou curta; dado que lhe são descontadas as *stopwords* da coleção, que foram: "decidir", "usar", "impactar" e "absorca".

Consequentemente, para formar os *n-gramas* com n>1 e n<4, ter-se-iam os resultados apresentados no quarto retângulo. Faltaria, então, decidir quais desses *n-gramas* são estatisticamente significativos e não redundantes na coleção de textos em que esse texto se insere, o que corresponde ao quarto passo, detalhado a seguir; e, estando no quinto retângulo um exemplo de quais *n-gramas* poderiam ter sido escolhidos para esse texto.

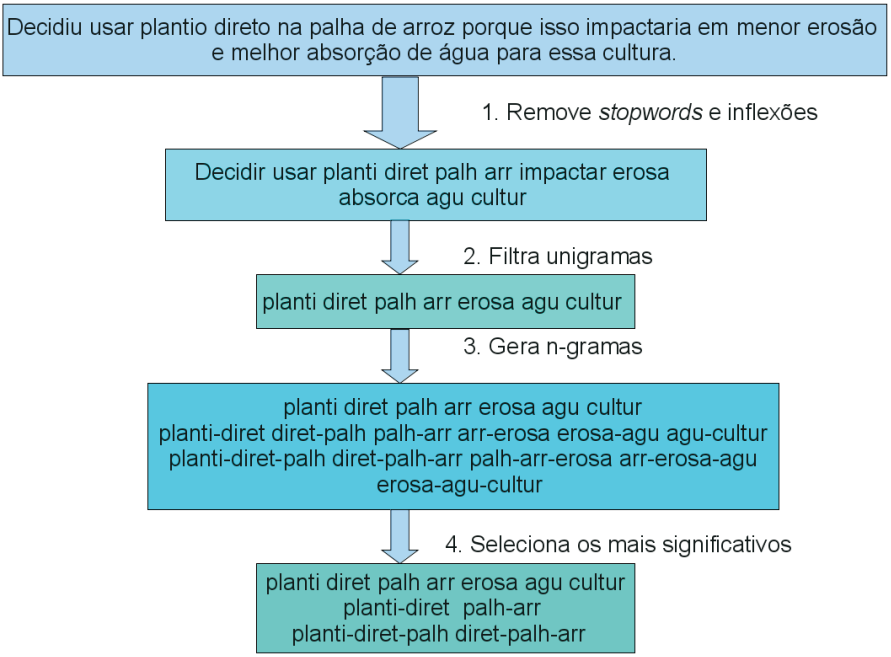


Figura 7. Ilustração dos passos 1 ao 3 e possível resultado do 4.

3.4 Passo 4 - seleção dos *n-gramas* e eliminação de redundâncias

Neste passo, os *n-gramas*, com n=3 ou 2, devem ser estatisticamente testados para serem ou não selecionados como atributos. Isto é feito, realizando-se testes de dependência entre os *n-gramas* de mais baixa ordem que os formam, de acordo com suas frequências de ocorrência na coleção de textos. De modo a simplificar a escolha dos testes de independência e sua aderência ao problema, selecionou-se o estimador Q de Yule, que não depende da distribuição dos dados, aplicado a decomposições das tabelas de frequências dos *n-gramas*, como ilustrado na seção 1.3; para detalhes sobre o estimador e decomposição das tabelas veja (BISHOP et al., 1975).

Todos os trigramas selecionados como atributos são *rankeados* de acordo com suas frequências de ocorrência na coleção de textos, isto é, pelos

valores obtidos como f_{123}. Tão logo todos os trigramas tenham sido testados, selecionados e *rankeados*, todos os bigramas e unigramas cujas frequências observadas formam os trigramas por combinações lineares, são eliminados com o Algoritmo 1. Para isso, utilizam-se as frequências observadas, notadas por O_f neste trabalho. Logo, tem-se: O_{fij} com *i,j=1,2,3* para os bigramas e O_{fi} com i=1,2,3 para os unigramas. Todos os *n-gramas* são tratados pelo algoritmo na ordem em que foram *rankeados* pelas suas frequências A base do algoritmo é que para cada trigrama, há algum(ns) bigrama(s) e unigrama(s) com freqüência maior ou igual à deles. Nesse caso, a eliminação da redundância funciona como mais um critério de seleção de atributos, deve-se salientar que, apenas os melhores candidatos a atributos entre os trigramas ficam na lista de atributos finais.

Input: *TriGramList*: lista de trigramas ordenados pelo valor de seus *ranks*
Output: *FinalAttributesList*: lista de atributos finais com freqüências atualizadas
for *todos os trigramas em TriGramList* **do**
  selecionar o trigrama melhor *rankeado*, $w_1w_2w_3$;
  procurar por w_1w_2, com $O_{f_{12}} \geq O_{f_{123}}$;
  procurar por w_2w_3, com $O_{f_{23}} \geq O_{f_{123}}$;
  procurar por w_1, com $O_{f_1} \geq O_{f_{123}}$;
  procurar por w_2, com $O_{f_2} \geq O_{f_{123}}$;
  procurar por w_3, com $O_{f_3} \geq O_{f_{123}}$;
  subtrair $O_{f_{123}}$ de todos os n-gramas acima, de cada $O_{f_{ij}}$ e O_{f_i};
  adicionar $w_1w_2w_3$ em *FinalAttributesList*;
end

Algoritmo 1. Seleção dos trigramas e eliminação das redundâncias.

Após a seleção dos trigramas e eliminação de redundâncias, o mesmo processo e idéias são aplicados aos bigramas, que não foram anteriormente eliminados. Deste modo, os candidatos a bigramas são selecionados com o teste de independência e cálculo do estimador Q de Yule; e a seguir, rankeados pela sua freqüência de ocorrência. Tão logo, todos eles tenham sido selecionados e rankeados, rankeiam-se os unigramas remanescentes do processo anterior de eliminação de redundâncias, pela sua freqüência de ocorrência. Obtém-se as frequências O_{fi}, *i=1,2* e aplica-se o Algoritmo 2.

Logo, apenas os melhores bigramas candidatos a atributos são selecionados no final do processo. Também restam apenas os unigramas que não representavam combinações lineares com os *n-gramas* de mais alta

Input: *BiGramList*: lista dos bigramas remanescentes ordenados pelos valores de seus *ranks*
Output: *FinalAttributesList*: lista de atributos finais com freqüências atualizadas
for *para todos os bigramas em BiGramList* **do**
    selecionar o bigrama melhor *rankeado*, w_1w_2 com sua freqüência remanescente $O_{f_{11}}$;
    procurar por w_1, com $O_{f_1} \geq O_{f_{11}}$;
    procurar por w_2, com $O_{f_2} \geq O_{f_{11}}$;
    subtrair $O_{f_{11}}$ de todos os bigramas acima, isto é de O_{f_1} e O_{f_2};
    adicionar w_1w_2 em *FinalAttributesList*;
end

Algoritmo 2. Seleção dos bigramas e eliminação das redundâncias.

ordem. Deste modo, novamente a eliminação da redundância serviu como um critério de seleção de atributos.

Embora o processo tenha sido aplicado apenas a trigramas, bigramas e unigramas, é fácil notar que ele pode ser expandido para qualquer *n*, desde que se apliquem as devidas partições às tabelas de contingência dos *n-gramas*.

3.5 Modelo para avaliação do método proposto

Para testar o método proposto para seleção de atributos *n-gramas* foram implementados alguns protótipos para realizar os testes, *ranking* e seleção dos atributos. Deste modo é possível obter as listas de candidatos a atributos e as listas finais resultantes do método proposto, para compará-los. A melhor comparação deveria ser uma análise subjetiva, dado que se tratam de atributos que também são candidatos a termos de domínio. Logo, seria interessante que um grupo de especialistas do domínio, preferencialmente, terminólogos, avaliassem os resultados. No entanto, qualquer que seja o tamanho da coleção de textos, para que se possa representar pelo menos parte de um domínio de conhecimento, é gerado um número muito grande de atributos. Logo, uma análise subjetiva seria uma tarefa nada trivial e muito onerosa.

Uma análise objetiva simples seria utilizar coleções de textos rotuladas e verificar o desempenho de um ou mais classificadores mediante diferentes conjuntos de atributos. Pois, o método proposto possibilita criar diferentes conjuntos de atributos, e uma forma de medir se esses diferentes conjuntos influenciam o desempenho de modelos de aprendizado de máquina

é comparando alguns classificadores construídos com esses conjuntos (MITCHEL, 1997).

Porém, mesmo com a coleção de textos rotulada, a seleção de atributos deve ser não supervisionada, ou seja, em nenhum momento deve-se incluir a informação do rótulo na seleção de atributos. Assim, constroem-se dois conjuntos de atributos para cada coleção de textos considerada:

1. **atributos rankeados**: geram-se os unigramas e os demais *n-gramas*. Filtram-se os unigramas e encontram-se as *stopwords* da coleção. E, então todos os *n-gramas* são selecionados e *rankeados* de acordo com o teste de interdependência anteriormente explicado.
2. **atributos não redundantes**: aplica-se o método proposto, todos os seus passo sequencialmente, para obter o conjunto de atributos.

O teste de seleção dos *n-gramas* é aplicado igualitariamente aos dois conjuntos de atributos, para que não se insiram outros efeitos à comparação, além da eliminação de redundâncias e delimitação das *stopwords* da coleção.

Então, uma matriz atributoxvalor é construída para cada conjunto de atributos; ou seja, uma matriz atributo x valor para o conjunto de dados com atributos apenas *rankeados* e outra matriz atributo x valor para o mesmo conjunto de dados porém removidas as *stopwords* da coleção e as redundâncias. Em cada matriz atributo x valor, as células correspondem à freqüência de ocorrência do atributo no documento e, a última coluna corresponde ao rótulo do documento.

Para validar os resultados, para cada conjunto de atributos, dois modelos de classificação são construídos, utilizando dois algoritmos de classificação amplamente conhecidos: C4.5 para árvores de decisão e *Support Vector Machines* (SVM). Ambos escolhidos devido ao fato de suportarem bem matrizes esparsas; especialmente, o SVM comporta-se bem e encontra um modelo mesmo na presença de muitos dados redundantes (JOACHIMS, 1998). Adicionalmente, para estimar a medida de acurácia dos classificadores, utiliza-se um modelo de amostragem cruzada, para que se possam obter médias e desvios padrões dessas medida, por exemplo, o modelo de 10 pastas (*10-fold cross validation*). Finalmente,

após obter as médias e desvios das medidas de acuracidade de cada classificador, essas são comparadas usando-se o estimador *t de Student*. Dessa forma, se não houver diferenças estatisticamente significantes entre essas médias, é porque os diferentes conjuntos de atributos não influenciaram o desempenho desses classificadores; logo, os diferentes conjuntos de atributos poderão ser considerados discriminativamente equivalentes.

4 Resultados e discussão

Para validar o método proposto escolheram-se três conjuntos de dados de diferentes domínios e línguas, dado que seria interessante também observar se a mudança de língua provocaria efeitos diferentes nos resultados.

O primeiro conjunto de dados é uma coleção de textos de artigos completos do Instituto Fábrica do Milênio (IFM). Esse conjunto de dados é composto por 614 documentos em português, divididos em quatro classes, com 291 documentos na classe majoritária. A segunda coleção de documentos é o conjunto de artigos *Case Based Reasoning - Inductive Logic Programming - Information Retrieval - Sonification* (CBR-ILP-IR-SON)[6] composto por 681 resumos de documentos em inglês, classificados em quatro classes e com 276 documentos na classe majoritária, tratando-se exclusivamente de resumos de artigos científicos e neste trabalho denominada por CIIS. A terceira coleção são artigos completos em inglês sobre inteligência artificial (IA), divididos em cinco categorias: agentes e multiagentes, lógica *fuzzy*, aprendizado de máquina, planejamento e robótica.

Um processo completamente aleatório foi utilizado para balancear as coleções de textos, escolhendo cinquenta documentos por classe. Exceto pela coleção IFM; nessa, três classes sofreram o processo de escolha aleatória de 50 de seus documentos, porém, uma das classes tinha apenas 43 documentos e ficou com todos os originais. Deste modo, procurou-se minimizar efeitos causados pelo desbalanceamento das classes.

[6] *Disponível em http://infoserver.lcad.icmc.usp.br/infovis2/PEx.*

4.1 Experimento e resultados

Na Tabela 11 encontram-se o número de documentos em cada coleção, o número total de unigramas antes e depois do filtro. Utilizou-se o filtro TC, tomando-se apenas 10% dos unigramas rankeados, o que, na verdade, era um resultado de experimentos prévios (NOGUEIRA et al., 2008). Na mesma tabela, as duas últimas colunas referem-se ao número de bigramas e trigramas após o filtro dos unigramas e a remoção das *stopwords* da coleção. Assim, as três últimas colunas da Tabela 11 correspondem à cardinalidade dos primeiros conjuntos obtidos de candidatos a atributos.

Tabela 11. Número de documentos, unigramas, bigramas e trigramas nas coleções.

Coleção de textos	#docs	#unigramas	#unigramas pós-filtro	#bigramas	#trigramas
IA	250	42753	4276	87297	30326
CIIS	200	10810	1081	17940	17944
IFM	193	20795	2080	82325	84200

Para encontrar o conjunto de *n-gramas* que é estatisticamente significativo na coleção de textos, aplicam-se os testes de interdependência e reduz-se os conjuntos de atributos às cardinalidades mostradas na Tabela 12. Note que, para a coleção de IA o percentual de bigramas caiu para 61%, para a CIIS esse número caiu para 23% e para a IFM caiu para 59%; e, embora para as coleções de IA e do IFM o número de trigramas tenha sido pouco reduzido (caiu para 99% e 98%), para a coleção CIIS a queda foi a 16%. A seguir, removendo-se as redundâncias que ainda sobraram entre esses atributos, tem-se as cardinalidades da Tabela 13. Note que, novamente, observam-se quedas no percentual dos *n-gramas*, os bigramas foram para 49% na coleção IA, 32% na CIIS e 14% na do IFM; enquanto os trigramas

Tabela 12. Cardinalidades dos conjuntos de atributos ainda com redundâncias.

Coleções de textos	#unigramas	R#bigramasR	#trigramas R
IA	4276	53595	30060
CIIS	1081	4284	2811
IFM	2080	48775	82354

Tabela 13. Cardinalidades dos conjuntos de atributos removidas as redundâncias.

Coleções de textos	#unigramas NR	#bigramas NR	#trigramas NR
IA	4264	26270	24303
CIIS	1065	1379	1831
IFM	1984	7043	42741

caíram um pouco menos na de IA (81%), 65% na CIIS e 51% na do IFM. Deste modo, para cada coleção ficou-se com dois conjuntos de atributos: só rankeados (R) e não redundantes (NR).

Para cada coleção e conjunto de atributos, os classificadores C4.5 e SVM foram aplicados e suas medidas de acuracidade foram estimadas usando validação *10-fold cross*. É muito importante ressaltar que os classificadores não estão sendo comparados; o foco é observar o comportamento de mais de um classificador com diferentes conjuntos de atributos.

Pelos resultados apresentados nas Tabelas 14 e 15, nota-se que tanto para o grupo de atributos rankeados (R) como para o não redundantes (NR) as médias das acuracidades foram estatisticamente iguais. Na última coluna das tabelas encontram-se os resultados do teste *t de Student* para a hipótese de igualdade das médias das acuracidades,

Tabela 14. Resultados da aplicação do C4.5 aos dados com diferentes conjuntos de atributos.

Coleções de textos	Rankeados acc±se	Não Redund. acc±se	Teste t, p_value
IA	90±5.42	89.6±7.11	0.3838
CIIS	87±9.19	87±9.19	0.3880
IFM	70.45±8.28	75.63±7.88	0.1385

Tabela 15. Resultados da aplicação do SVM aos dados com diferentes conjuntos de atributos.

Coleções de textos	Rankeados acc±se	Não Redund. acc±se	Teste t, p_value
IA	79.6±8.1	82±7.12	0.2968
CIIS	90±8.82	90.5±6.85	0.3837
IFM	59.97±10.24	68.84±11.74	0.0835

isto é, H_0: acc_r=acc_{nr}. Tem-se que a média de accr e o desvio padrão ser correspondem respectivamente ao valor médio e ao desvio padrão das acuracidades mediante o *10-fold cross validation*, para o conjunto de *rankeados*. E, que accnr e senr correspondem respectivamente ao valor médio e ao desvio padrão de cada acurácia, mediante o *10-fold cross validation*, para o conjunto de não redundantes. A Equação 1 foi utilizada na obtenção de cada valor de t e então calculada a probabilidade acumulada desse valor sob uma distribuição *t-Student*. Se a probabilidade acumulada ficar no intervalo [0.025,0.975], a hipótese de igualdade das acuracidades é aceita. Esse intervalo corresponde a valores com nove graus de liberdade. Logo, o valor de *p-value* na última coluna corresponde ao complemento dessa probabilidade em relação a um. Como todos os valores obtidos são maiores que 0.025 então a hipótese de igualdade é aceita em todos os casos.

$$t_{value} = \frac{\overline{acc_r} - \overline{acc_{nr}}}{\frac{(se)_r^2}{10} + \frac{(se)nr^2}{10}}$$

Equação 1. Teste t de Student.

Num primeiro momento, focando apenas na acuracidade de cada classificador, parece que retirar a redundância foi melhor, especialmente devido ao intervalo de valores dos desvios padrões. Mas, ao aplicar o teste t vê-se que estatisticamente as acuracidades são iguais, a um nível de significância de 95%. Deste modo, podemos observar que a remoção da redundância não sacrificou a efetividade da categorização, como esperado, ou seja, os diferentes conjuntos de atributos não afetaram os resultados dos classificadores e podem ser considerados equivalentes do ponto de vista discriminativo.

4.2 Discussão

Como o método proposto não depende do conhecimento prévio de rótulos, pode ser utilizado em aprendizado supervisionado ou não supervisionado. No entanto, para aprendizado supervisionado seria interessante trocar o processo de seleção dos unigramas, no primeiro filtro do método. Para ter uma melhor seleção dos unigramas, poder-se-ia acrescentar o conhecimento dos rótulos.

O método proposto depende somente dos testes estatísticos sobre uma coleção desordenada de palavras (ou radicais, ou lemas, etc) *n-gramas* em lugar de caracteres *n-gramas* contíguos, como no método de Dejun et al. (2004). Além disso, Dejun et al. (2004) acreditavam que seus métodos eram independentes de língua, e o método aqui proposto o é para a remoção de redundâncias. Deve-se lembrar que a identificação dos *n-gramas* depende da língua, após esse processo, a remoção de redundâncias é independente. Como evidência dessa afirmação pode-se considerar as coleções de textos utilizadas no experimento, pois uma está em português e as duas outras em inglês.

Além disso, o método proposto define e utiliza *stopwords* da coleção de textos para remover combinações indesejáveis de *n-gramas*. Essa remoção de *stopwords* tanto funciona como um filtro de atributos para que não ocorram *n-gramas* formados pelos unigramas cortados, quanto substitui janelas de formação de *n-gramas*. Por exemplo, no trabalho de Zhang e Zhu (2007) há uma proposta de estimação de tamanhos de janelas entre os gramas, de modo que sejam eliminadas algumas combinações indesejáveis. Com a remoção das *stopwords* da coleção acredita-se que o método proposto se aproxima das idéias de Zhang e Zhu (2007), causando uma melhora da informação semântica similar à estimativa de janelas, porém em um processo muito mais simples.

5 Conclusões

Um método para resolver o problema de automaticamente selecionar bons atributos *n-gramas* não redundantes é proposto. O método baseia-se no pressuposto que a redundância é sempre presente quando se usam *n-gramas*, n>1, devido à sua natureza estocástica.

O método proposto alcançou seus objetivos, selecionando bons candidatos a atributos *n-gramas* e eliminando as redundâncias apresentadas. Nos experimentos foram obtidas boas estimativas de acuracidade, o que evidencia uma boa qualidade dos atributos. Além disso, a efetividade dos classificadores não foi diminuída após a remoção das redundâncias entre os atributos, o que indica que apenas as redundâncias foram realmente eliminadas.

Além disso, o método pode ser utilizado simplesmente como um método de seleção de atributos, pois reduziu drasticamente as cardinalidades dos conjuntos de atributos selecionados, o que por si só é um resultado bastante interessante para a mineração de textos.

Como trabalho futuro, é necessário expandir os algoritmos e implementações para n qualquer, dado que o modelo apresentado limitou-se a n<4.

6 Referências

BANERJEE, S.; PEDERSEN, T. The design, implementation, and use of the Ngram Statistics Package. In: CONFERENCE ON INTELLIGENT TEXT PROCESSING AND COMPUTATIONAL LINGUISTICS, 4., 2003, Mexico. **Proceedings**... Berlin: Springer-Verlag, 2003. p. 370-381. (Lecture Notes in computer science, 2588). CICLing. Doi: 10.1007/3-540-36456-0_38.

BISHOP, Y. M. M.; FIENBERG, S. E.; HOLLAND, P. W. **Discrete multivariate analysis**: theory and practice. London: MIT Press, 1975. 557 p.

CONRADO, M. S.; MOURA, M. F.; MARCACINI, R. M.; REZENDE, S. O. **Avaliando diferentes formas de geração de termos a partir de coleções textuais**. São Carlos, SP: ICMC, USP. 2009. (Relatório técnico, 334). Disponível em: <http://www.icmc.usp.br/~biblio/BIBLIOTECA/rel_tec/RT_334.pdf>. Acesso em: 12 nov. 2010.

DEJUN, X.; MAOSONG, S.; HIGH-DEGREE, R. Overlapped character bigrams into trigrams for dimensionality reduction in chinese text categorization. In: CONFERENCE ON INTELLIGENT TEXT PROCESSING AND COMPUTATIONAL LINGUISTICS, 5., 2004, Seoul. **Proccedings**…Heidelberg: Springer-Verlag, 2004, p. 584-595. CICLing.

JOACHIMS T. Text categorization with suport vector machines: learning with many relevant features. In: EUROPEAN CONFERENCE ON MACHINE LEARNING, 10., 1988. **Proccedings**… Heidelberg: Springer-Verlag, 1998, p. 137-142. ECML.

LUHN H. P. The automatic creation of literature abstracts. **Journal of Research and Development**, v. 2, n. 2, p. 159-165, 1958.

MANNING, C. D.; SCHÜTZE H. **Foundations of statistical natural language processing**, Cambridge: MIT Press, 1999. 620 p.

MATSUBARA, E. T.; MARTINS, C. A.; MONARD, M. C. **Pre-TexT**: uma ferramenta para pré-processamento de textos utilizando a abordagem bag-of-words. ICMC, USP, 2003. Technical report, 209. Disponível em: <ftp://ftp.icmc.usp.br/pub/BIBLIOTECA/rel_tec/RT_209.pdf>. Acesso em:

MOURA, M. F.; REZENDE, S. O. Choosing a hierarchical cluster labelling method for a specific domain document collection. In: ENCONTRO PORTUGUÊS DE INTELI-GÊNCIA ARTIFICIAL, 13., 2007, Guimarães. **New triends in artificial intelligence**: Guimarães: Associação Portuguesa para Inteligência Artificial, 2007. p. 812-823.

MOURA, M. F.; NOGUEIRA, B. M.; CONRADO, M. S.; SANTOS, F. F.; Rezende, S. O. Making Good Choices of Non-Redundant N-gram Words. In: INTERNATIONAL WORKSHOP ON DATA MINING AND ARTIFICIAL INTELLIGENCE, 1.; INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION TECHNOLOGY, 11., 2008, Khulna, Bangladesh. Khulna, Bangladesh: IEEE Faculty of Electrical and Electronic Engineering and Khulna University of Engineering and Technology, 2008a. v. 1. p. 64-71.

MOURA, M. F.; MARCACINI, R. M.; NOGUEIRA, B. M.; CONRADO, M. S.; REZENDE, S. O. A proposal for building domain topic taxonomies. In: WORKSHOP ON WEB AND TEXT INTELLIGENCE, 1.; SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL, 19., 2008, Salvador. **Proceedings**... São Carlos, SP : USP, ICMC, 2008b. v. 1, p. 83-84.

MIAO Y.; KESELJ, V.; MILIOS, E. E. Document clustering using character N-grams: a comparative evaluation with term-based and word-based clustering. In: ACM INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, 14., 2005. **Proccedings**… New York: ACM. 357-358. doi>10.1145/1099554.1099665.

MITCHEL, T. **Machine learning**. Singapore: McGraw-Hill. 1997.

NOGUEIRA, B. M.; MOURA, M. F.; CONRADO, M. S.; ROSSI, R. G.; MARCACINI, R. M.; REZENDE, S. O. Winning some of the document preprocessing challenges in a text mining process. In: WORKSHOP EM ALGORITMOS E APLICAÇÕES DE MINERAÇÃO DE DADOS, 4.; SIMPÓSIO BRASILEIRO DE BANCO DE DADOS, 23, 2008. **Anais**... Porto Alegre: SBC, 2008. p. 10-18.

SALTON, G.; YANG, C. S.; YU, C. T. A theory of term importance in automatic text analysis. **Journal of the American Association Science**, v. 1, n. 26, p.33-44, 1975.

SOARES, M. V. B.; PRATI, R. C.; MONARD, M. C. **PreTexT II**: descrição da reestruturacão da ferramenta de pré-processamento de textos. São Carlos, SP: ICMC/USP 2008. Relatório técnico, n. 333. Disponível em: <http://www.icmc.usp.br/~biblio/BIBLIOTECA/rel_tec/RT_333.pdf>. Acesso em: 12 nov. 2010.

SOUPICA ,P.; MINEAU, G. W. Beyond TFIDF Weighting for Text Categorization in the Vector Space Model. In: INTERNATIONAL JOINT CONFERENCES ON ARTIFICIAL INTELIGENCE,19., 2005, Edinburgh. **Proceedings**… San Francisco: Morgan Kaufmann Publishers, 2005. p. 1130-1135. IJCAI'05. Disponível em:< http://ijcai.org/papers/0304.pdf>. Acesso em: 12 nov. 2010

WEISS, S. M.; INDURKHYA, N.; ZHANG, T.; DAMERAU, F. J. **Text mining** - predictive methods for analizing unstructured information. New York: Springer, 2005. 236 p.

ZHANG, X.; ZHU X. A new type of feature - loose N-Gram feature in text categorization. In: IBERIAN CONFERENCE ON PATTERN RECOGNITION AND IMAGE ANALYSIS, 3., 2007. **Proceedings**… Berlin: Springer-Verlag, 2007. p. 378-385. (Lecture notes in computer science, v. 4477). Doi: 10.1007/978-3-540-72847-4_49.

ZIPF, G. K. **Human behavior and the principle of least effort**: an introduction to human ecology. Cambridge: Addison-Wesley Press, 1949. 573 p.

Informática Agropecuária

Ministério da
Agricultura, Pecuária
e Abastecimento

CGPE 9158