Documentos

Abril, 2007 67

ISSN 1677-9274

Estudos sobre o Funcionamento dos Softwares TMSK e RIKTEXT para Classificação de Textos e Indução de Regras

Embrapa

ISSN 1677-9274
Abril, 2007

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

Documentos 67

Estudos sobre o Funcionamento dos Softwares TMSK e RIKTEXT para Classificação de Textos e Indução de Regras

Luiz Manoel Silva Cunha
Silvia Maria Fonseca Silveira Massruhá
Leandro Henrique Mendonça de Oliveira

Campinas, SP
2007

Embrapa Informática Agropecuária
Área de Comunicação e Negócios (ACN)
Av. André Tosello, 209
Cidade Universitária "Zeferino Vaz" – Barão Geraldo
Caixa Postal 6041
13083-970 – Campinas, SP
Telefone (19) 3789-5743 – Fax (19) 3289-9594
URL: http://www.cnptia.embrapa.br
e-mail: sac@cnptia.embrapa.br

Supervisor editorial: *Ivanilde Dispato*
Normalização bibliográfica: *Marcia Izabel Fugisawa Souza*
Editoração eletrônica: *Área de Comunicação e Negócios (ACN)*

1ª. edição on-line - 2007

Cunha, Luiz Manoel Silva.
Estudos sobre o funcionamento dos softwares TMSK e RIKTEXT para classificação de textos e indução de regras / Luiz Manoel Silva Cunha, Silvia Maria Fonseca Silveira Massruhá, Leandro Henrique Mendonça de Oliveira. – Campinas : Embrapa Informática Agropecuária, 2007.
27 p. : il. – (Documentos / Embrapa Informática Agropecuária ; 67).

ISSN 1677-9274

1. Mineração de textos. 2. Software TMSK. 3. Software RIKTEXT. I.Massruhá, Silvia Maria Fonseca Silveira. II. Oliveira, Leandro Henrique Mendoça. III. Título. IV. Série.

CDD – 006.3 (21st. Ed.)

Autores

Luiz Manoel Silva Cunha
M.Sc. em Ciência da Computação e Matemática Computacional, Analista da Embrapa Informática Agropecuária, Caixa 6041, Barão Geraldo
13083-970 - Campinas, SP.
Telefone (19) 3789-5748
e-mail: luizm@cnptia.embrapa.br

Silvia Maria Fonseca Silveira Massruhá
Dra. em Computação Aplicada, Pesquisadora da Embrapa Informática Agropecuária, Caixa 6041, Barão Geraldo
13083-970 - Campinas, SP.
Telefone (19) 3789-5814
e-mail: silvia@cnptia.embrapa.br

Leandro Henrique Mendonça de Oliveira
M. Sc. em Ciência da Computação e Matemática Computacional, Analista da Embrapa Informática Agropecuária, Caixa 6041, Barão Geraldo
13083-970 - Campinas, SP.
Telefone (19) 3789-5814
e-mail: leandro@cnptia.embrapa.br

Apresentação

A Agência de Informação Embrapa, ou Agência, é um projeto que vem sendo executado visando o estabelecimento de um portal *web* composto por várias Agências de produtos, organizando informações técnicas relevantes das várias cadeias produtivas do agronegócio brasileiro. Os resultados visam atender produtores rurais, extensionistas, pesquisadores, técnicos, professores, estudantes, entre outras classes.

Visando contribuir para a melhoria do processo de classificação de textos na Agência, foi proposto o projeto Incorporação de Ferramentas Inteligentes na Agência de Informação Embrapa. Esse projeto tem como objetivo evoluir e incorporar novas ferramentas de apoio à metodologia de estruturação das Agências visando reuso de informações bem como a incorporação de outros serviços que facilitem a transferência de tecnologia e conhecimento.

Para alcançar um dos objetivos do projeto acima citado, várias ações encontram-se em desenvolvimento em busca de seleção de ferramentas inteligentes para as tarefas de seleção, classificação de documentos e qualificação de textos.

Este trabalho apresenta os resultados parciais dos estudos sobre o funcionamento dos softwares TMSK e RIKTEXT, nos processos de classificação de documentos e indução de regras, utilizando uma amostra da base de dados da Agência de Informação Feijão (http://www.agencia.cnptia.embrapa.br/Agencia4/AG01/Abertura.html). Os resultados obtidos demonstraram que ambos os *softwares* possuem bons recursos e são candidatos a serem incorporados na plataforma tecnológica da Agência.

Espera-se que este documento seja um instrumento introdutório para o entendimento dos processos estudados e possa subsidiar outros estudos que estão por vir, no escopo do projeto Incorporação de Ferramentas Inteligentes na Agência de Informação Embrapa.

Eduardo Delgado Assad
Chefe-Geral

Sumário

Estudos sobre o Funcionamento dos Softwares TMSK e RIKTEXT para Classificação de Textos e Indução de Regras

Luiz Manoel Silva Cunha
Silvia Maria Fonseca Silveira Massruhá
Leandro Henrique Mendonça de Oliveira

Introdução

A Empresa Brasileira de Pesquisa Agropecuária - Embrapa ao longo de sua existência, gerou um grande acervo documental. Para melhor disponibilizar o conteúdo desse acervo, vários projetos visando a organização, estruturação, armazenamento e recuperação de informação foram propostos e executados.

A metodologia adotada na organização da informação da Agência é um diferencial em relação as outras metodologias utilizadas em outros projetos desenvolvidos na Embrapa. Ela permite que o conhecimento da cadeia produtiva seja armazenada numa forma hierárquica, chamada de árvore do conhecimento (Moura, 2004). Os primeiros nós da árvore armazenam os conhecimentos mais genéricos, enquanto os nós inferiores armazenam os conhecimentos mais específicos. Esses conhecimentos podem estar em vários formatos (por exemplo, texto, vídeo) e podem ser oriundos de várias fontes de informações.

As etapas de seleção e de classificação de recursos de informação são importantes para garantir a boa qualidade dos conhecimentos a serem disponibilizados pelas Agências. Segundo Moura (2004), na seleção, busca-se a obtenção de recursos que correspondem à informação bibliográfica existente a ser referenciada na Agência, ou seja as referências a outras obras que completam a informação, e também àquela utilizada para auxiliar a construção/utilização de seus conteúdos. Todo esse trabalho é realizado ou pelos especialistas em informação ou por especialistas do domínio da Agência.

O processo de classificação de recursos de informação identifica categorias pré-definidas de assuntos para novos documentos, a partir de um conjunto de documentos classificados previamente (Weis et al., 2005). Esse processo apresenta duas premissas básicas:

- as classes devem existir previamente - devem ser conhecidos o número de classes e sua identificação (significado);
- deve haver conhecimento sobre as classes ou sobre elementos que permitam decidir onde alocar novos elementos.

Para alguns pesquisadores da área de Mineração de Textos, os processos de classificação e de categorização de texto (Medeiros, 2004) executam a mesma função, ou seja, uma tarefa que decide se um documento pertence a uma classe, previamente estabelecida. Nesse trabalho foi escolhido o termo classificação por ser o termo mais usual, segundo a bibliografia consultada.

O processo de classificação ocorre após a fase de catalogação dos recursos. Nesta fase, uma relação de metadados (Pierre, 2002) é vinculada aos recursos entre os quais estão categoria e palavra-chave. Uma vez encerrada esta etapa, o documento está apto a ser classificado em um dos três eixos, são eles: árvore do conhecimento, palavra-chave e categoria (Moura, 2004).

Por não serem tarefas triviais, a seleção e a classificação de textos, no contexto de Mineração de Texto, tem despertado interesse cada vez maior na criação de *softwares* para facilitar a seleção e a classificação de textos e tornando essas ações cada vez mais automatizadas.

A Mineração de Texto pode ser definida como um processo para extração de padrões ou conhecimentos interessantes e desconhecidos em documentos textuais. Esse processo pode ser aplicado em seleção, classificação e agrupamento de documento (Rezende, 2003).

Para alcance de um dos objetivos do projeto Incorporação de Ferramentas Inteligentes na Agência de Informação Embrapa (Massruhá, 2005), foi proposto entre várias ações, estudos de *softwares* para obtenção de aglomerados, de categorias, visualização e identificação de associações entre dados textuais. Assim, combinando-se os resultados obtidos através dos *softwares*, com o processo manual de classificação de documentos em uso na Agência, pretende-se estabelecer um novo processo semi-automatizado que configure maior produtividade e maior confiabilidade de resultados.

O Text-Miner Software Kit (TMSK) é um pacote de *software* para mineração preditiva de textos (Indurkhya, 2004). Este possui funcionalidades

para pré-processar documentos de textos no formato XML e provê implementações para as seguintes tarefas:

- pré-processamento: tokenização, *stemming*, criação de dicionário e detecção de fim de sentença;
- predição: classificadores baseados em regras de decisão, predição usando Naïve Bayes e modelos lineares;
- recuperação de informação: listas invertidas (termos que apontam para os documentos);
- *clustering*: agrupamento de documentos utilizando o algoritmo *k*-Means (Pichiliani, 2006);
- extração de informações: identificação de entidades nomeadas (NER).

O RIKTEXT (Rule Induction Kit for Text) é um *sofware* para classificação de documentos baseado em regras de decisão. O objetivo é determinar o melhor conjunto de regras para a predição e a classificação, onde o melhor é o menor número de regras com o erro mínimo. Os dados para este classificador devem estar na forma de tabela, onde cada linha corresponde a um documento e cada coluna corresponde a um termo do dicionário. Cada célula da tabela é preenchida com valores booleanos indicando a presença ou a ausência do termo no documento ou a freqüência linear do termo (número de vezes que o termo aparece no documento). O RIKTEXT complementa o TMSK, disponibilizando métodos para construção e uso de regras para classificação de documentos.

Ambos os *softwares* foram escritos na linguagem de programação Java e podem ser executados nos sistemas operacionais MS DOS, Windons e Linux. A interação humano-computador de ambos é via linha de comando.

Para exercitar os processos de classificação de documentos e indução de regras de predição, utilizando o TMSK e o RIKTEXT, 180 documentos (registros) foram extraídos da base de dados da Agência de Informação Feijão e utilizados.

Os resultados obtidos demonstraram que ambos os *softwares* possuem bons recursos para classificação de documentos e para indução de regras. Isso, os credenciam para uma futura incorporação à plataforma tecnológica da Agência.Outro aspecto importante é que, tanto esses *softwares* quanto a Agência utilizam a linguagem de programação JAVA.

[1] http://www.agencia.cnptia.embrapa.br/Agencia4/AG01/Abertura.html

Esse trabalho, apresenta alguns conceitos básicos visando melhor entendimento dos processos de classificação de documentos e de indução de regras. De forma simplificada, são apresentadas as fases que devem ser executadas em ambos os processos. Um experimento utilizando os *softwares* TMSK e RIKTEXT foi desenvolvido e os resultados obtidos são apresentados e discutidos. Finalizando, encontram-se as considerações finais.

Conceitos Básicos

Nessa seção, são apresentados conceitos importantes aplicados aos processos já citados (Silva, 2005; Moura, 2004; Rezende, 2003):

Texto - é uma coleção de documentos não estruturados sem requerimentos específicos para organizá-los.

Stopwords - são palavras que não possuem conteúdo semântico significativo no contexto, sendo consideradas não relevantes para o processo de análise. Elas podem ser conjunções, artigos, verbos, auxiliares, nomes próprios, etc.

Palavras-chave - são palavras utilizadas para identificar um objeto. Neste caso, os objetos são documentos armazenados numa base de dados.

Stem - em muitos casos, variantes morfológicas das palavras têm o mesmo significado semântico, e podem ser consideradas como equivalentes dentro da proposta das aplicações de Recuperação de Informações. Assim, os termos de um documento podem ser representados pelo seu radical ou *stem*.

Stemming - o processo de *stemming* consiste na eliminação de sufixos que identificam plurais, gênero ou formas verbais, transformando termos equivalentes em uma única representação. Isso reduz o tamanho do dicionário, isto é, o número de termos distintos que representam o conjunto de documentos é muito menor, resultando em redução no espaço de armazenamento necessário e no tempo de processamento. Sendo assim, os algoritmos de *stemming* são dependentes da língua para a qual estão sendo desenvolvidos.

Stem Dictionary - local onde são armazenados os radicais das palavras extraídas por um *stemmer*.

Dicionário - local onde são armazenadas as palavras que têm relevância nos documentos pela freqüência de aparição nos textos.

***Sparse Vectors* -** trata-se de um formato de arquivo utilizado pelo *software* TSMK. Eles armazenam um conjunto de documentos convertido, no formato

de planilha, onde cada coluna corresponde a uma palavra do dicionário. Para cada palavra, é registrado a freqüência em que ela aparece no texto. As palavras que apresentarem freqüência igual a zero, não devem ser registradas no vector. Isto levará a um melhor uso do *Sparse Vectors*.

Regra - relacionamento se (X) então (Y) onde X e Y são conjunto de itens ($X \cap Y = \emptyset$)**.**

Classe - reúne num mesmo espaço objetos com atributos semelhantes de forma que eles possam ser recuperadas utilizando poucas palavras.

Processos de Classificação de Indução de Regras

Segundo Weiss et al. (2005) e Oliveira & Mendonça (2004) o processo de classificação de documentos passa pelas fases de: a) seleção de documentos; b) pré-processamento de documentos; c) desenvolvimento de classificadores; e d) classificação de documentos.

Inicialmente, os documentos são selecionados de uma base de dados utilizando um *softwar*e extrator ou a Linguagem de Consulta Estruturada (SQL) e, posteriormente, são armazenados em um arquivo. O passo seguinte é transformar esses documentos num formato que possa ser entendido pelos *softwares* de classificação e de indução de regras.

Normalmente, essa transformação faz uso da análise léxica, conversão de caracteres, remoção de *stopwords*, normalizações morfológicas, redução de dimensionalidade e outros métodos aplicáveis. O resultado é expressado em forma de uma planilha.

Durante o desenvolvimento de um classificador recomenda-se utilizar o algoritmo, que melhor se adapta ao problema, sobre dois terços dos documentos já qualificados. O um terço restante, aplica-se na etapa de testes do classificador. Nessa fase, busca-se identificar características identificadas durante seu desenvolvimento que, agrupadas formam as classes. Se os documentos possuem essas características, eles são inseridos nessas classes.

A Indução de Regras, ou Rule Induction, se refere à detecção de tendências dentro de grupos de dados, ou de “regras” sobre o dado. As regras são, então, apresentadas aos usuários como uma lista “não encomendada” (Coutinho, 2003).

Cada regra gerada cobre um subconjunto de ocorrências que pertence a uma classe específica. As regras podem ser ordenadas e não ordenadas.

Ao contrário dos complexos modelos numéricos, as regras são simples e altamente preditivas. O objetivo é determinar o menor e o melhor conjunto de regras para classificação e predição, com o erro próximo do mínimo (Rezende, 2003).

A Fig. 1, apresenta uma visão geral dos processos comentados e como eles estão relacionados. Na próxima seção, é apresentado um exemplo de classificação de documentos utilizando os classificadores Naive Bayes e Linear e Indução de Regras através dos *softwares* TMSK e o RIKTEXT.

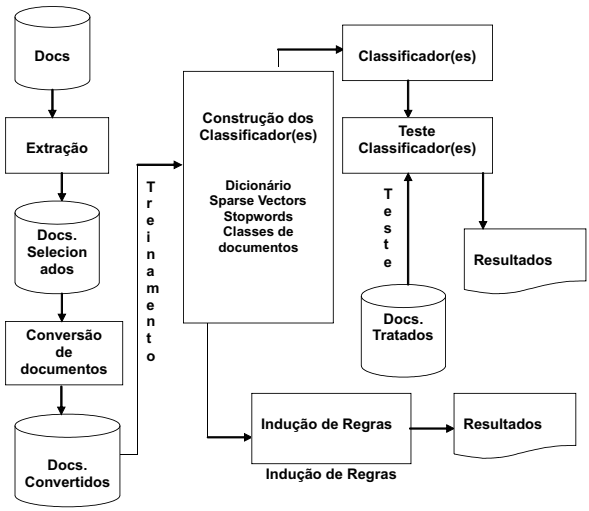

Fig. 1. Processos de classificação de documento e de indução de regras.

Experimentando o TMSK e o RIKTEXT

Utilizando o *software* DbVisualizer sobre a base de dados da Agência Feijão, aplicou-se um filtro para seleção e extração de registros contendo os dados titulo, descrição e categoria dos documentos.

Os registros extraídos foram exportados dando origem a um arquivo no formato American Standard Code for Information Interchange (ASCII). Esse foi convertido para o formato eXtensible Markup Language (XML) para manipulação pelo TMSK. Na conversão foi utilizado um programa desenvolvido na linguagem de programação Perl.

O arquivo, no formato XML, foi composto de 180 registros e dividido em dois outros arquivos, treinamento e teste. O arquivo treinamento, com

120 registros, para construção/treinamento dos classificadores enquanto que o arquivo teste, com 60 registros disjuntos, para verificar o grau de eficiência dos classificadores.

Um arquivo de *stopwords* foi estabelecido para auxiliar o processo de criação do dicionário. Esse foi organizado com rigor para não prejudicar a geração dos *Sparce Vectors.* A seguir um exemplo parcial do arquivo de *stopwords* utilizado.

a	caminho	dois	ou	fará	mais	que
as	certamente	do	um	faz	onde	para
	.....					
	.....					

O TMSK possui um arquivo de configuração de parâmetros chamado *tmsk.properties*. Ele armazena informações que são utilizadas por várias rotinas desse *software*. Durante o desenvolvimento do experimento, os parâmetros relacionados na Tabela 1, tiveram seus valores iniciais modificados.

Tabela 1. Parâmetros/descrição e respectivos valores iniciais.

Parâmetro/descrição	*Valor*
Doctag - tag XML para identificação dos documentos	doc
Bodytags – tag XML para indicar as partes dos documentos a serem analisadas	titulo e descrição
Labeltag – tag XML que indica onde está a categoria/classe do documento	topics
infile – nome do arquivo de entrada no fomato XML	feijao_trein.xml
Dictionary – nome do dicionário gerado e utilizado pela rotina *vectorize*	feijao.dic
stopwords – nome do arquivo de stopwords a ser utilizado na geração do dicionário	stopwords.wds
minimum-frequency - freqüência mínima de aparição das palavras a serem incluídas no dicionário	2
vectorfile – nome do arquivo dos Sparse vectors	feijao_trein.vec
indexfile - nome do arquivo de índices invertidos gerados pela rotina	feijao.idx
vectorize	feijao_trein.vec

Após a conversão dos dados, da criação do arquivo de *stopwords e* da configuração dos parâmetros do arquivo *tmsk.properties,* iniciou-se os processos de construção e de testes dos classificadores de documentos. Nessas fases foram utilizados os comandos a seguir apresentados.

- *java mkdict 30 feijao.dic*
  - mkdict → rotina para construção do dicionário
  - 30 → quantidade máxima de palavras no dicionário
  - feijao.dic → nome do dicionário criado

- *java vectorize Economics feijao_trein.vec*
  - vectorize → rotina para geração dos *sparce vectors*
  - Economics → classe especificada
  - feijao_trein.vec → arquivo contendo os S*parse* V*ectors* para treinamento do classificador

- *java vectorize Economics feijao_test.vec*
  - vectorize → rotina do TMSK para geração do *sparce vectors*
  - Economics → classe especificada
  - feijao_trein.vec → arquivo contendo os S*parse* V*ectors* para treinamento do classificador

Inicialmente, tentou-se gerar os *Sparse Vectors* utilizando as classes de documentos estabelecidas na Agência Feijão*,* mas não foi possível. Elas foram extraídas da National Agricultural Library - United States Department of Agriculture (http://www.nal.usda.gov/). Tal dificuldade se deu em função dos nomes das classes apresentarem espaços em brancos entre as palavras que as constituem. Mesmo utilizando aspas duplas no início e no final delas, o resultado não foi satisfatório.

Para resolver esse problema, foi criada uma tabela de correspondência de classes, Tabela 2. Conforme pode ser visto, cada classe na Agência passou a ser referenciada por apenas um nome. Cada um deles foi inserido no arquivo feijão_trein.xml, em substituição ao nome da classe original.

Tabela 2. Correspondência de classes.

Agência	*TMSK*
E Economics, Business and Industry	Economics
Q Food and Human Nutrition	Food
D Government, Law and Regulations	Government
X Research, Technology and Engineering	Research
F Plant Science and Plant Products	Plant
Natural Resources, Earth and Environmental Sciences	Environmental
Health and Pathology	Health
Rural and Agricultural Sociology	Rural
Farms and Farming Systems	Farms
G Breeding and Genetic Improvement	Breeding
W Physical Sciences	Physical

Após análise das classes, conclui-se que a melhor, para o objetivo desse trabalho, seria *Economics* pois poderia gerar melhores resultados. A indicação da classe é necessária para a rotulagem dos *Sparse Vectors* TMSK (Weiss et al., 2005). Essa rotulagem indica que o documento faz parte da classe definida. Uma vez definida a classe (Economics), dois classificadores foram construídos, são eles: clasfeijaonaive (Bayseano) e clasfeijaolinear (Linear). Para isso, fez-se uso dos comandos que seguem:

- *java nbayes clasfeijaonaive*
  - nbayes → rotina para construção do classificador Naive Bayes. O teorema de Bayes utilizado encontra-se descrito em Oliveira & Mendonça (2004).
  - clasfejiaonaive → arquivo contendo o classificador gerado

- *java linear clasfeijaolinear*
  - linear → rotina para construção do classificador linear. O método utilizado é uma função linear de marcação e a equação desta função encontra-se em Medeiros (2004) e Weiss et al. (2005).
  - clasfeijaolinear → arquivo contendo o classificador gerado

Além dos classificadores Naive Bayes e Linear outros podem ser construídos utilizando Redes Neurais Artificiais (RNAs) e Support Vector Machines (SMV) (Medeiros, 2004). Para utilizá-los, outros *softwares* são necessários.

Na fase de testes dos classificadores, os parâmetros infile e vectorfile tiveram seus valores alterados para feijao_test.xml e feijao_test.vec, respectivamente. Os comandos utilizados foram:

- java testnbayes clasfeijaonaive posclassfeijao negclassfeijao
  - estnbayes → ativa a rotina para testar o classificador
  - clasfeijaonaive → classificador naivebayes
  - posclassfeijao → arquivo contendo os documentos classificados de forma correta
  - negclassfeijao → arquivo contendo os documentos classificados de forma incorreta, a princípio.

- java testline clasfeijaolinear posclassfeijao negclassfeijao
  - testline → ativa a rotina para testar o classificador
  - clasfeijaolinear → classificador linear
  - posclassfeijao → arquivo contendo os documentos classificados de forma correta
  - negclassfeijao → arquivo contendo os documentos classificados de forma incorreta, a princípio.

Induzindo Regras

Para indução de regras, foi utilizado o Rule Induction Kit for Text (RIKTEXT), que gera regras compactas para auxiliar na classificação de documentos.

O RIKTEXT foi projetado para tarefa de classificação de textos em somente duas classes, o documento pertence ou não a classe definida e, assim, trabalha de forma satisfatória com grande número de atributos. Uma vez definido o nome da classe, essa passa ser a situação positiva.

O classificador baseado em regras, gerado pelo RIKTEXT, é uma lista nas quais as regras estão ordenadas. Todas elas são situações positivas, exceto a última, uma regra *default* que atende a uma situação negativa.

O RIKTEXT se utiliza dos arquivos *Sparse Vectors* gerados pelo *software* TMSK. Para o treinamento do classificador baseado em regras, *é necessário* que os S*parse Vectors* estejam rotulados, mas na aplicação dele isto não é necessário. Caso estejam, o desempenho do classificador será melhor. A geração do dicionário e dos *Sparse Vectors* seguem a mesma lógica utilizada pelo *software* TMSK.

Assim como o *software* TMSK tem seu arquivo de propriedades *tmsk.propeties*, o RIKTEXT possui o seu*, rikttext.properties* (Weiss et al., 2005). Na primeira vez que o *software* é utilizado, este é criado com valores

default para os atributos que o compõe.

O RIKTEXT é utilizado de forma *standalone*, não requerendo nenhum outro pacote de *software* e nem instalações especiais. A tecnologia e os algoritmos contidos neste *software* são discutidos em Weiss et al. (2005) e Indurkhya (2004). Todo o trabalho de preparação dos dados é realizado pelas rotinas contidas no *software* TMSK.

Várias são as opções empregadas para geração de regras e essas encontram-se mencionadas a seguir. Para maiores detalhes consulte, Indurkhya (2004).

- -q→ é utilizada para apresentação de todas a regras geradas a partir dos casos de treinamento. Dessa forma, é possível visualizar, de uma só vez, a complexidade máxima das soluções obtidas.

- -p→ ao invés da obtenção de um único conjunto de regras, como na opção anterior, são mostrados vários conjuntos de regras. Esta opção exibe um resumo, na forma de tabela, com as várias soluções. Isto é útil para exibir e comparar a complexidade das diferentes soluções e a performance entre elas. Algumas vezes, isto pode ressaltar problemas no dados ou em parâmetros que estão sendo utilizados.

- -h→ casos de teste selecionado de forma aleatória. O usuário determina a porção dos casos que serão utilizados durante o treinamento do classificador e o restante é utilizado no teste do classificador.

- -T→ se um conjunto de casos de validação está disponível, esses casos podem ser utilizados para avaliar o conjunto de regras extraídas.

- -r→ quando é limitado um número de casos e é desejável maximizar o seu uso, para aprendizado do classificador, técnicas de amostragem com reposição (resample), tal como *cross-validation,* são úteis para a obtenção do conjunto de testes estimados.

- -a→ permite que um conjunto de regras obtido seja aplicado a novos casos.

- -s→ permite selecionar um conjunto de regras específico.

Resultados e Discussão

Utilizando o *software* DbVisualizer e a linguagem SQL extraiu-se da base de dados da Agência Feijão, 180 registros compostos pelas seguintes informações: título, descrição e categoria. Esses registros foram armazenados num arquivo no formato Texto e, posteriormente, convertido para o formato XML. Uma amostra desse arquivo é exibido a seguir.

<doc>
<titulo>Mercado de feijão</titulo>
<descricao>O feijão é cultivado em mais de 100 países, porém 63% da produção mundial é obtida em apenas cinco, sendo o Brasil o maior produtor e consumidor de feijão-comum (Phaseolus vulgaris L.). Nos últimos anos foram importadas, em média, cerca de 100 mil toneladas. Da quantidade importada, a maior parte é de feijão preto, seguido pelo feijão de cores e menos que 1% é de outros tipos de feijões</descricao>
<topics>
<topic>Economics</topic></topics>
</Doc>

.....
.....

O DbVisualizer mostrou-se adequado para os processos de extração e geração dos arquivos de dados no formato ASCII, assim como o programa de conversão de formatos desenvolvido. Para efetuar mudanças nesse programa, são requeridos conhecimentos razoáveis da estrutura da linguagem Perl.

O dicionário gerado utilizou todos o documentos (registros) contidos no arquivo em XML. Isto ocorreu por não ter sido informada nenhuma categoria. Caso ela tivesse sido informada, somente documentos dessa categoria seriam utilizados.

Embora tivessem sido definidas 30 palavras para o dicionário, somente 10 (feijão, feijoeiro, solo, arroz, embrapa, irrigação, plantio, sementes, sistema, plantas) foram selecionadas. Além da eliminação das *stopwords,* outras não alcançaram a freqüência mínima de aparição nos documentos. Se o arquivo de *stopwords* não for bem estruturado, palavras indesejadas serão incorporadas ao dicionário e, isto poderá comprometer o processo de classificação de documentos.

Para classificar os documentos foram gerados dois arquivos *Sparse Vectors feijao_trein.vec* e *feijao_test.vec* conforme exemplo a seguir:

1 1@5
1 1@1 2@1
0 1@1 2@1
1 1@2
.....

} Conjunto de documentos

Cada linha do arquivo representa um documento podendo este estar ou não rotulado. No exemplo acima, os documentos contidos nas linhas 1, 2 e 4, foram rotuladas (1), ou seja, eles pertencem a classe *Economics.* O documento rotulado com 0 indica que não pertence à classe definida.

Ainda na primeira linha, a palavra de número 1 apareceu 5 vezes (1@5) no primeiro documento enquanto que no segundo documento, segunda linha, ela apareceu somente uma vez. Nesse mesmo documento, a palavra de número 2 também apareceu uma vez. As palavras que não atingiram a freqüência mínima estabelecida não são incorporadas ao dicionário.

Aplicando os comandos de teste dos classificadores Naive Bayes e Linear, já mencionados, sobre o arquivo de dados de teste, e utilizando valores *default* para os parâmetros probability-threshold e reject-threshold, contidos no arquivo de propriedades do TMSK, os resultados foram:

Resultados	**Naive Bayes**	**Linear**
Precision	66,6667%	63,6364%
Recall	50,0000%	58,3333%
F-measure	57,1429%	60,8696%

Além desses resultados, dois arquivos foram gerados. Um contendo documentos classificados de forma positiva (correta) e outro contendo os documentos classificados de forma errada, a princípio.

Esses resultados dizem respeito a precisão do classificador. *Precision* indica a percentagem de documentos que foram corretamente rotulados como pertencentes à classe. *Recall* indica a porcentagem de todos documentos pertencentes à classe em questão que conseguiram ser recuperados, medida de cobertura (Silva, 2005). *F-measure* é a média harmônica entre *Precision* e *Recall* (Weis et al., 2005).

Como pode-se ver, os resultados foram bem parecidos. Provavelmente, eles ainda poderão ser melhorados. Um exame superficial nos arquivos posclassfeijao e negclassfeijao constatou a presença de ruídos nos dados.

Esses podem ter sido introduzidos na fase de preparação dos dados. Ajustes nos parâmetros no arquivo de propriedades do TMSK poderão contribuir para aumentar a precisão dos classificadores. Como o objetivo desse trabalho foi estudar o funcionamento dos *softwares* TMSK e RIKTEXT.

Além dos classificadores Naive Bayes e Linear, algumas regras foram geradas, utilizando o *software* RIKTEXT. Para isso, foram utilizados comandos no seguinte formato: riktext - <opção> <classe> <arquivo(s) de entrada> <classe> <arquivo de treinamento | arquivo de teste> > <arquivo de resultado> onde:

- riktext → ativa o programa para indução de regras.
- <opção> → uma das opções acima apresentada.
- <arquivo(s) de entrada> → nome(s) dos arquivo(s) de entrada. Dependendo da opção, mais de um arquivo é necessário.
- <classe> → nome da classe para classificação do documento.
- <arquivo de treinamento | arquivo de teste> → nome do arquivo para treinamento ou teste do classificador.
- > → indica que o arquivo resultado será gravado.
- <arquivo de resultado> → nome do arquivo onde será gravado as respostas geradas.

A seguir são apresentados os resultados de dois comandos utilizados para geração de regras:

- riktext - q feijao.dic Economics feijao_trein,vec > feijao.rul

RSet	Rules	Vars	TrainErr	TestErr	TestSD	MeanVar	Err/Var
1	5	10	0.789	0.0000	0.0000	0.0	0.00

- Rset → número do conjunto de regras. O conjunto de regras marcado com * é visto como o conjunto que apresenta o padrão de erro mínimo. O conjunto de regras marcado por **, indica que está no mínimo ou muito próximo do mínimo, mas também pode está abaixo do mínimo.

- Rules → número de regras contidas em Rset.
- Vars → número total de conjunções do lado esquerdo das regras que pertencem ao conjunto de regras.
- TrainErr → taxa do erro do conjunto de regras geradas a partir dos dados de treinamento.
- TestErr → indica a taxa do erro estimada.
- TestSD → indica o desvio padrão do erro.
- MeanVar → indica o número médio ou as variáveis do conjunto de regras com reposição, que mais se aproxima do conjunto de todas as regras. Isto ajuda a determinar a confiabilidade das amostras com reposição estimativas.
- Err/Var → indica o número de novos erros por variável, que foram introduzidos, quando o conjunto de erros original foi diminuído de tamanho.

Mais detalhes sobre cada uma das variáveis apresentadas são obtidos em Indurkhya (2004). As regras exibidas foram:

1. produção & cultura→ Economics
2. produção & cultivar→ Economics
3. produção & regiões→ Economics
4. cultura & feijão & regiões→ Economics
5. [TRUE]→ ~Economics

Em linhas gerais, os documentos que apresentaram no título ou na descrição as combinações produção & cultura, produção & cultivar, produção & regiões e cultura & feijão & regiões foram classificados na classe Economics. A regra número 5 representa os casos que não apresentaram nenhuma das combinações apresentadas. Daí não puderam ser classificados na classe Economics.

E para encerrar a apresentação dos resultados, as estatísticas adicionais referentes aos casos de treinamento (Training Cases) são apresentadas:

Precision	Recall	F-measure
100%	50%	73%

- riktext - h 66.7 feijão.dic Economics feijao_trein.vec > feijao.rul

Neste segundo caso, 66,7% dos documentos foram utilizados para treinamento e 33,3% para teste do classificador. Os documentos para o teste do classificador foram escolhidos de forma aleatória. Os resultados foram:

RSet	Rules	Vars	TrainErr	TestErr	TestSD	MeanVar	Err/Var
1	3	5	0.0658	0.0789	0.0437	0.0	-0.20
2	2	3	0.0789	0.1053	0.0498	0.0	0.50
3	1	1	0.1053	0.1053	0.0498	0.0	1.00

Na seção Select rule set desse comando, o resultado apresentado foi:

1. [TRUE] → ~Economics, nenhum documento foi encontrado para as condições estabelecidas.

As estatísticas adicionais foram:

Estatísticas Adicionais	Precision (%)	Recall (%)	F-measure (%)
Casos de Treinamento	100	0	0
Casos de Teste	100	0	0

Em ambos os casos, o valor do Recall foi zero. Acredita-se que tal fato ocorreu em função de nenhuma regra válida ter sido encontrada para a classe *Economics*. Em trabalhos futuros, isso será investigado.

Considerações Finais

Durante os estudos, foram abordados conceitos básicos importantes para o entendimento inicial a respeito do processo de classificação de documentos. Buscou-se, também, apresentar de forma simplificada, as fases e as particularidades de cada etapa do processo de classificação de documentos. Junto a elas encontra-se o processo de indução de regras. O processo de indução de regras depende da execução de rotinas do *software* TMSK. Para consolidação dos conhecimentos adquiridos, dois classificadores foram gerados e testados. Encerrado os testes, pode-se concluir que a etapa de preparação dos dados é trabalhosa e consome bastante tempo. Quanto mais bem estruturada a base de dados, mais precisos serão os modelos preditivos construídos.

Quanto ao desempenho dos classificadores, os resultados iniciais atenderam as expectativas. Para uma melhor avaliação deles, é necessário uma base de dados maior e testes com novos valores para os parâmetros de configuração

O TMSK apresenta bons recursos para as fases de pré-processamento, construção e testes de classificadores de textos, mas a interface humano-computador um ponto fraco. O TMSK apresenta bons recursos para as fases de pré-processamento, construção e testes de classificadores de textos. Embora os *softwares* TMSK e RIKTEXT sejam utilizados sobre o Windows, toda interação humano-computador é através via linhas de comandos.

Um ponto fraco do TMSK e do RIKTEXT diz respeito à configuração dos parâmetros dos arquivos de configuração de ambos os *softwares*. Para realizá-la, é necessário editar de forma direta esses arquivos. Caso o usuário não tenha noções de informática, problemas poderão ocorrer ou, em alguns casos, danificar o arquivo. Por medida de segurança, mantenha uma cópia deles

Quanto ao RIKTEXT, pode-se dizer que seu uso é simples, bastando para isso entender e distinguir o funcionamento das opções ofertadas. Um ponto fraco desse *software* é a dependência do *software* TMSK.

Como trabalhos futuros, estão previstos estudos mais aprofundados das rotinas de geração dos classificadores Naive Bayes e Linear e também, da rotina de agrupamento de documentos. Junto a esses estudos, serão investigados, com mais profundidade, a utilização dos parâmetros de configuração de ambos os *softwares*, visando a obtenção de melhores resultados.

Referências Bibliográficas

COUTINHO, F. V. *Data mining*. [S. l.]: DW Brasil, 2003. Disponível em: <http://www.dwbrasil.com.br/html/dmining.html>. Acesso em: out. 2006.

INDURKHYA, N. *TMSK*: Text-Miner Software Kit. 2004. 35 p. Disponível em: <http://www.data-miner.com/tmsk.pdf>. Acesso em: 15 fev. 2007.

MASSRUHÁ, S. M. F. S. (Coord.). *Incorporação de ferramentas inteligentes na Agência de Informação Embrapa*. [Campinas: Embrapa Informática Agropecuária, 2004]. 30 p. (Embrapa. Macroprograma 3 - Desenvolvimento Tecnológico Incremental. Projeto).

MEDEIROS, E. A. de. *Técnicas de aprendizado de máquina para categorização de textos*. Recife: Universidade de Pernambuco - Escola Politécnica de Pernambuco, 2004. 61 p. Trabalho de Conclusão de curso de Engenharia da Computação.

MOURA, M. F. *Proposta de utilização de mineração de textos para seleção, classificação e qualificação de documentos*. Campinas: Embrapa Informática 2004. 30 p. (Embrapa Informática Agropecuária. Documentos, 47).

OLIVEIRA, G.; MENDONÇA, M. ExperText: uma ferramenta de combinação de múltiplos classificadores Naives Bayes. In: JORNADA IBERO-AMERICANA DE ENGENHARIA DE SOFTWARE E ENGENHARIA DE CONHECIMENTO, 2004, Madri. *Anales de la 4a Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería de Conocimiento*. Madri: Servício de Publicaciones de la Faculdad de Informática de la Universidad Politécnica de Madrid, 2004. v. 1, p. 317-332.

PICHILIANI, M. *Data mining na prática:* algoritmo K-means. 2006. Disponível em:<http://www.imasters.com.br/artigo/4709/sql_server/data_mining_na_pratica_algoritmo_k-means/>. Acesso em: 16 fev. 2007.

PIERRE, J. M. *Mining knowledge from text collections using automatically generated metadata*. [2002]. Disponível em: <http://www.sukidog.com/jpierre/pakm2002.pdf>. Acesso em: 15 fev. 2007.

REZENDE, S. O. *Sistemas inteligentes:* fundamentos e aplicações. Barueri: Ed. Manole, 2003. 525 p.

SILVA, J. U. *Text mining como uma aplicação na validação dos registros de ocorrências policiais na Região da Grande Florianópolis*. 2005. 122 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Santa Catarina, Florianópolis.

WEISS, S. M.; INDURKYA, N.; ZHANG, T.; DAMERAU, F. J. *Text mining:* predictive methods for analyzing unstructured information. New York: Springer, 2005. 237 p.

Embrapa

Informática Agropecuária

Ministério da
Agricultura, Pecuária
e Abastecimento

BRASIL
UM PAÍS DE TODOS
GOVERNO FEDERAL