Anais do 6º Congresso Brasileiro de Agroinformática – SBIAgro 2007, São Pedro, SP, Brasil, 8-11 outubro 2007,
Embrapa Informática Agropecuária, p.386-390.

TMSK E RIKTEXT: SOFTWARE PARA CLASSIFICAÇÃO DE TEXTOS E INDUÇÃO DE REGRAS

LUIZ MANOEL SILVA CUNHA[1]
SILVIA MARIA FONSECA SILVEIRA MASSRUHÁ[2]
LEANDRO HENRIQUE MENDONÇA OLIVEIRA[3]

RESUMO: No mundo moderno, as companhias, cada vez mais, procurar novas maneiras de extrair conhecimentos novos de documentos não estruturados e classificá-los para o uso em diversas situações, como exemplo, organização de bases de conhecimentos. Para isso, sistemas inteligentes estão sendo usados. Este trabalho apresenta os estágios envolvidos nos processos de seleção e de classificação de textos e de indução das regras, no contexto de mineração de textos, e como foram aplicados os softwares TMSK (*Text-Miner Software Kit*) e RIKTEXT (*Rule Induction Kit for Text*) nestes estágios. Dois classificadores de textos e conjuntos foram gerados, bem como um conjunto de regras. O TMSK apresentou bons recursos para pré-processamento de textos, desenvolvimento e testes dos classificadores. O RIKTEXT complementa TMSK, fornecendo métodos para a indução de regras de classificação fáceis de serem utilizados. Ambos os *softwares* foram executados no sistema operacional Windows XP, utilizando interface textual.

PALAVRAS-CHAVE: Mineração de Texto**,** Descoberta de Conhecimentos, Sistema Inteligente.

TMSK AND RIKTEXT: SOFTWARE FOR CLASSIFICATION OF TEXTS AND RULES INDUCTION

ABSTRACT: In the modern world, the companies, each time more, search new ways to extract new knowledge from unstructured document and to classify them for use in some situations, for exemple, knowledge bases organization. For this, intelligent systems are being used. In this work is presented the involved stages in the processes of text classification, induction of rules and how to run softwares TMSK (Text Miner Software Kit) and RIKTEXT (Rule Induction Kit Text) in these stages. Two classifiers and a classification rules set had been developed. The TMSK presented good resources for pre-processing of texts, classifiers construction and tests. The RIKTEXT complements TMSK by providing methods for induction of classification rules. Both softwares had been executed in the operational system Windows XP, using textual interface.

KEY-WORDS: Text Mining**,** Discovery of Knowledge, Intelligent System.

1. INTRODUÇÃO

A Empresa Brasileira de Pesquisa Agropecuária – Embrapa, ao longo de sua existência, gerou um grande acervo documental oriundo dos resultados obtidos em projetos de pesquisas. Alguns destes projetos foram executados de modo a tornar disponível o conteúdo deste acervo visando a organização, estruturação, armazenamento e recuperação de informação, de forma adequada. A Agência de Informação Embrapa, ou simplesmente Agência, é um projeto que

[1] Msc. em Matemática Computacional e Ciência da Computação, Analista da Embrapa Informática Agropecuária, Email: luizm@cnptia.embrapa.br.
[2]Doutora em Computação Aplicada, Pesquisadora da Embrapa Informática Agropecuária, Email: silvia@cnptia.embrapa.br.
[3]Msc. em Ciência da Computação, Analista da Embrapa Informática Agropecuária, Email: leandro@cnptia.embrapa.br.

foi concebido com objetivo de estabelecer um portal *Web* composto por várias Agências de produtos, organizando informações técnicas relevantes das várias cadeias produtivas do agronegócio brasileiro (Moura, 2004). A metodologia adotada na organização da informação da Agência é um diferencial em relação às outras metodologias utilizadas em outros projetos desenvolvidos na Embrapa. Ela permite que o conhecimento da cadeia produtiva seja armazenado de maneira hierárquica, em uma estrutura de árvore (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 descritos em vários formatos, por exemplo, texto, vídeo e outros, oriundos de várias fontes. As tarefas de seleção e de classificação de recursos de informação são partes importantes dentro da metodologia aplicada nas Agências. É através delas que são decididos quais recursos serão anexados aos nós da árvore. Com isso, estará assegurada uma melhor qualidade das informações recuperadas e disponibilizadas pelas Agências. Caso contrário, o processo de recuperação ficará comprometido, ou seja, informações indesejáveis serão disponibilizadas na Internet. Esta tarefas manipulam muitas referências a outras obras que completam a informação, e também àquela utilizada para construção/atualização de seus conteúdos. Isto faz com que estas tarefas não sejam triviais. Além disso, envolvem especialistas em informação e especialistas do domínio. Para agilizar e facilitar a execução destes processos, pesquisadores das áreas de Mineração de Texto, Gestão da Informação e de áreas correlatas vêm buscando, através do desenvolvimento de *softwares*, automatizar ou semi-automatizar estas tarefas (Moura, 2004). A Mineração de Texto pode ser definida como um processo para extração de padrões ou conhecimentos interessantes e desconhecidos em documentos textuais (Weiss, 2005). Esse processo pode ser aplicado em seleção, classificação e agrupamento de documentos. Visando a melhoria do processo de Classificação de textos do Projeto Agência, foi proposto o projeto Incorporação de Ferramentas Inteligentes na Agência de Informação Embrapa (Massruhá et al, 2005). Esse projeto tem como objetivo evoluir e incorporar novas ferramentas de apoio à metodologia de estruturação das Agências, visando o reuso de informações contidas nelas, bem como a incorporação de outros serviços que facilitem a transferência de tecnologia e conhecimento. Para inclusão de melhorias nos processos de Seleção, Classificação e Qualificação de Dados Textuais e, também, identificar associações entre os dados, foram identificados e estudados *softwares* que incorporam componentes inteligentes (Rezende, 2003). Assim, combinando-se os resultados gerados por estes *softwares* com os resultados do processo manual em uso, pretende-se estabelecer um novo processo semi-automatizado que configure maior produtividade e maior confiabilidade de resultados. Como resultados parciais destes estudos, conceitos importantes utilizados na seleção e classificação de textos foram levantados, absorvidos e aplicados. Também, foram exercitaaos as etapas para classificação de textos segundo Weiss (2005), utilizando os *softwares* TMSK e RIKTEXT, o que evidenciou as virtudes e as fraquezas desses *softwares.* Estes resultados são importantes para tomada de decisão visando a incorporação deles à Agência de Informação Embrapa. Na seção 2, são apresentados os objetivos desse trabalho. Na seção 3, encontram-se os materiais e os métodos. Na seção 4, são apresentados os resultados alcançados com a aplicação dos *softwares* sobre um arquivo contendo cento e vinte documentos. As conclusões estão descritas na seção 5.

2. OBJETIVOS

Apresentar as fases e o relacionamento delas nos processos de classificação de textos e de indução de regras, conceitos utilizados e resultados obtidos da aplicação dos softwares TMSK e RIKTEXT sobre uma base de dados.

3. MATERIAL E MÉTODOS

O processo de classificação de textos descrito em Weiss (2005) e Oliveira (2004) é composto

dos seguintes passos: a) seleção de documentos; b) pré-processamento de documentos; c) desenvolvimento dos classificadores e d) testes dos classificadores. Como esse trabalho envolveu estudos do *software* RIKTEXT, o passo (e), indução de regras, foi acrescentado ao processo. Na etapa de seleção de documentos, aplicou-se um filtro para extrai-los da base de dados da Agência Feijão[4], um conjunto de 120 registros para formação de um arquivo, no formato Texto (TXT). Para isso, foram utilizados recursos do *software* DbVisualizer (http://www.minq.se/). Na etapa de pré-processamento, o arquivo TXT foi convertido para o formato eXtensible Markup Language (XML), utilizando um programa escrito na linguagem de programação Perl. Essa conversão fez-se necessária para que o arquivo convertido pudesse ser reconhecido pelo *software* TMSK. O arquivo TXT foi dividido em dois outros: um com 84 registros, para auxiliar no desenvolvimento dos modelos de classificação (classificadores) e o segundo, com 36 registros, para testá-los. Um arquivo de S*topwords* foi criado. As S*topwords* são palavras contidas nos documentos que não possuem conteúdo semântico significativo no contexto, sendo consideradas não relevantes para o processo de análise. Estas palavras podem ser artigos, pronomes, advérbios, verbos, numerais, conjunções, entre outras. Para abrigar as palavras que têm relevância nos documentos pelas freqüências de aparição nos textos, um dicionário foi criado. Sem este dicionário, dificuldades são encontradas para construção dos *Sparse Vectors*. Estes são vetores que armazenam as palavras que aparecem nos documentos seguidos de suas respectivas freqüências (Weiss, 2005). Para cada documento, um vetor é criado. As palavras que não aparecem nos documentos recebem zero para suas respectivas freqüências e são removidas dos vetores. Os dados armazenados nestes vetores seguem o seguinte formato: (x@y), onde x representa a palavra e y o número de vezes em que ela aparece no documento. Durante o processo de investigação dos *softwares* TMSK e RIKTEX, dois arquivos contendo *Sparse Vectors* foram criados: *feijão_trein.vec e feijão_test.vec.* O primeiro, foi utilizado no desenvolvimento dos classificadores e, o segundo para testá-los. Após criados, os *Sparse Vectors* foram rotulados*.* Um rótulo serve para indicar se o documento pertence ou não a uma classe, também conhecida como categoria, de documentos previamente definida. Uma vez obtido o classificador, esse rótulo é utilizado na classificação de novos documentos, ou seja, documentos sem rótulos. Nesse trabalho, foi utilizada a classe ***"****Economics****"***, para criação dos rótulos, por se tratar daquela de maior representatividade, ou seja, classe com maior número de documentos classificados dentre as várias contidas no conjunto de dados analisado. Para a construção dos *Sparse Vectors* e a rotulagem deles, foi utilizada a função *vectorize,* integrante do pacote TMSK. A execução destas duas fases foram e são indispensáveis para criação dos classificadores. Caso contrário, problemas ocorrerão no desenvolvimento dos classificadores. A Figura 1, exibe um exemplo parcial de *Sparse Vectors* já rotulados.

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

Figura 1: Exemplo parcial dos *Sparse Vectors* rotulados.

Nesse trabalho foram criados dois classificadores: o Naive Bayes e o Linear. O teorema que suporta o primeiro classificador encontra-se descrito em Oliveira (2004). O segundo foi construído com base na função linear, aquela que estabelece entre x e y uma relação tal que y/x é constante é dita linear. Esta função é discutida em Medeiros (2004). As rotinas *nbayes e linear*, ambas contidas no *software* TMSK, foram aplicadas. Indurkhya (2004) apresenta como utilizá-las, bem como seus respectivos parâmetros. Para entender o funcionamento do

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

software RIKTEXT, um conjunto de regras para classificação de textos foram gerados partindo dos *Sparse Vectors* rotulados. Para isso, utilizou-se a rotina *riktext*, contida nesse *software,* variando alguns de seus parâmetros. Também em Indurkhya (2004), encontram-se explicações de como utilizar essa rotina e seus parâmetros. O RIKTEXT somente gera regras, todo trabalho de preparação dos dados é executado pelo TMSK.

4. RESULTADOS E DISCUSSÃO

Além dos *softwares* já mencionados, foram utilizados: a plataforma de desenvolvimento Java 2 (SDK) e o sistema operacional Windows XP, todos eles instalados e configurados num computador com 512 Mbytes de memória RAM. O trecho, em negrito, exibe parte do arquivo convertido do formato TXT para o formato XML.

<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.).....</descricao>
<topics>
<topic>Economics</topic></topics>
</doc>

Os documentos foram separados pelas *tags* (marcas) <doc>..</doc> e compostos pelos campos título, descrição e categoria, que é descrita entre as *tags* <topic>...</topic>. Quando o documento é classificado em mais de uma categoria, elas são colocadas entre as *tags* <topics>....</topics>. Utilizando o arquivo *feijão_trein.*vec, foram obtidos os classificadores *clasfeijaonaive* (Naive Bayes) e *clasfeijaolinear* (Linear). Aplicando ambos sobre a massa de dados de teste e utilizando o arquivo *feijão_test.vec*, os resultados alcançados são mostrados na Tabela 1.

Tabela 1: Resultados da aplicação dos classificadores.

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

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; é uma medida de cobertura. *F-measure* é a média harmônica entre *Precision* e *Recall* (Weis, 2004). Além da tabela, dois arquivos foram gerados: *posclassfeijao* e *negclassfeijao.* O primeiro, armazena os documentos classificados de forma correta e o segundo, contém os documentos que, inicialmente, foram classificados de forma errônea. De posse desse último arquivo, é possível investigar as causas que levaram estes documentos a serem classificados de forma errada. De acordo com os resultados exibidos, o classificador Linear apresentou, no geral, um melhor resultado. Acredita-se que estes resultados podem ser melhorados através de análises do arquivo *negclassfeijão* e/ou ajustando parâmetros contidos no arquivo *tmsk_proprieties*. Estes processos serão realizados em estudos posteriores. Utilizando-se o RIKETXT sobre o arquivo no formato XML, dividido em 66.7% dos registros para geração do classificador e os 33.7% restante para teste do mesmos, os resultados obtidos estão apresentados na Tabela 2.

Tabela 2: Resultados da aplicação do RIKTEXT.

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

Foram gerados três conjuntos de regras (Rset), cada um deles contendo um determinado número de regras (Rules). Uma regra é descrita da seguinte forma: i*f A then B*, A representa as conjunções (lado esquerdo) e B a expressão (lado direito) da regra. A complexidade das regras está vinculada aos valores dos atributos Rules e Vars. Se o número de regras e de conjunções são próximos, isto indica que as regras geradas são simples. O atributo TrainErr, indica a taxa de erro do conjunto de regras no treinamento dos dados. Na Tabela 2, o conjunto de regras número 1 é o que apresenta a menor taxa de erro. Quando comparado os conjuntos de regras, o erro do treinamento indica um acréscimo do limite (*upper-bound*) de performance para futura execução. Segundo Indurkhya (2004), é pouco provável que a performance, para novos casos, venha a ser melhor do que a dos casos de treinamento utilizados para derivar as regras, na primeira vez. TestErr (taxa de erro estimado) e TestSD (desvio padrão do erro estimado) estão relacionados a desempenhos futuros. O valor estimado para TestErr depende de como o RIKTEXT é utilizado: através da seleção aleatória de casos de teste para treinamento do classificador, por reutilização (cross-validation) ou por conjunto de testes individualizados. MeanVar é o número médio ou variáveis do conjunto de regras que foram reutilizadas e que se aproximam em tamanho para todo conjunto de dados. Este parâmetro ajuda a determinar a confiabilidade das estimativas reutilizadas (*resampled*). Err/var indica o número de novos erros por variáveis que foram introduzidos quando o conjunto de regras foi reduzido ao menor tamanho. Isto indica a qualidade das soluções.

5. CONCLUSÕES

Este trabalho apresentou as fases envolvidas no processo de classificação de documentos (textos) e como os *softwares* TMSK e RIKTEXT podem suportá-las. O TMSK apresentou bons recursos para as fases de pré-processamento, construção e testes de classificadores de textos. O RIKTEXT contém várias opções para geração de regras mas é dependente do TMSK na preparação dos dados. Em ambos os softwares, toda a interação humano-computador é realizada via linha de comando, o que não chega a ser uma dificuldade para utilizá-los. Como trabalhos futuros, estão previstos estudos mais aprofundados das rotinas de geração dos classificadores Naive Bayes e Linear e testes com a rotina de agrupamento de documentos. Além disso, serão investigados os parâmetros dos arquivos *tmsk_proprieties* e *riktext_proprieties,* visando a melhoria de performance dos classificadores e geração de regras.

6. REFERÊNCIAS BIBLIOGRÁFICAS

REZENDE, S. O. **Sistemas Inteligentes - Fundamentos e Aplicações.** Ed. Manole, 1ª Edição, 2003, 525p.

WEISS, S.M.; INDURKYA, N. ZHANG, T.; DAMERAU, F.J. TEXT MINING: **Predictive Methods for Analyzing Unstructured Information**. New York, NY:Springer, 2005. 237p.

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, Espanha: Servício de Publicaciones de la Faculdad de Informática de la UPM (ww.fi.upm.es), 2004. v. 1, p. 317-332.

MOURA, M. F. Proposta de utilização de Mineração de Textos para Seleção, Classificação e Qualificação de Documentos. Campinas: Embrapa Informática 2004, 30p. (Embrapa Informática Agropecuária. Documentos, 47).

MASSRUHÁ, S.M.F.S. Incorporação de ferramentas inteligentes na Agência de Informação Embrapa. [Campinas: Embrapa Informática Agropecuária, 2004]. 30p. (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 Pernanbuco, 2004. 61p. Trabalho de Conclusão de curso de Engenharia da Computação.

INDURKHYA, N. TMSK: Text-Miner Software Kit: Manual do Usuário, 2004, 35p.