Aprendizado de classificadores das ementas da Jurisprudência do Tribunal Regional do Trabalho da 2ª. Região - SP

Thiago Ferauche, Maurício Amaral de Almeida
Laboratório de Pesquisa em Ciência de Serviços – Programa de Mestrado -
Centro Estadual de Educação Tecnológica Paula Souza – SP – Brasil
thiago.ferauche@gmail.com, madealmeida@gmail.com

Abstract – This paper presents the use of text mining techniques, and the results of training and testing of three classifiers algorithms using the Jurisprudence summaries of Tribunal Regional do Trabalho da 2ª Região – SP.

Keywords: Jurisprudence, Text Mining, Machine Learning, Text Categorization.

Resumo – Este artigo apresenta a aplicação de técnicas de mineração de textos, e os resultados obtidos durante o treinamento e teste de três algoritmos classificadores utilizando as ementas da Jurisprudência do Tribunal Regional do Trabalho da 2ª Região – SP.

Palavras-chave: Jurisprudência, Mineração de Textos, Aprendizado de Máquina, Classificação de Textos.

Introdução

A ementa é um resumo de uma decisão (acórdão) tomada por um colegiado de desembargadores. As ementas das decisões mais relevantes compõem a jurisprudência de um Tribunal. Com a finalidade de facilitar a pesquisa jurisprudencial do Tribunal Regional do Trabalho da 2ª. Região – São Paulo, um especialista em Direito realiza a tarefa de classificá-las, seguindo a ontologia mantida pela Secretaria de Gestão da Informação Institucional, porém de maneira empírica e altamente dependente do nível de conhecimento e experiência do especialista.

A Mineração de Textos (MT) tem como objetivo descobrir informações relevantes através de dados não-estruturados, contidos em formato texto. Uma definição genérica inclui todos os tipos de processamento de texto que tratam de encontrar, organizar e analisar informação [1].

A aplicação de técnicas de MT e de Inteligência Artificial permite a formação de um mecanismo de auxílio à tarefa de classificação das ementas trabalhistas. A classificação, em Mineração de Textos, visa a identificar os tópicos principais em um documento e associar este documento a uma ou mais categorias predefinidas [2]. Segundo Konchady [1], o problema da classificação pode ser descrito como a classificação de documentos em múltiplas categorias, onde se tem um conjunto de *n* categorias $\{C_1, C_2, \ldots C_n\}$ para as quais são associados *m* documentos $\{D_1, D_2, \ldots D_m\}$.

Conforme Feldman e Sanger [3], assim como em outras tarefas de inteligência artificial, existem duas principais abordagens para a classificação

de textos. A primeira é a abordagem da engenharia do conhecimento (*knowledge engineering*) onde o conhecimento de especialistas sobre as categorias está codificado no sistema, seja declarativamente ou na forma de regras procedimentais de classificação. A outra abordagem é o aprendizado de máquina (*machine learning*) onde geralmente através de um processo indutivo é construído um classificador aprendendo a partir de um conjunto de exemplos pré-classificados. Este trabalho utiliza a abordagem do aprendizado de máquina, devido a alta complexidade em extrair o conhecimento dos especialistas e expressá-las dentro do código. Desta maneira foi utilizada uma coleção de ementas classificadas pelos especialistas em Direito como exemplos pré-classificados para o treinamento dos algoritmos classificadores, e outra coleção de ementas para a validação do aprendizado dos algoritmos classificadores.

Metodologia

A análise estatística de textos demonstra ser a mais interessante para se aplicar a textos jurídicos, pois os textos empregam uma linguagem técnica com muitos termos em latim. Nesse tipo de análise, a importância dos termos é dada basicamente pelo número de vezes que eles aparecem nos textos. É interessante ressaltar que este tipo de estratégia pode ser conduzido independentemente do idioma [2].

O processo de mineração de texto pode ser dividido em quatro etapas, conforme Gonçalvez e Rezende [4]:

1. **Coleta de Documentos:** nesta fase, os documentos relacionados com o domínio da aplicação final são coletados.
2. **Pré-processamento:** consiste de um conjunto de ações realizadas sobre o conjunto de textos obtido na etapa anterior, com o objetivo de prepará-los para a extração de conhecimento.
3. **Extração de Conhecimento:** utilizam-se alguns algoritmos de aprendizado com o objetivo de extrair, a partir de documentos pré-processados, conhecimento na forma de regras de associação, relações, segmentação, classificação de textos, entre outros.
4. **Avaliação e Interpretação dos Resultados:** nessa etapa os resultados obtidos são analisados, filtrados e selecionados para que o usuário possa ter um melhor entendimento dos textos coletados. Esse entendimento maior pode auxiliar em algum processo de tomada de decisão.

Com o objetivo de validar o aprendizado de máquina utilizando as ementas da jurisprudência do Tribunal Regional do Trabalho da 2ª. Região – SP, foi utilizada a análise estatística seguindo as 4 etapas citadas anteriormente, adaptando a etapa de Avaliação e Interpretação dos Resultados para comparar os resultados de indução dos algoritmos classificadores com a classificação dos especialistas previamente realizada.

Os classificadores que trouxeram melhores resultados em pesquisas já realizadas foram: SVM, AdaBoost, kNN e métodos de regressão. Naive Bayes apesar de não ter apresentado bons resultados, é muito utilizado em conjunto com outros classificadores. As árvores de decisão foram pouco utilizadas como classificadores, e em alguns resultados foram quase tão bem quanto o SVM

[3]. Desta maneira foram utilizados os algoritmos SVM, pois já apresentou bons resultados em pesquisas anteriores, Naive Bayes para verificar a viabilidade de utilizar seus resultados com outros algoritmos e Árvores de Decisão para comparar com os resultados do SVM.

Coleta de Documentos

Os conteúdos das Ementas foram extraídos a partir de arquivos textos, originalmente utilizados para o envio às Editoras que compõem a revista jurisprudencial. Tais arquivos possuem a extensão ".JUR" e layout próprio. Cada arquivo deste contém ementas de um mês, com todas as categorias misturadas. Os arquivos utilizados como exemplos para o treinamento dos algoritmos correspondem aos meses de janeiro a dezembro, dos anos de 2008 a 2010. Os arquivos utilizados como exemplos para testes dos algoritmos correspondem ao mês de janeiro de 2011. Pode ser verificado nesta fase que a quantidade de documentos não era distribuída de forma equânime pelas categorias, muito pelo contrário existiam categorias com um único documento, e categorias com milhares de documentos. As categorias das ementas também apresentaram uma distribuição irregular quanto a relação entre quantidade de documentos e tamanho de bytes de cada categoria, de forma que existem categorias com menos documentos, porém com mais informação, assim como existem categorias com menos informação, porém com mais documentos, conforme a Tabela 1.

Tabela 1: Exemplo de 10 categorias e a distribuição de seus documentos e tamanho em bytes

CATEGORIA	QUANTIDADE DE DOCUMENTOS	TAMANHO (Bytes)
PREVIDENCIA SOCIAL	12865	10.250.190
EXECUÇÃO	5370	4.191.805
MÃO-DE-OBRA	4308	4.743.027
EMBARGOS DECLARATÓRIOS	4248	2.561.049
PROVA	3689	2.867.747
RELAÇÃO DE EMPREGO	2922	2.583.514
PRESCRIÇÃO	2834	2.728.237
DANO MORAL E MATERIAL	2532	2.529.515
COMPETÊNCIA	2151	2.397.916
SINDICATO OU FEDERAÇÃO	2094	2.079.449

Pré-processamento

Nesta etapa é necessário preparar os documentos, e extrair um conjunto de características dos mesmos, utilizando a abordagem estatística, para formar o chamado vetor atributo-valor, onde cada termo é um atributo do vetor, e é atribuído um valor para cada atributo. Para a extração dos termos e atribuição de valor aos mesmos, foi utilizada a ferramenta PRETEXT II [5] pois utiliza a técnica de *bag of words,* e faz uso de cortes de palavras baseados em frequência, utilizando a Lei de Zipf [5] e os cortes de Luhn [5], para restringir o

problema da alta dimensionalidade do vetor atributo-valor geralmente ocorrida na mineração de textos. Como métrica foi utilizado o critério de medida *Term Frequency – Inverse Document Frequency* (tf-idf), e critérios de suavização e normalização quadrática por atributo (coluna), com o objetivo de amenizar o problema da irregularidade de distribuição da quantidade de documentos e de informação nas categorias, capturando assim o máximo das características relevantes dos documentos.

Tabela 2: Exemplo de 3 Categorias utilizadas e quantidade de exemplos selecionados

Categoria	Real[1]	Selec[2]	Outras	Real[1]	Selec[2]
EXECUÇÃO	5370	500	EMBARGOS DECLARATÓRIOS	4248	181
			RELAÇÃO DE EMPREGO	2922	125
			SINDICATO OU FEDERAÇÃO	2094	89
			MANDADO DE SEGURANÇA	1612	69
			RESPONSABILIDADE SOLIDÁRIA/SUBSIDIÁRIA	895	39
			Total de Outros	**11771**	**503**
MÃO-DE-OBRA	4308	500	PROVA	3689	212
			SINDICATO OU FEDERAÇÃO	2094	121
			RECURSO	1297	75
			RESPONSABILIDADE SOLIDÁRIA/SUBSIDIÁRIA	895	52
			PROCESSO	726	42
			Total de Outros	**8701**	**502**
EMBARGOS DECLARATÓRIOS	4248	500	PRESCRIÇÃO	2834	171
			SINDICATO OU FEDERAÇÃO	2094	126
			CONCILIAÇÃO	1377	83
			HORAS EXTRAS	1136	69
			NORMA COLETIVA (EM GERAL)	885	54
			Total de Outros	**8326**	**503**

Do total de 187 categorias, foram escolhidas 10 categorias que possuem no mínimo 500 documentos, distribuídos da seguinte maneira: uma categoria que possui até 1000 documentos, duas categorias que possuem entre 1000 e 2000 documentos, duas categorias que possuem entre 2000 e 3000 documentos, uma categoria que possue entre 3000 e 4000 documentos, duas categorias que possuem entre 4000 a 5000 documentos e duas categorias que possuem acima de 5000 documentos. Não foram utilizadas todas as categorias nem selecionados todos os documentos das categorias escolhidas, pois não houve poder computacional disponível para tanto, pelo mesmo motivo foi decidido construir classificadores binários para cada uma das 10 categorias. Foram selecionados aleatoriamente 500 documentos de uma categoria, confrontados com mais 500 documentos de 5 das 177 categorias restantes, selecionadas também aleatoriamente respeitando a distribuição proporcional da quantidade de documentos real das 187 categorias, seguindo a teoria *PAC-learning* [6], compondo um conjunto de exemplos para treinamento que

[1] Quantidade real de exemplos presentes na categoria
[2] Quantidade de exemplos selecionados aleatoriamente

contenham uma distribuição de exemplos positivos (da categoria que pretende-se aprender) e de exemplos negativos (das outras categorias diversas), conforme pode ser visto na Tabela 2.

Foram gerados os vetores atributo-valor de cada categoria a ser aprendida, em conjunto com outras categorias selecionadas de maneira aleatória. O PRE-TEXT trabalha com um formato próprio de tabela atributo-valor, e para servir de entrada para a fase seguinte deve ser traduzido para o formato ARFF (Attribute-Relation File Format).

Extração de Conhecimento

Os vetores atributo-valor, após a tradução para o formato ARFF (Attribute-Relation File Format) foram inseridos na ferramenta WEKA - *Waikato Environment for Knowledge Analysis*, para que os dados fossem processados por algoritmos de aprendizado de máquina, e assim fossem criados modelos de conhecimento. Foi utilizado o algoritmo J4.8 como implementação do algoritmo de árvore de decisão disponível através da ferramenta WEKA. É uma implementação posterior, com poucas melhorias do algoritmo C4.5 *revision* 8. A ferramenta WEKA possui também a implementação do classificador probabilístico Naive Bayes, utilizando a distribuição normal para modelar os atributos [7]. Uma variante do algoritmo SVM, denonimada SMO (*Sequential Minimal Optimization*), foi utilizado como algoritmo classificador SVM, sendo implementada através da ferramenta WEKA. O SMO surgiu da necessidade de implementação de um algoritmo SVM de maneira rápida, simples e capaz de tratar conjuntos de dados mais extensos. Além disso, possui a capacidade de tratar um conjunto de dados esparsos, que possuem um número substancial de elementos com valor zero. Park [8] afirma que a optimização realizada no SMO encontra-se na programação quadrática analítica, ao invés da abordagem numérica tradicional.

Portanto, foram montados 3 modelos de aprendizado, utilizando as 3 implementações de algoritmos de aprendizado (J4.8, Naive Bayes e SMO), para cada uma das 10 categorias selecionadas. A técnica utilizada para o treinamento dos algoritmos foi o *cross-validation*. Essa técnica quebra o conjunto de exemplos em dois, um conjunto usado para treinar o algoritmo e outro utilizado para testá-lo, de forma a poder avaliar a precisão do algoritmo treinado. A escolha dos exemplos para cada conjunto é realizada de forma aleatória, e para que o algoritmo aprenda com uma diversidade maior de exemplos, e possa ir ajustando sua taxa de erro, é recomendado repetir o processo várias vezes, alternando os exemplos dos conjuntos [7]. É possível fixar o número de *folds*, ou partições dos exemplos a serem utilizados. Foram utilizados 3 *folds* para o treinamento dos algoritmos. Portanto, os exemplos foram divididos em 3 partes aproximadamente iguais, e uma por vez foi utilizada para testar, enquanto o restante foi utilizado para treinar, ou seja, foram utilizados dois terços para treinar e um terço para testar, sendo repetido o processo por três vezes, para que no final cada parte seja utilizada para teste. Essa maneira de treinar é conhecida como *threefold cross-validation* [8]. A taxa de acertos durante os testes do treinamento foram altas, com poucas variações entre os algoritmos, sendo que na maioria das vezes o algoritmo SMO obteve melhores taxas, porém a diferença dele para os outros algoritmos

foi muito pouca, o que dificulta afirmar qual o algoritmo que teve melhor índice de acerto, como pode ser visto na Tabela 3.

Tabela 3: Índices de Acertos dos algoritmos por categoria durante treinamento

Categorias	Acertos durante treinamento (cross-validation)		
	J4.8	Naïve Bayes	SMO
EXECUÇÃO	92,30%	93,90%	95%
PREVIDÊNCIA SOCIAL	97,30%	98,30%	98,20%
MÃO-DE-OBRA	92,91%	91,91%	93,21%
EMBARGOS DECLARATÓRIOS	99,20%	97,70%	98,50%
PROVA	90,90%	87,93%	94,50%
RELAÇÃO DE EMPREGO	93,50%	94,40%	97,60%
SINDICATO OU FEDERAÇÃO	97,10%	97,90%	97,40%
HONORÁRIOS	97,40%	97,30%	97,30%
NULIDADE PROCESSUAL	96,30%	92,10%	95,40%
RESPONSABILIDADE SOLIDÁRIA/SUBSIDIÁRIA	98%	97,70%	98,40%

A diferença principal entre os algoritmos foi a saída apresentada do modelo de aprendizado de cada algoritmo. O J4.8 como uma implementação de uma árvore de decisão, permitiu identificar facilmente termos (*stems*) relevantes para diferenciar uma categoria das demais.

Avaliação e Interpretação dos Resultados

Os modelos de aprendizado dos algoritmos foram salvos, e depois confrontados com exemplos dessas mesmas categorias, que são desconhecidos pelos modelos de aprendizado. Foram selecionados aleatoriamente, 5 exemplos de cada categoria treinada, porém exemplos de 2011, desconhecidos para os modelos aprendidos, totalizando 50 documentos a serem preditos. Assim, os classificadores binários, devidamente treinados, receberam 50 exemplos desconhecidos para realizarem a sua predição individual. A Tabela 4 demonstra a taxa de erro da categoria e a taxa de erro total do algoritmo. A taxa de erro da categoria demonstra a predição incorreta (falsos negativos) dentro dos exemplos verdadeiros do classificador de uma categoria, e a taxa de erro total demonstra a predição incorreta dentro de todos os exemplos a serem preditos (soma dos falsos negativos e falsos positivos).

O algoritmo Naive Bayes foi o algoritmo que obteve maior taxa de erro por categoria, ou seja, foi o algoritmo que menos conseguiu predizer verdadeiros positivos, chegando a predizer nenhum verdadeiro positivo na categoria “Execução”, obteve uma taxa de 60% de erros na categoria “Mão-de-obra”, e 20% nas categorias “Previdência social” e “Prova”. O algoritmo SMO apesar de apresentar taxas de erros somente em duas categorias, “Execução” e “Prova”, teve uma taxa de erro de 80% na categoria “Execução”. Já o algoritmo J4.8, apesar de apresentar taxas de erros nas categorias “Execução”, “Mão-de-obra” e “Prova”, todas as taxas foram inferiores a 40%.

Tabela 4: Taxa de erro por categoria e taxa de erro total

Categorias	J4.8		Naïve Bayes		SMO	
	Taxa de Erro da Categoria	Taxa de Erro Total	Taxa de Erro da Categoria	Taxa de Erro Total	Taxa de Erro da Categoria	Taxa de Erro Total
EMBARGOS DECLARATÓRIOS	0,00%	4,00%	0,00%	12,00%	0,00%	10,00%
EXECUÇÃO	40,00%	4,00%	100,00%	14,00%	80,00%	12,00%
HONORÁRIOS	0,00%	2,00%	0,00%	38,00%	0,00%	4,00%
MÃO-DE-OBRA	20,00%	10,00%	60,00%	18,00%	0,00%	8,00%
NULIDADE PROCESSUAL	0,00%	4,00%	0,00%	36,00%	0,00%	14,00%
PREVIDÊNCIA SOCIAL	0,00%	0,00%	20,00%	8,00%	0,00%	2,00%
PROVA	20,00%	14,00%	20,00%	28,00%	20,00%	12,00%
RELAÇÃO DE EMPREGO	0,00%	44,00%	0,00%	36,00%	0,00%	38,00%
RESPOSABILIDADE SOLIDÁRIA/SUBSIDIÁRIA	0,00%	12,00%	0,00%	78,00%	0,00%	10,00%
SINDICATO OU FEDERAÇÃO	0,00%	16,00%	0,00%	28,00%	0,00%	2,00%

Tabela 5: Acuidade dos algoritmos classificadores

Categorias	Acuidade Total		
	J4.8	Naïve Bayes	SMO
EMBARGOS DECLARATÓRIOS	96,00%	88,00%	90,00%
EXECUÇÃO	96,00%	86,00%	92,00%
HONORÁRIOS	98,00%	62,00%	88,00%
MÃO-DE-OBRA	86,00%	78,00%	96,00%
NULIDADE PROCESSUAL	96,00%	64,00%	78,00%
PREVIDÊNCIA SOCIAL	100,00%	92,00%	100,00%
PROVA	86,00%	72,00%	54,00%
RELAÇÃO DE EMPREGO	56,00%	64,00%	68,00%
RESPONSABILIDADE SOLIDÁRIA/SUBSIDIÁRIA	88,00%	22,00%	60,00%
SINDICATO OU FEDERAÇÃO	84,00%	72,00%	82,00%

A taxa de erro total esteve presente em todas as categorias para os classificadores Naive Bayes e SMO. O algoritmo J.48 foi o que apresentou menor taxa de erro total. O comitê classificador conseguiu anular a taxa de erro total apresentada pelos algoritmos Naive Bayes e SMO na categoria “Previdência Social”, porém teve taxas de erro total maiores que o algoritmo J4.8 nas categorias “Responsabilidade Solidária/Subsidiária” e “Sindicato ou Federação”. A acuidade total individual (verdadeiros positivos, somados aos verdadeiros negativos) de cada um dos algoritmos classificadores, após a predição dos 50 exemplos desconhecidos, pode ser analisada através da Tabela 5. É possível notar que não há um classificador que obteve uma acuidade maior em todas as categorias. Na maioria das categorias o algoritmo J4.8 teve maior acuidade em relação aos algoritmos Naive Bayes e SMO, mas por uma diferença pequena de menos de 10%. O algoritmo SMO foi superior ao algoritmo J4.8 para as categorias “Embargos Declaratórios” e “Honorários”, mas também por um diferença de no máximo 10%.

Conclusão

O processamento aplicando as implementações dos algoritmos J4.8, Naive Bayes e Support Vector Machines (SMO – Sequential Minimal Optimization) obtiveram excelente desempenho durante o treinamento dos modelos de aprendizagem. Todavia, não é possível afirmar qual o melhor, pois a diferença entre eles foi mínima. Os testes de predição demonstraram que apesar de não apresentarem muita diferença durante o treinamento, durante a predição os resultados obtidos pelos algoritmos foram bem distintos, onde o algoritmo Naive Bayes obteve a pior desempenho e o J4.8 obteve melhor desempenho quanto a acuidade total em todas as categorias, exceto a categoria “Relação de Emprego” onde o Support Vector Machine obteve o maior desempenho.

Os resultados diversificados encontrados principalmente na taxa de acuidade total, onde os algoritmos por vezes se alternam com melhores resultados dependendo da categoria, ocorrem devido ao fato da linguagem jurídica utilizar os mesmos termos, principalmente em latim, dentro de categorias diferentes, dificultando a predição de algoritmos puramente probabilísticos como o Naive Bayes. Termos com significados diferentes dependendo do contexto em que são utilizados também são comuns na linguagem jurídica, o que dificulta a predição de algoritmos analíticos e algébricos como J4.8 e SMO. Seria necessária a realização de um mapa semântico para auxiliar estes algoritmos, pois nesta pesquisa foi utilizada apenas a abordagem estatística, levando em consideração apenas a frequencia dos termos encontrados nos textos.

Apesar dos resultados não demonstrarem qual o melhor algoritmo a ser utilizado para classificar as ementas, a variação da acuidade total entre os algoritmos classificadores, levanta a hipótese de que a combinação dos resultados dos três algoritmos pode ser utilizada para obter o melhor de cada classificador, trazendo resultados com menor taxa de erro e melhor acuidade.

Referências

[1] KONCHADY, M. Text Mining Application Programming: Charles River Media, 2006. ISBN 1-58450-460-9.

[2] EBECKEN, N. F. F.; LOPES, M. C. S.; COSTA, M. C. A. (2003), Mineração de Textos. *In*: REZENDE, S. O. Sistemas Inteligentes: fundamentos e aplicações. Barueri, SP: Manole.

[3] FELDMAN, R; SANGER, J. (2007), The Text Mining Handbook: Cambridge University Press.

[4] GONÇALVES, L. S. M.; REZENDE, S. O. (2002), Categorização em Text Mining. Disponivel em: <http://www.icmc.usp.br/~std-cd/Artigos/Computacao/IC/LeaSilviaMG.pdf.> Acesso em: 05/09/2007

[5] SOARES, M. V. B. (2009), Aprendizado de máquina parcialmente supervisionado multidescrição para realimentação de relevância em recuperação de informação para a WEB, Dissertação de Mestrado, ICMC / USP, São Paulo, 95p.

[6] RUSSELL, S.; NORVIG, P. (2004), Inteligência Artificial: trad. da 2ª ed. Rio de Janeiro: Elsevier.

[7] WITTEN, I. H.; FRANK E. (2000), Data Mining – Pratical Machine Learning Tools e Tecnhiques with JAVA Implementations: Morgan Kaufmann.

[8] PARK, A. F. M I. (2010), Aplicação de Técnicas de Mineração de Textos para categorização de eventos de Segurança no CITR Gov. , Dissertação de Mestrado, UnB, Brasília, 82p.

Contato

Thiago Ferauche, professor da Universidade Católica de Santos e Agente de Tecnologia da Informação do Tribunal Regional do Trabalho 2ª Região – SP.
E-mail: thiago.ferauche@gmail.com