

Classificadores de Texto Reduzido Baseados em SVM

Eduardo Akira Kinto
eakinto@yahoo.com

Laboratório de Sistemas Integrado – *Departamento de Engenharia e Sistemas Eletrônicos*
Escola Politécnica da Universidade de São Paulo
Av. Prof. Luciano Gualberto, trav.3, n.158 - 05508 900 – São Paulo – SP – Brazil

Emílio Del Moral Hernandez
emilio_del_moral@ieee.org

Sumário

Este artigo apresenta estudos no campo da recuperação da informação e classificação de textos reduzidos, ou seja, textos que apresentam poucas palavras (frases). Tradicionalmente os estudos em classificação de textos têm sido feitos com textos ou documentos que apresentam muitas palavras. Este trabalho visa explorar a capacidade da classificação de frases, usando-se Máquinas de Vetores Suporte, situação comum em muitas aplicações como, por exemplo, filtro do título de mensagens de e-mail e classificação de resumos de descrições detalhadas. O trabalho foi conduzido usando-se de documentos em língua portuguesa coletados da internet.

1. Introdução

Atualmente, é comum possuir-se uma base de dados contendo milhares de documentos digitalizados. Por exemplo, dados catalográficos de uma biblioteca. No entanto, a ausência de uma forma eficiente de localizar as informações torna a base de dados virtualmente inútil. A recuperação da informação (IR do inglês, *Information Retrieval*) [8] visa apresentar formas apropriadas para a localização precisa e rápida de informações armazenadas em uma base de dados.

Essa quantidade gigantesca de informação deve ser classificada (TC do inglês, *Text Categorization*) [7], [9] para permitir a implementação de alguma forma eficiente de IR. IR e TC são campos de pesquisas correlatos, ou seja, muitas práticas usadas em IR são também usadas em TC. Um sistema de classificação de texto consiste nas seguintes etapas:

- **1) coleta da informação**
- **2) seleção do termo:** nesta fase os termos mais relevantes para fins de classificação são identificados
- **3) atribuição de peso ao termo:** um peso é atribuído para o termo selecionado

- **4) aprendizado do classificador:** usam-se os textos de treino representados através de seus pesos para gerar um classificador. Esse processo envolve uma atividade de aprendizado de máquina.
- **5) classificação**

A coleta da informação (documentos e textos) se faz através de programas (*crawlers*) que navegam pela internet obtendo as informações desejadas.

Seleção do termo e atribuição de peso ao termo compõe o pré-processamento da informação. Ele é o responsável pela remoção das palavras que não contribuem para o aprendizado do classificador e para a classificação (categorização). Maiores detalhes serão apresentados nas próximas páginas.

Como exemplo de sistema de IR pode-se citar o site de busca Google, o site mais acessado em sua categoria. Com uma base de dados com mais de 8 bilhões de documentos totalizando mais de 1 terabyte de dados seria impossível manter seu conteúdo atualizado sem algum mecanismo eficiente de armazenamento da informação. Assim a categorização dos documentos é um dos mecanismos que auxiliam na busca da informação.

Se por um lado o volume reduzido de palavras por documento é algo positivo do ponto de vista do processamento da informação, por outro lado ele torna o processo de classificação mais complexo porque técnicas tradicionais da IR utilizam-se muito da frequência da ocorrência das palavras dentro de um documento.

Este trabalho irá apresentar os resultados da aplicação do SVM para fins de classificação de frases (adquiridas na internet), buscando a melhor função Kernel e algumas técnicas de pré-processamento da informação. SVM foi escolhido entre outras técnicas existentes porque ele tem apresentado bom desempenho [1], [6], [7] (taxa de acerto em um classificador e capacidade de generalização).

Tradicionalmente o aprendizado de máquina entra apenas nas etapas 2 (seleção do termo) e 4 (aprendizado do classificador). A proposta deste trabalho é analisar o

uso do aprendizado de máquina para a etapa 3 também, conforme feito em [14].

Um outro ponto importante a ressaltar neste trabalho é o uso de diversas funções para atribuição de pesos às palavras de um texto. Geralmente usam-se apenas o TFIDF. Os detalhes serão apresentados nos próximos itens.

Este artigo está organizado da seguinte forma: A seção 2 apresenta conceitos a respeito da IR e TC, a seção 3 apresenta o SVM, a seção 4 apresenta a metodologia de trabalho, a seção 5 apresenta os resultados obtidos, a seção 6 apresenta as considerações finais do trabalho e algumas propostas para futuros trabalhos, finalmente a seção 7 apresenta a bibliografia consultada.

2. Recuperação da informação e Classificação de Texto

A recuperação da Informação visa extrair informações armazenadas em algum repositório de dados segundo critérios que melhor atendam as expectativas do usuário solicitante. Desta forma todo documento “bruto” deve ser pré-processado para que se possa identificar os elementos-chave que o tornam elegível para aparecer no resultado de uma busca (consulta ou *query*).

A classificação de texto consiste em categorizar as frases em uma ou mais categorias existentes. No presente trabalho foi estabelecido que um documento pode pertencer a apenas uma categoria. A seguir serão apresentadas as etapas que compõem um sistema de RI/TC.

2.1 Coleta das informações

A coleta dos documentos (informações) é geralmente feita através do uso dos robos chamados de *crawlers*, que nada mais são do que programas que varrem a *internet* baixando para sua base todas as páginas de seu interesse. Esse processo é repetido periodicamente para manter a base de dados atualizada. No presente trabalho foi utilizado o robo desenvolvido pela empresa BuscaPé denominada de Spider. Ele é um robo especializado em varrer a *internet* buscando os *sites* de *e-commerce* (empresas que vendem produtos pela *internet*).

Desta forma os documentos obtidos correspondem aos *sites* que realizam vendas pela *internet* (brasileira). Deste ponto em diante, um documento (texto) refere-se à descrição textual de um produto que está a venda pela *internet*, capturada pela Spider.

Exemplo: *Monitor 15POL CRT 105S59 / 48 Marfim Philips Com resolução máxima em até 1024 x 768, este monitor de 15" da Philips conta com compatibilidade total com PC e MAC e tem instalação simples e rápida. (Cod:108693)*

2.2 Pré-processamento das informações

O pré-processamento consiste na remoção das informações que não trazem nenhum benefício para o processo da IR e TC; é a transformação de um conjunto de documentos não estruturados em arquivos para uso do sistema da IR e TC. O pré-processamento também é a fase onde se realça os elementos-chave para construir um bom sistema de IR e TC.

Existem diversos *sites* de busca como Google, Yahoo, AltaVista e Cadê entre outros. Todos fazem buscas por *sites* da *internet* como resultado de uma consulta feita pelo usuário, no entanto existem diferenças fundamentais entre eles resumidas em:

- Velocidade da busca: é uma consequência de um sistema de IR bem otimizado, estruturado e planejado.
- Relevância dos resultados: o resultado de uma pesquisa deve, na medida do possível, atender as expectativas daqueles que buscam a informação, quanto mais próximo estiver o resultado apresentado do que se desejava melhor será o sistema.

O pré-processamento das informações foi realizado neste trabalho em duas etapas, sendo que na primeira etapa removeram-se algumas palavras (stop-words) muito comuns na língua portuguesa como as preposições (mas, de, com, sem, para), os artigos (a, o) e os conectores (e). Na segunda etapa foram removidas outras palavras (também são chamados de stop-words) baseadas na lei de Zipf [2] que diz que existem poucas palavras que ocorrem com frequência e muitas palavras que não são usadas com frequência, ou seja, removendo-se as palavras que ocorrem com muita frequência pode-se reduzir muito as informações não importantes para o contexto da IR e TC.

A tabela 1 abaixo apresenta as 15 palavras com maior frequência de ocorrência dentro da categoria de alimentos e bebidas. Nota-se claramente que algumas destas palavras podem aparecer em qualquer descrição de produto (neste artigo uma descrição de produto ou frase é um documento). $r \times Pr$ representa o produto do *ranking* de cada palavra pela sua probabilidade de ocorrência, ela torna-se quase constante após a 11ª palavra conforme esperado (lei de Zipf).

Tabela 1. Verificação da lei de Zipf para produtos normalizados, ranking de 1 a 5, e 11 a 15.

Palavra	Frequência	$r \times Pr$	Palavra	Frequência	$r \times Pr$
1.Pct	5317	0.064	11. nestle	2713	0.359
2.500g	4863	0.117	12. morango	2464	0.355
3.livro	4015	0.144	13. tinto	2459	0.385
4.light	3723	0.179	14. branco	2366	0.398
5.750ml	3707	0.223	15. 400g	2181	0.393

Removendo-se as palavras não relevantes (obtidas

durante o processo de seleção do termo) com alto índice de ocorrência consegue-se diminuir bem a dimensão do vetor de índice usado no processo de mapeamento da palavra para números que alimentarão a máquina de aprendizado.

Uma vez removidos os *stop-words* foi realizado um levantamento da importância de cada palavra (cálculo do seu peso) no contexto da classificação de texto (em inglês, *feature selection*). Para tanto foi calculada a frequência com que uma palavra aparece em uma categoria (TFc – do inglês *Term Frequency*) e a frequência inversa da ocorrência do termo (IDFc - do inglês, *Inverse Document Frequency*). A letra “c” das duas siglas anteriores refere-se à categoria.

Tradicionalmente [3], [9] TF e IDF são calculados como:

- TF, é a frequência da ocorrência de uma palavra em um único documento
- IDF, veja a equação 1 a seguir, mede quão rara é uma palavra e por este motivo ela é mais importante

$$IDF(w) = -\log_2 \frac{df_w}{D} \quad (\text{Eq. 1})$$

Onde,

df_w é a frequência da ocorrência da palavra w em documentos distintos

D é a quantidade de documentos

A nova métrica de peso proposta (TFc, IDFc, TFIDFc) neste trabalho pode ser expressa pelas equações 2, 3 e 4 abaixo.

$$TF_c(w) = \sum w_c \quad (\text{Eq. 2})$$

Na equação 2, TF_c é a soma de todas as ocorrências da palavra w em uma categoria c não contando as repetições dentro da mesma frase (devido ao baixo índice de repetição de palavras em textos reduzidos).

$$IDF_r(w) = -\log_2 \frac{cf_w}{C} \quad (\text{Eq. 3})$$

Onde,

cf_w , representa a quantidade de vezes em que uma palavra w apareceu em uma categoria

C , representa o total de categorias

TFc é uma adaptação do TF que calcula a quantidade de ocorrências de uma palavra em uma categoria, não considerando as repetições em uma mesma frase. IDFc é uma adaptação do IDF para permitir seu cálculo em situação com poucas palavras por documento. Ele também visa medir a “raridade” de uma palavra.

Foi também usada uma combinação do TFc e IDFc resultando em TFIDFc expressa pela equação 4 abaixo.

$$TFIDF_c(w) = \left(\sum w_c \right) \cdot \left(-\log_2 \frac{cf_w}{C} \right) \quad (\text{Eq. 4})$$

Foram usadas mais duas técnicas tradicionais no campo da IR e TC, a Informação Mútua (MI – do inglês, *Mutual Information* veja equação 5) e o Chi-quadrado conforme apresentado na equação 6 [4], [11]. As duas medem o grau de associação entre palavra e categoria. Em outras palavras elas medem se uma palavra é ou não importante para uma determinada categoria.

$$MI(w, c) = \log_2 \frac{\Pr(w \wedge c)}{\Pr(w) \times \Pr(c)} \quad (\text{Eq. 5})$$

Quando o termo w e a classe c tiverem alguma associação, ou seja, os eventos são dependentes, então a probabilidade $\Pr(w \wedge c)$ será maior que $\Pr(w) \times \Pr(c)$ assim a informação mútua será muito maior do que zero. Por outro lado se os eventos w e c forem independentes então $\Pr(w \wedge c) = \Pr(w) \times \Pr(c)$ assim a informação mútua será zero.

Abaixo encontra-se a expressão para o cálculo do Chi-quadrado.

$$\chi^2(w, c) = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)} \quad (\text{Eq. 6})$$

Onde:

A é a quantidade de vezes em que w e c ocorreram

B é a quantidade de vezes em que w ocorreu sem c

C é a quantidade de vezes em que c ocorreu sem w

D é a quantidade de vezes em que nem w nem c ocorreram

N é a quantidade total de texto

A fase de pré-processamento transforma, conforme comentado no início desta seção, um documento não estruturado em uma outra forma para facilitar seu uso pelo sistema. Deve-se lembrar que a quantidade de documentos é enorme, assim a quantidade da informação envolvida também é enorme. Uma otimização nesta etapa significa em alta performance computacional.

Geralmente as palavras que compõem um documento são indexadas segundo um vetor de palavras onde cada palavra traz consigo um valor (peso) que representa a sua importância para IR e TC. TFc, IDFc, TFIDFc, MI e χ^2 visam reduzir a dimensão (soma de todas as palavras distintas de todos os documentos) do vetor de palavras, selecionando-se apenas as mais relevantes.

Neste trabalho usou-se as mesmas funções empregadas no *feature selection* para atribuição de peso a cada palavra do *feature vector*. Normalmente usa-se apenas o TFIDF.

3. Máquinas de Vetores Suporte

As Máquinas de Vetores Suporte (SVM - do inglês, *Support Vector Machine*) são técnicas de aprendizado de máquina (ML - do inglês *Machine Learning*) do tipo aprendizado supervisionado, ou seja, fornecem-se elementos para que o programa aprenda por meio de treinamento [5], [10], [11].

O SVM tem como sua principal característica uma boa capacidade de generalização e suporte a dados de grande dimensão. O fundamento do SVM está na teoria de aprendizado estatístico de Vladimir Vapnik. Em poucas palavras pode-se dizer que ele implementa um classificador linear cuja função discriminante geral é da forma:

$$f(x, w, b) = \text{sign}(w \cdot x - b) \quad (\text{Eq. 7})$$

Onde:

w , é o vetor de peso associado a cada elemento do vetor suporte

x , são os elementos (amostras) do vetor suporte

Usando-se da equação anterior separam-se dois espaços, um positivo e outro negativo cujas fronteiras são os vetores suporte, pontos da extremidade da margem que servem como fronteira da maximização da mesma.

Assim o objetivo do SVM é maximizar a margem M (definida a seguir) de separação entre os elementos positivos e negativos.

Dado que:

- o hiperplano positivo é separado pela reta, $\{x^+ : w \cdot x^+ + b = +1\}$
- o hiperplano negativo é separado pela reta, $\{x^- : w \cdot x^- + b = -1\}$
- $x^+ = x^- + \lambda w$

Pode-se calcular a margem M em função de w e b , (para maiores detalhes consulte [11])

$$M = \frac{2}{\sqrt{w \cdot w}} \quad (\text{Eq. 8})$$

Resolver um problema através do SVM resume-se então a:

Minimizar uma função objetiva

$$f : f = \frac{2}{\sqrt{w \cdot w}} \quad (\text{Eq. 9})$$

Com R restrições da forma

$$\begin{aligned} w \cdot x + b &\geq +1, & \text{se } y_k &= +1 \\ w \cdot x + b &\leq -1, & \text{se } y_k &= -1 \end{aligned} \quad (\text{Eq. 10})$$

onde R representa o espaço amostral do problema, e a função f é um função de dimensão m .

Um SVM pode trabalhar com classes linearmente não separáveis através do uso de funções de Kernel. Elas realizam uma transformação dos elementos do espaço de entrada (as amostras de treino/teste) para um novo espaço denominado de espaço das características. Desta forma pode-se utilizar um SVM linear para problemas mais complexos.

Uma função de Kernel recebe como entrada dois pontos X_i e X_j do espaço das entradas e calcula o produto $\rho(X_i) \cdot \rho(X_j)$ do espaço das características. Em geral a função de mapeamento ρ é muito mais complexa do que a função de Kernel (que é o resultado do produto das funções de mapeamento), desta forma é comum definir-se a função Kernel ao invés da função de mapeamento.

Algumas das funções de Kernel mais usadas são, lineares, RBF (gaussianas), Redes Neurais Artificiais simples (sigmóidais) e as funções polinomiais.

A escolha da função de Kernel bem como seus parâmetros constitui a principal dificuldade no uso de uma SVM, a sua escolha afeta diretamente o resultado de um classificador. Na próxima seção serão apresentados alguns resultados para a escolha da função de Kernel deste trabalho.

Finalmente, os SVM foram originalmente propostos para resolução de problemas binários. Existem diversas abordagens para a criação de um SVM multi-classe. Seu grau de complexidade e principalmente o seu custo computacional é muito maior do que um classificador binário. Assim formas alternativas que usam o SVM tradicional compõem a maioria das implementações de SVM multi-classe. Podem-se citar duas formas muito usadas: a um-contra-um e a um contra todos [13].

4. Metodologia de pesquisas

O sistema proposto combina métodos algorítmicos (estrutura de dados) e aritméticos (SVM) visando a implementação de um sistema de auxílio à classificação de documentos (páginas da *internet*). A estrutura do sistema proposto é ilustrado pela figura 1 abaixo.

Existem nesta estrutura basicamente 3 grandes módulos:

- módulo de captura: ele é responsável pela captura das informações (páginas web), armazenando seu conteúdo em arquivos e banco de dados
- módulo de pré-processamento: nesta etapa é realizada a filtragem das palavras irrelevantes, normalização das mesmas (em inglês, *word stemming*), e atribuição de pesos para as palavras segundo seu TFIDFc, MI e χ^2 .
- módulo de classificação: esta etapa é dividida em pré-categorização e pós-categorização, a primeira usa-se de métodos algorítmicos e a segunda de método aritméticos.

A pré-categorização visa ser uma etapa rápida e simples (*ranking* das categorias mais prováveis usando os

pesos TFIDFc, MI e χ^2). Ela consiste em classificar as páginas segundo 4 principais categorias. O pós-classificador visa refinar o pré-classificador usando-se de métodos aritméticos; tendo o número de categorias a analisar reduzido, pode-se empregar o SVM de forma eficiente e rápida.

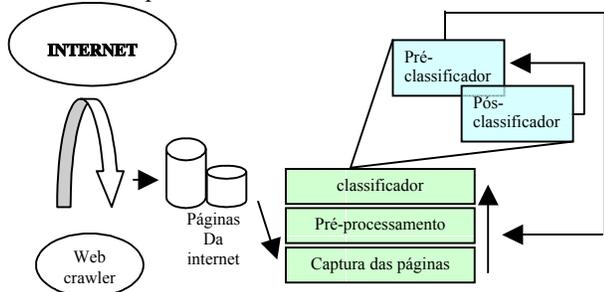


Figura 1. Estrutura do sistema classificador.

Um outro motivo para a existência da etapa do pré-classificador está na impossibilidade do SVM em tratar categorias com um número muito reduzido de documentos. A estrutura da figura 1 foi elaborada tendo em vista que o SVM necessita de categorias uniformes. No entanto, isso nem sempre é verdadeiro, podendo haver categorias com centenas de documentos e categorias com 3 ou 4 documentos. Desta forma o categorizador ilustrado na figura 1 pode ser expandido para a figura 2 abaixo.

A realimentação da etapa do pós-classificador para a etapa do pré-classificador existe para bloquear classificações sucessivas erradas. SVM pode ser utilizado quando se tem documento em grandes quantidades para treino e teste para cada categoria. Existem, no entanto situações onde não se têm um número aceitável de documentos em uma categoria. Nesse caso ao invés de se utilizar o SVM (pós-classificador) sugere-se o uso de um classificador baseado nos pesos (método de Rochio, 1971) de cada palavra do vetor de palavras.

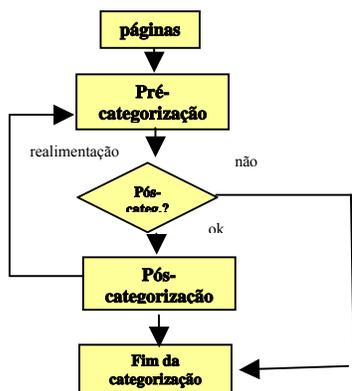


Figura 2. Estrutura expandida do módulo categorizador.

5. Resultados experimentais

A tabela abaixo apresenta os resultados de testes realizados para a determinação da melhor Kernel Function. Foram selecionadas duas funções (lineares e RBFs) após testes preliminares com as outras (polinomiais e sigmoidal) cujos resultados não foram bons.

A tabela 2 mostra que uma função de base radial com largura de gaussiana G de 0.0009 apresenta o melhor índice de acerto. Quanto menor o valor de G maior será a largura do *Kernel* RBF, assim será necessário menos vetores suporte. Para o teste foram escolhidas duas categorias sem eliminação dos stop-words, usando-se TFIDFc como peso das palavras. As funções de avaliação de termo MI e χ^2 , não apresentadas neste trabalho, também mostraram que RBF é superior do que uma função linear. Os valores de G obtidos foram diferentes dos valores para TFIDFc.

Na tabela 2 a coluna *Precision* e a coluna *Recall* representam:

Precision (ou precisão) representa a porcentagem de classificação correta em relação a tudo que foi classificado pelo sistema.

Recall (ou cobertura) representa a porcentagem de classificação correta em relação a tudo que deveria ser classificado corretamente pelo sistema.

Tabela 2. Testes para determinar a melhor Kernel Function.

	Tempo gasto ms	Iterações do programa	Recall	Precision
Modelo linear	1735,35	63992	17,68	17,70
Rbf G=0.1	1015,07	11451	57,90	57,38
Rbf G=0.5	1093,17	10853	47,96	47,75
Rbf G=0.05	1184,78	11125	62,17	61,41
Rbf G=0.005	2121,11	8830	73,80	72,65
Rbf G=0.0001	1308,45	9623	67,56	67,28
Rbf G=0.0005	1868,81	10813	73,40	72,69
Rbf G=0.0009	1727,33	9565	74,19	73,56

A seguir foi realizado experimento para determinar a quantidade de palavras necessárias no vetor de palavras, foram definidos os limiares apresentados na tabela 3 abaixo referentes ao TFC.

Tabela 3. Classificação analisada segundo variação do limiar das palavras. RBF, G=0.0009.

	Tempo gasto (segundos)	Recall	Precision	F-measure
MI	353.43	84.79	81.40	83.06
χ^2	346.22	81.43	82.04	81.73
TFIDF	1264.9	78.03	78.94	78.48

A tabela 3 apresenta uma comparação dos resultados obtidos na classificação de um mesmo documento. Utilizou-se 3 métodos diferentes para atribuição de peso a cada palavra do *feature vector* e como critério de escolha dos mesmos (*features selection*). Os valores apresentados

na tabela 3 foram obtidos aumentando-se o tamanho do *feature vector* gradualmente até chegar a um ponto em que não há melhoria no *F-measure*.

Um outro resultado importante está na tabela 4 onde foram realizados testes para verificar a quantidade máxima de palavras a considerar no vetor de palavras. Por exemplo, “tv 28 polegadas” é um 3-gram, enquanto que “tv 28” é um 2-gram.

Pode-se ver pelo resultado obtido que o melhor a considerar são *features* de tamanho 1 (1-gram), este resultado é muito importante porque quanto maior o grau do *gram*, maior o *feature vector* e maior será o processamento exigido no SVM.

Tabela 4. Análise do “grams”.

RBF G=0.0009	Tempo gasto ms	Iterações do programa	Recall	Precision
1-gram	362.33	2227	82.81	83.80
2-gram	521.41	2217	76.80	81.02
3-gram	654.65	2422	74.06	79.01
4-gram	654.05	2422	74.06	79.01

Os testes realizados para determinar os valores da tabela 4 foram feitos usando-se como função de seleção e peso o TFIDFc. A estrutura apresentada na figura 2 permitiu o tratamento de todas as categorias, assim ela se mostrou bem robusta.

6. Conclusão

Foi explorada neste trabalho a possibilidade de classificar frases usando-se do SVM. Um aspecto importante que deve ser ressaltado se refere à fase de pré-processamento das frases, onde foram atribuídos pesos segundo TFIDFc, MI e Chi-quadrado para cada palavra.

Usualmente trabalha-se com classificação de documentos, no entanto, quanto menor o documento, menor será o processamento exigido pela máquina e menor será o seu custo de armazenamento. Assim a adaptação do TFIDFc proposto neste trabalho mostrou ser funcional e eficiente para o tratamento de frases com poucas palavras.

Os resultados obtidos indicam que uma função Kernel do tipo RBF é aquela que melhor se adapta para a classificação. Um outro resultado obtido se relaciona com as relações semânticas entre as palavras. Por exemplo, o uso de *features* de 3-gram (“tv 50 pol”, por exemplo) não apresenta resultados melhores daqueles obtidos quando se usa *features* de 1-gram (“tv”), ou seja, este resultado sugere eliminar-se as combinações entre palavras do *feature vector*, diminuindo seu tamanho.

Um outro resultado obtido mostra que pode-se usar MI como *feature selection* e como *feature weighting*. Ele superou o Chi-quadrado e o TFIDF que normalmente apresentam resultados melhores do que o MI em aplicações de textos extensos.

Futuros trabalhos nesta área visam a exploração de técnicas alternativas para a fase de pré-processamento.

7. Bibliografia

- [1] Burges, C. J. C., “**A Tutorial on Support Vector Machines for Pattern Recognition**”, Data Mining and Knowledge Discovery, v.2, n.2, p.121-167, 1998
- [2] Carlos M Urzua, “**A simple and efficient test for Zipf’s Law**”, Economics Letters, v.66, p.257-260, 2000
- [3] Church, K. e Gale, W., “**Inverse Document Frequency (IDF): A Measure of Deviation from Poisson**”, *Proceedings of the Third Workshop on Very Large Corpora*, p. 121-130, 1995
- [4] Church, K. W. e Hanks, P., “**Word Association Norms, Mutual Information, and Lexicography**”, *Proceedings of the 27th. Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, p.76-83, 1989
- [5] Cortes, C. e Vapnik, V., “**Support Vector Networks**”, *Machine Learning*, v.20, p.273-297, 1995
- [6] Dumais, S. T. e Platt, J. e Heckerman, D. and Sahami, M., “**Inductive learning algorithms and representations for text categorization**”, *Proceedings of the Seventh International Conference on Information and Knowledge Management*, 1998
- [7] Joachims, T., “**Text Categorization with Support Vector Machines: Learning with Many Relevant Features**”, *Proceedings of ECML, 10th European Conference on Machine Learning*, p.137-142, 1998
- [8] Mendelzon, A. e Mihaila, G. A. e Milo, T., “**Querying the World Wide Web**”, *International Journal on Digital Libraries*, 1997
- [9] SEBASTIANI, F. “**Machine learning in Automated Text Categorization**”, Technical report, IEI B4-31-12-99, Instituto di Elaborazione della Informazione, Consiglio Nazionale delle Ricerche, 1999
- [10] Vapnik, V., “**The Nature of Statistical Learning Theory**”, Springer 1995
- [11] Yang, Y. e Pedersen, J. O., “**Comparative Study on Feature Selection in Text Categorization**”, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, 1997
- [12] Cristianini, N. e Shawe-Taylor, J., “**An Introduction to Support Vector Machines**”, Cambridge University Press, 2000
- [13] Chih-Wei Hsu e Chih-Jen Lin, “**A Comparison of Methods for Multi-class Support Vector Machines**”, Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2001.
- [14] Debole, F. e Sebastiani, F. “**Supervised Term Weighting for Automated Text Categorization**”, *Proceedings of SAC-03, 18th ACM Symposium on Applied Computing*, 2002