PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Programa de Pós-Gradução em Informática

FILTRO DE CONTEÚDO PARA SISTEMAS SMS BASEADO EM CLASSIFICADOR BAYESIANO E AGRUPAMENTO POR PALAVRAS

Dirceu Otávio Vieira Belém

Belo Horizonte
2009

Dirceu Otávio Vieira Belém

FILTRO DE CONTEÚDO PARA SISTEMAS SMS BASEADO EM CLASSIFICADOR BAYESIANO E AGRUPAMENTO POR PALAVRAS

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para obtenção do título de Mestre em Informática.

Orientadora: Fátima de Lima Procópio Duarte Figueiredo

Belo Horizonte
2009

FICHA CATALOGRÁFICA

Elaborada pela Biblioteca da Pontifícia Universidade Católica de Minas Gerais

Belém, Dirceu Otávio Vieira
B428f Filtro de conteúdo para sistemas SMS baseado em classificador bayesiano e agrupamento por palavras / Dirceu Otávio Vieira Belém. – Belo Horizonte, 2009.

81f. : il.

Orientadora: Fátima de Lima Procópio Duarte Figueiredo.
Dissertação (Mestrado) – Pontifícia Universidade Católica de Minas Gerais. Programa de Pós-graduação em Informática.
Bibliografia.

1. Telefonia celular – Teses. 2. Mensagens eletrônicas não solicitadas. 3. Redes de computadores. I. Figueiredo, Fátima de Lima Procópio Duarte. II. Pontifícia Universidade. Católica de Minas Gerais. III. Título.

CDU: 681.3.01:621.39

Bibliotecário: Fernando A. Dias – CRB6/1084

PUC Minas
Programa de Pós-graduação em Informática

FOLHA DE APROVAÇÃO

Filtro de Conteúdo para Sistemas SMS Baseado em Classificador Bayesiano e Agrupamento por Palavra

DIRCEU OTÁVIO VIEIRA BELÉM

Dissertação defendida e aprovada pela seguinte banca examinadora:

Profa. Fátima de Lima Procópio Duarte Figueiredo - Orientadora (PUC Minas)
Doutora em Ciência da Computação - UFMG

Prof. José Marcos da Silva Nogueira - (UFMG)
Doutor em Engenharia Elétrica - Unicamp

Prof. Luis Enrique Zárate Gálvez - (PUC Minas)
Doutor em Engenharia Metalúrgica de Minas - UFMG

Belo Horizonte, 21 de dezembro de 2009.

Anel Rodoviário – KM 23,5, Rua Walter Ianni, 255 – São Gabriel / Belo Horizonte – MG / CEP: 31980-110
Fone: (31) 3439-5204 – Fax: (31) 3439-5204 – e-mail: mestradoinfsec@pucminas.br

Dedico,

À minha irmã Ana Cristina, referência de sucesso na família a qual tento me espelhar.

Aos meus pais Humberto e Luisa pela força e incentivo contínuo.

Ao meu irmão e amigo Junior, que esteve sempre presente.

Ao meu afilhado Diego, que ilumina nossas vidas.

A minha amada namorada Edilaine, pelo incentivo e apoio fundamental.

AGRADECIMENTOS

Agradeço a Deus, pela oportunidade, pela vida, e por sempre iluminar meu caminho.

A minha orientadora Professora Fátima de Lima Procópio Duarte Figueiredo, pela paciência e por ter sempre acreditado em mim e no projeto, mesmo com tantos obstáculos.

Ao Professor Luis Enrique Zárate pelas referências sobre Classificador Bayesiano.

Aos meus irmãos Ana Cristina e Junior, que sempre me apoiaram em todas as decisões e projetos da minha vida.

Aos amigos Patrício Souza, Hugo Leonardo e Ricardo Ferreira, pelas dicas e opiniões sobre melhorias.

À Aorta Entretenimento, pelo espaço para apresentações do projeto e ensaio da dissertação, e ao Franklin Valadares pelas críticas e dicas totalmente pertinentes sobre o assunto.

A todos os meus professores e mestres de trabalho que me ajudaram a ser o que sou profissionalmente e intelectualmente.

E a todos que contribuíram diretamente ou indiretamente para o término desta pesquisa.

“O mestre que caminha à sombra do templo, rodeado de discípulos, não dá, de sua sabedoria, mas sim de sua fé e de sua ternura. Se ele for verdadeiramente sábio, não vos convidará a entrar na mansão de seu saber, mas vos conduzirá antes ao limiar de sua própria mente. O astrônomo poderá falar-vos de sua compreensão do espaço, mas não vos poderá dar a sua compreensão. Porque a visão de um homem não empresta suas asas a outro homem. E assim como cada um de vós se mantém isolado na consciência de Deus, assim cada um deve ter sua própria compreensão de Deus e sua própria interpretação de coisas da terra.”

Texto de Khalil Gibran em “O Profeta”.

RESUMO

Existem muitas pesquisas sobre filtro de spam para e-mails. No entanto, existem poucos trabalhos que abordam o assunto para sistemas SMS (*Short Message Service*). Essa diferença vem da dificuldade de acesso às plataformas SMS das operadoras de telefonia celular. A quantidade de spam’s para sistemas SMS tem aumentado a cada ano.

O principal objetivo deste trabalho foi propor a implementação de um filtro de conteúdo para sistemas SMS baseado em Classificador Bayesiano e agrupamento por palavras. Para avaliar o desempenho do filtro, foram utilizadas 120.000 mensagens de um provedor de conteúdo que presta serviço para operadoras de telefonia celular. Os resultados apresentados demonstraram que o filtro proposto apresentou um índice de acerto, na detecção de spam’s, próximo de 100%.

Palavras-chave: Filtro de Conteúdo para Spam. Sistemas SMS. Classificador Bayesiano

ABSTRACT

There are many researches about e-mail spam filters. However, there are few researches that look at this issue for SMS (Short Message Service) systems. This is a result of the difficulty in having access to SMS platforms of mobile operators. Furthermore, the volume of spams to SMS systems has increased year after year.

The main objective of this work was to propose the implementation of a content filter for SMS systems based on the Bayesian Classifier and word grouping. In order to evaluate the performance of this filter, 120 thousand messages sent from a content provider that services mobile operators were tested. The results demonstrated that the proposed filter reached a correct index spam detection close to 100%.

Key-words: Content Filter for SMS Systems. SMS Systems. Bayesian Classifier.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

SMS	*Short Message Service*
SMSC	*Short Message Service Center*
MSC	*Mobile Switching Center*
SMSG	*Short Message Service Gateway*
SMPP	*Short Message Peer To Peer*
HLR	*Home Location Register*
VLR	*Visitor Location Register*
URL	*Uniform Resource Locator*
ESME	*External Short Message Entity*
ERB	*Estação Rádio Base*
PDU	*Protocol Data Unit*
CAPTCHA	*Completely Automatic Public Turing Test to Tell Computer and Human Apart*
HTTP	*Hypertext Transport Protocol*
HTTP-GET	*É o método mais comum: solicita algum recurso como um arquivo ou um script CGI (qualquer dado que estiver identificado pelo URI) por meio do protocolo HTTP. O método GET é reconhecido por todos os servidores.*
HTTP-POST	*Envia dados para serem processados (por exemplo, dados de um formulário HTML) para o recurso especificado. Os dados são incluídos no corpo do comando.*
SOAP	*Simple Object Access Protocol*
XML	*Extensible Markup Language*
GPRS	*General Packet Radio Service*
SVM	*Support Vector Machine*
KNN	*k Nearest Neighbors*
MD5	*Message-Digest Algorithm 5*
API	*Application Programming Interface*
APN	*Access Point Name*
HTML	*HyperText Markup Language*
VM	*Virtual Machine*

SUMÁRIO

1 INTRODUÇÃO

1.1 Motivação

Desde o lançamento da telefonia celular, até os dias de hoje, é possível perceber uma evolução dos dispositivos e dos serviços prestados pelas operadoras. Uma grande variedade de serviços passou a ser oferecida. O SMS (*Short Message Service*), conhecido popularmente como torpedos, tornou-se um dos mais importantes serviços por ser de simples utilização e de baixo custo. A utilização dos serviços SMS representou em 2007 12% da receita das operadoras de telefonia celular da Europa (Gartner, 2008). Na China em 2000, foram enviadas 1 bilhão de mensagens SMS. As estimativas de faturamento global previstas até 2013, na utilização de serviços SMS, devem ser de US$ 177 milhões (Boas, 2008).

O serviço SMS representa comodidade e traz vários benefícios aos usuários. Uma mensagem SMS pode ser enviada, por exemplo, para uma pessoa que está em uma reunião e não pode atender ao telefone, ou para os investidores mais assíduos, quando a bolsa de ações tiver uma queda. Confirmar a entrega de um produto comprado pela Internet, ou avisar que o alarme da sua casa foi desativado e por qual usuário, também são exemplos de uso do SMS. A revista britânica *The Economist*, revela o crescimento na utilização de serviços SMS em programas de televisão que interagem com os espectadores, principalmente os *reallity show*, programas de auditório, programas de músicas e até boletins de notícias, que os espectadores podem votar, participar, responder enquetes e fazer comentários. A Figura 1 mostra o crescimento no envio de mensagens de texto dos usuários nos Estados Unidos, nessa figura é possível perceber que o crescimento no envio de mensagens dobrou nos últimos seis anos, e a estimativa para 2010 é manter o crescimento.

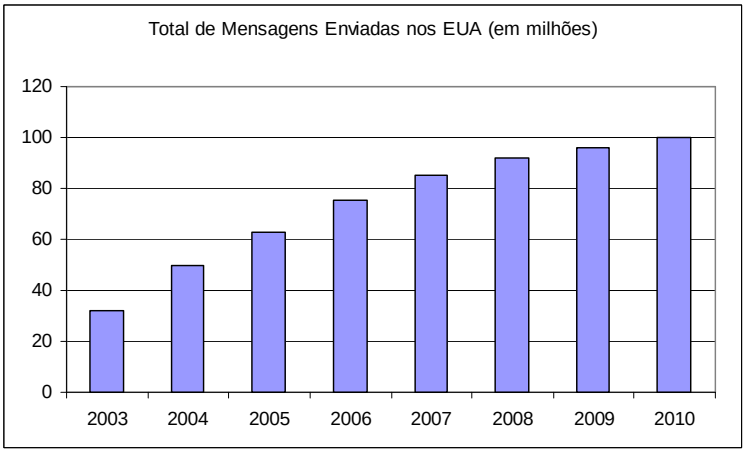

Figura 1. Utilização das Mensagens de Texto nos Estados Unidos.
Fonte: Text Message Statistics, 2009.

Mensagens disparadas por sistemas corporativos e programas de televisão podem trazer um resultado negativo na utilização do serviço, por sobrecarregarem o usuário com informações não solicitadas. Mensagens de propaganda não solicitadas sobre produtos que não interessam, também são indesejáveis. Se cada usuário associar que, ao participar de um programa de televisão numa enquete ou fazer um comentário sobre seu time de futebol, passará a receber mensagens de propaganda não solicitadas ou promoções que não lhe interessam, esses usuários não participarão mais dos programas, além de reclamarem do recebimento indevido a suas operadoras. Essas reclamações podem acarretar um cancelamento no canal SMS ou processos judiciais, em caso de mensagens com conteúdo impróprio, como mensagens pornográficas, políticas e religiosas. Existem ainda, casos de criminosos que utilizam as mensagens de texto ou números de telefone para a obtenção de dados pessoais. De acordo com Wu (2006), durante o ano de 2006, foram enviadas 1,2 milhões de mensagens SMS diárias na China. Dentre elas, mais de 30% foram consideradas spam's. Um spam é uma mensagem eletrônica não solicitada, utilizada mais comumente para fins publicitários ou envio de vírus (Sahami, Dumais, Heckerman e Horvitz, 1998).

Este trabalho propôs a implementação de um filtro de conteúdo para sistemas SMS baseado em Classificador Bayesiano e agrupamento por palavras. A implementação do filtro foi realizada em uma ESME (*External Short Message Entity*). Essa entidade é responsável por entregar as mensagens enviadas pelos provedores de conteúdo à SMSC (*Short Message Service Center*). Tanto a ESME quanto o SMSC são componentes de uma arquitetura da

operadora de telefonia celular, que será explicada detalhadamente no trabalho. O papel do filtro de conteúdo na ESME é classificar e bloquear as mensagens classificadas como spam.

1.2 Objetivo

O objetivo geral deste trabalho é desenvolver um filtro de conteúdo para sistemas SMS baseado em Classificador Bayesiano e agrupamento por palavras.

Os objetivos específicos são:

- Implementar um algoritmo que utilize o Classificador Bayesiano para classificar as mensagens de sistemas SMS a partir de um agrupamento por palavras.
- Apresentar uma proposta de uma arquitetura para o filtro de conteúdo em sistemas SMS.
- Realizar os testes em um ambiente real de uma operadora de telefonia celular.

1.3 Justificativa

Considerando o alto percentual de mensagens indesejadas em sistemas SMS, este trabalho apresenta um filtro de conteúdo baseado em Classificador Bayesiano, para detectar spam's previamente. Na bibliografia consultada, outros autores (Deng e Peng, 2006) implementaram filtros de conteúdo baseados em Classificadores Bayesianos. A proposta deste trabalho deferencia-se da deles por permitir o agrupamento de palavras das mensagens no filtro e na escalabilidade dos testes. O algoritmo desenvolvido na implementação do filtro foi integrado a uma plataforma de envio de mensagens para sistemas SMS, de uma empresa provedora de conteúdo SMS, integrada as grandes operadoras de telefonia celular do país.

1.4 Organização do Texto

Este trabalho está organizado da seguinte forma: o Capítulo 2 apresenta a fundamentação teórica, onde são apresentadas as tecnologias existentes sobre filtros de conteúdo. O Capítulo 3 descreve os trabalhos relacionados. O Capítulo 4 apresenta a proposta Implementação da Classificação Bayesiana em um Sistema SMS. O Capítulo 5 apresenta os experimentos, os resultados e análise. O Capítulo 6 apresenta as conclusões e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Spam

Spam é definido por Sahami, Dumais, Heckerman e Horvitz (1998) como uma forma de encher caixas de mensagens com mensagens não solicitadas a respeito de qualquer coisa, desde artigos para venda e formas de como enriquecer rapidamente, a informações sobre como acessar sites pornográficos. Para Cranor e LaMacchia (1998), os principais fatores que contribuem para o crescimento do número de spam são a facilidade de enviá-lo para um grande número de destinatários e de se obter endereços de e-mails válidos, além do baixo custo de envio. Cormack e Lynam (2005), definem o spam como e-mail não solicitado, emitido de forma indiscriminada, direta ou indiretamente, por um remetente que não tem nenhum relacionamento com o destinatário. O spam pode ser considerado o equivalente eletrônico das correspondências indesejadas e dos telefonemas de telemarketing não solicitados. Silva (2009), define spam como um termo utilizado para denominar mensagens eletrônicas que, na maioria das vezes, são de intuito publicitário, visando à promoção de serviços, produtos ou eventos, e que são enviadas sem o consentimento prévio dos destinatários.

2.2 Formas de Bloqueio e Detecção de Spam

As formas de bloqueio e detecção de um spam estão divididas em: listas de bloqueio, bloqueio temporário de mensagens (*greylisting*), autenticação da organização que está enviando aquele e-mail (*DomainKeys Identified Mail*) e filtros de conteúdo. As três primeiras técnicas se baseiam em bloquear o e-mail de acordo com o remetente ou quem está enviando o e-mail, podendo bloquear mensagens vindas de um remetente, servidor, ou domínio. A

última técnica se baseia em analisar o conteúdo do e-mail e detectar se aquele e-mail é ou não spam. Algumas soluções podem utilizar mais de uma técnica ao mesmo tempo.

2.3 Filtros de Conteúdo propostos para E-mail

Os primeiros filtros de conteúdo que surgiram foram para spam's de e-mail. Segundo Ming, Yunchun e Wei (2007), várias técnicas sobre filtragem de spam para e-mails foram criadas. Elas podem ser divididas em três técnicas: a primeira é baseada em palavra-chave, onde os algoritmos extraem características do corpo do e-mail e identifica as correspondências com essa palavra-chave. Essa técnica é considerada passiva e demorada. Por exemplo, quando os *spammers* alteram as palavras dos e-mails, o método se torna ineficaz por não encontrar as palavras e, quando há muitas palavras, as pesquisas são muito demoradas.

A segunda técnica filtra o conteúdo. Essa técnica pode ser vista como um caso particular de categorização de texto, onde apenas duas classes são possíveis: spam e não spam. As soluções que utilizam essa técnica estão entre os principais métodos: Classificador Bayesiano, SVM (*Support Vector Machine*), K-NN K-Vizinhos Mais Próximos, Redes Neurais Artificiais, Árvores *Boosting*, Aprendizado Baseado em Memória, Rocchio e Sistemas Imunológicos Artificiais. A filtragem de conteúdo possui algumas desvantagens. Primeiramente, ocorre um desperdício de largura de banda de rede, onde não é possível tomar decisão sobre o tipo do e-mail enquanto o mesmo não tenha sido baixado inteiramente no cliente. Em segundo lugar, é difícil garantir a atualidade da amostra, pela quantidade de e-mails e mudanças nos conteúdos de spam, sendo difícil garantir o efeito e a persistência do algoritmo anti-spam.

A terceira técnica é baseada na filtragem de spam. Essa técnica pode detectar spam e evitar as desvantagens da primeira e da segunda técnica. É uma técnica que reconhece spam através do comportamento dos usuários ao enviar novos e-mails, construindo o processo de decisão para receber e-mails. Essa técnica utiliza o comportamento dos *spammers* para detectar novos spams.

2.4 Métodos de detecção de spam's de E-mail baseados em filtros de conteúdo

Essa sessão apresenta os principais métodos de aplicação da segunda técnica de filtragem de conteúdo. Serão apresentados os métodos: Classificador Bayesiano, Máquina de Vetor de Suporte Multiclasses ou SVM, K-NN K-Vizinhos Mais Próximos, Redes Neurais Artificiais, Árvores *Boosting*, Aprendizado Baseado em Memória, Rocchio e Sistemas Imunológicos Artificiais.

2.4.1 Redes Neurais Artificiais

Para Braga, Carvalho e Ludermir (2000), as Redes Neurais Artificiais são modelos matemáticos que se assemelham às estruturas neurais biológicas e que têm capacidade computacional adquirida por meio de aprendizagem e generalização. Esses modelos almejam semelhança com o sistema nervoso dos seres vivos e a com sua capacidade de processar informações. As redes neurais são sistemas que possuem a capacidade de adquirir, armazenar e utilizar conhecimento experimental. Esse conhecimento está embutido na rede sob a forma de estados estáveis, que podem ser relembrados, em resposta à apresentação de estímulos (Zurada, 1992). A capacidade de aprender com exemplos, a robustez, a velocidade de processamento, a generalização e a adaptabilidade, possibilitam a utilização das redes neurais artificiais na solução de uma grande variedade de problemas, entre eles, problemas de classificação, otimização, categorização, aproximação, análise de sinais ou imagens e predição (Braga, Carvalho e Ludermir, 2000).

2.4.2 Árvores Boosting

Para Carreras e Marquez (2001), a ideia geral deste método é encontrar uma regra de classificação de alta precisão através da combinação de regras fracas ou hipóteses fracas.

Essas regras ou hipóteses devem ser moderadamente precisas. Esse método mantém um conjunto de pesos relevantes dos padrões de treinamento, utilizando-os para forçar, a partir de cálculos probabilísticos, o aprendizado a concentrar-se nos exemplos de mais difícil classificação. É realizado um treinamento em um primeiro conjunto de amostras. Nesse treinamento são selecionados elementos para ser formar o segundo conjunto de amostras, e, assim, sucessivamente. Um exemplo de um algoritmo utilizando esse método é chamado de AdaBoost.

2.4.3 Máquina de Vetores de Suporte Multiclasses SVM

Para Lorena (2006) as Máquinas de Vetores de Suporte Multiclasse SVM são um conjunto de métodos de aprendizagem supervisionados utilizados para classificação e regressão. Essa máquina pode construir um hiperplano ou um conjunto de hiperplanos em um espaço que pode ser de dimensão infinita. Essa máquina pode ser utilizada para classificação, regressão ou outras tarefas. Uma boa separação é alcançada pelo hiperplano que tem a maior distância para a formação dos dados.

2.4.4 Aprendizado Baseado em Memória

Método que armazena dados de treinamento em uma estrutura de memória. O método assume que o ato de raciocinar é baseado diretamente na reutilização de experiências armazenadas em vez do conhecimento adquirido. Novos dados são classificados estimando-se a similaridade com os exemplos armazenados. Esse método foi empregado em Zhang, Zhu e Yao (2004) e Sakkis et al. (2003) [Silva et al., 2009].

2.4.5 Rocchio

A idéia básica do algoritmo de Rocchio, quando aplicada na categorização de texto, é utilizar um conjunto de documentos de treinamento para construir um vetor base por classes. Dado um documento não classificado, ele é comparado ao protótipo de cada uma das classes, aplicando-se uma função de similaridade. Obtém-se, então, a categoria na qual o documento tem a maior probabilidade de pertencer. Drucker, Wu e Vapnik (1999) utilizaram esse método [Silva et al., 2009].

2.4.6 Sistemas Imunológicos Artificiais

Sistemas Imunológicos Artificiais utilizam a heurística do aprendizado de máquina baseado no sistema imuno-biológico. Segundo De-Castro et al. (2001), é possível relacionar as características de imunovigilância e resposta imune aos procedimentos de segurança computacional, que podem incluir detecção de spam, eliminação de vírus e intrusos de rede. Guzella et al. (2005) utilizaram esse método na identificação de spams e obtiveram bons resultados [Silva et al., 2009].

2.4.7 Classificador Bayesiano

A técnica filtragem de conteúdo baseada em classificador Bayesiano, segundo Silva (2009), é bastante utilizada em problemas de categorização de texto e em filtros anti-spam. A ideia básica é usar a probabilidade na estimativa de uma dada categoria presente em um documento ou texto. Assume-se que existe independência entre as palavras, tornando-se mais simples e rápido. Essa solução foi apresentada nos trabalhos de Kosmopoulos, Paliouras e Androutsopoulos (2008), Zhang, Zhu e Yao (2004), Ozgur, Gungor e Gurgen (2004), Meyer e Whateley (2004), Androutsopoulos et al. (2000b), alcançando bons resultados. Algumas

ferramentas utilizam esse método, como o SpamAssassin (SPAMASSASSIN, 2008) e o Bogofilter (RAYMOND, 2003).

Na teoria da probabilidade, o teorema de Bayes é relacionado à probabilidade condicional de dois eventos aleatórios. Criado por Thomas Bayes, um famoso matemático britânico que viveu no século 18, o teorema de Bayes é utilizado para calcular probabilidades posteriores, a partir de informações coletadas no passado. Ele representa uma abordagem teórica estatística de inferência indutiva na resolução de problemas (Kantardzic 2003). Por exemplo, ao se observar alguns sintomas de um paciente, é possível, utilizando-se o teorema, calcular a probabilidade de um diagnóstico (Sahami, 1998).

De acordo com o teorema de Bayes, a probabilidade é dada por uma hipótese dos dados, considerada base posterior, que é proporcional ao produto da probabilidade vezes a probabilidade prévia. A probabilidade posterior representa o efeito dos dados atuais, enquanto a probabilidade prévia especifica a crença na hipótese, ou o que foi coletado anteriormente. O teorema nos permite combinar a probabilidade desses eventos independentes em um único resultado (Sahami, 1998).

$$P(H \mid X) = \frac{P(X \mid H) \cdot P(H)}{P(X)} \qquad (1)$$

Kantardzic (2003) apresenta a classificação bayesiana da seguinte forma, seja *X* uma amostra de dados, cuja classe seja desconhecida, e seja *H* alguma hipótese, tal que os dados específicos da amostra *X* pertencem à classe *C*. Pode-se determinar $P(H \mid X)$, a probabilidade da hipótese *H* possui dados observados na amostra *X*, onde $P(H \mid X)$ é a probabilidade posterior que representa a confiança na hipótese. $P(H)$ é a probabilidade prévia de *H*, para qualquer amostra, independente de como a amostra dos dados aparecem. A probabilidade posterior $P(H \mid X)$ baseia-se em obter mais informações na probabilidade prévia $P(H)$. O Teorema Bayesiano provê o cálculo da probabilidade posterior $P(H \mid X)$ utilizando as probabilidades $P(H)$, $P(X)$ e $P(X \mid H)$, conforme fórmula (1).

Suponha que existe um conjunto de *m* amostras $S = \{S_1, S_2, ..., S_m\}$ (conjunto de dados já treinados), onde cada amostra S_i é representada por um vetor de *n* dimensões $\{x_1, x_2, ..., x_n\}$. Valores de x_i correspondem aos atributos $A_1, A_2, ..., A_n$, respectivamente. Além disso, existem *k* classes $C_1, C_2, ..., C_k$ e cada uma das amostras pertencem a uma dessas

classes. Dada uma amostra de dados adicionais *x* (onde sua classe é desconhecida), é possível predizer a classe para *x* usando a probabilidade condicional mais elevada $P(C_i \mid X)$, onde $i = 1,...,k$. As probabilidades são obtidas a partir do Teorema de Bayes:

$$P(C_i \mid X) = \frac{P(X \mid C_i) \cdot P(C_i)}{P(X)} \qquad (2)$$

Em (2), $P(X)$ é constante para todas as classes, somente o produto $P(X \mid C_i) \cdot P(C_i)$ deve ser maximizado. $P(C_i)$ é o número de amostras treinadas para a classe C_i / m (*m* é o total de amostras treinadas). Tendo em vista a complexidade do cálculo de $P(X \mid C_i)$, especialmente para grandes conjuntos de dados, é realizado um pressuposto ingênuo de independência condicional entre os atributos. Utilizando esse pressuposto, é possível expressar $P(X \mid C_i)$ como um produto:

$$P(X \mid C_i) = \prod_{t=1}^{n} P(x_t \mid C_i) \qquad (3)$$

Em (3), x_t são valores para atributos na amostra *X*. As probabilidades $P(x_t \mid C_i)$ podem ser estimadas a partir do conjunto de dados treinados.

2.5 Filtros de Mensagens SMS

Dentre as técnicas conhecidas para filtragem de e-mail algumas delas também são utilizadas na filtragem de mensagens SMS (*Short Message Service*). Dentre elas, listas de bloqueio e filtros de conteúdo, utilizando técnicas KNN, SVM e Bayes. Além dessas técnicas, foi encontrada também uma técnica chamada CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*) nos trabalhos de Wen e Zheng (2009), He, Wen e Zheng (2008), Shahreza (2008), Zhang, He, Sun, Zheng e Wen (2008) e Shahreza (2006), onde é possível realizar um teste cognitivo para identificar um ser humano e um

computador. Nesse caso, o filtro solicita uma resposta ao remetente sobre uma pergunta que não pode ser realizada por um sistema, inibindo, assim, as mensagens indevidas.

2.6 O problema do falso positivo e do falso negativo

Um dos erros inaceitáveis na detecção de spams é não se enviar uma determinada mensagem ao usuário, por classificá-la incorretamente como spam. Esse erro é denominado falso positivo. Ou seja, ao classificar essa mensagem, o algoritmo classifica essa mensagem como spam e não envia ao usuário. Em programas de e-mail, esse problema pode ser parcialmente resolvido configurando-o para enviar mensagens classificadas como spam para uma pasta específica. Sendo assim, o usuário poderá verificar essa pasta constantemente, em busca de mensagens não spam classificadas incorretamente. Um outro tipo de erro, denominado falso negativo, não é tão grave, o único aborrecimento é que o usuário vai receber novas mensagens spam em sua caixa de entrada de e-mails. Quando isso ocorrer, o único trabalho que o usuário terá será excluir essa mensagem, ou denunciá-la como spam, caso o seu programa de e-mails tenha essa opção (Assis, 2006).

2.7 Estrutura de Serviços SMS

O serviço de mensagem de texto SMS (*Short Message Service*) é considerado como um dos maiores trunfos dos sistemas GSM (*Global System for Mobile Communications*), implantado também em outras redes de celulares. Inicialmente, o serviço foi criado pelo estudante da Nokia, Riku Pihkonen em 1993 (Mobilen50ar, 2009), para prover o envio de configurações de dados para o celular, como por exemplo envio de configurações de APN (*Access Point Name*) entre a operadora e o aparelho do cliente. Posteriormente, foi adotado para troca de mensagens. O serviço SMS utiliza faixas dos canais da rede GSM. Ele pode ser utilizado por telefones móveis, *notebooks* e computadores pessoais (Detter, 1997), que estejam conectados a uma rede de celulares.

2.7.1 SMSC

O serviço é gerenciado por um centro de distribuição denominado SMSC (*Short Message Service Center*), que pode enviar mensagens de texto SMS usando uma carga máxima de 140 octetos, definindo, assim, um limite de 160 caracteres, usando 7 bits de codificação. É possível também se utilizar 8 ou 16 bits de codificação, diminuindo o tamanho máximo da mensagem para 140 ou 70 respectivamente (Brown, 2007).

Além do envio de mensagens de texto, o serviço SMS é utilizado para envio de dados binários aos dispositivos móveis, como por exemplo: tons musicais (mais conhecidos como *ring tones*), troca de mensagens de imagens e temas, como, por exemplo, fundo de tela do dispositivo. Nesses casos, as mensagens originais podem ultrapassar o limite máximo de caracteres. Quando isso ocorrer, é definido no cabeçalho da mensagem o segmento de informações seqüenciais, para que o dispositivo possa agrupar as mensagens em um único arquivo (Brown, 2007).

Para disponibilizar um serviço SMS, uma operadora de telefonia celular necessita de uma infra-estrutura própria, conforme ilustra a Figura 2. Dentre os principais componentes dessa infra-estrutura, podem ser citados: SMSC (*Short Message Service Center*), responsável por encaminhar a mensagem ao dispositivo do usuário, validando se o cliente possui o serviço habilitado na base de dados HLR (*Home Location Register*), a MSC (*Mobile Switching Center*) responsável pelo roteamento das chamadas e mensagens SMS, a ERB (Estação Rádio Base) responsável por realizar a entrega da mensagem ao dispositivo móvel, o SMSG (*Short Message Service Gateway*) responsável por realizar a entrega de mensagens entre redes diferentes ou operadoras de telefonia celular diferentes, e a ESME (*External Short Message Entity*) entidade responsável por fazer a integração da SMSC com os provedores de conteúdo, responsáveis por enviar conteúdos em formato de texto aos clientes. Cada um dos componentes será melhor explicado nas subseções a seguir.

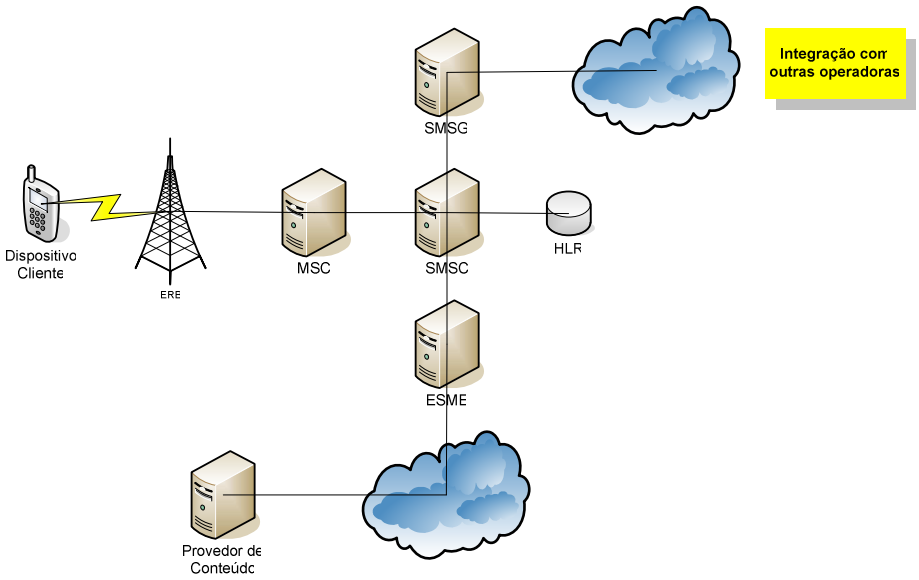

Figura 2. Arquitetura necessária para serviço SMS.

2.7.2 HLR

O HLR (*Home Locator Register*) é a base de dados que possui informações sobre os serviços contratados pelos assinantes das operadoras. Quando for necessário a SMSC solicita informações de encaminhamento e roteamento sobre o assinante ao HLR. Caso o celular não esteja disponível no momento da entrega da mensagem, o HLR recebe uma informação de pendência de mensagem. Quando o usuário estiver disponível novamente o HLR avisa a SMSC que o assinante está disponível novamente para que a mensagem seja entregue (Prieto, Cosenza e Stadler, 2004) (3GPP, 2006).

2.7.3 MSC

O MSC (*Mobile Switching Center*) realiza as funções de comutação por circuito dos sistemas de controle de chamadas para telefones e outros sistemas de dados. O MSC consiste em realizar a interface entre o sistema de rádio e a rede fixa. O MSC entrega a mensagem

enviada pela SMSC ao assinante através da ERB mais adequada. Também realiza a entrega de mensagens enviadas do assinante para a SMSC (3GPP, 2006).

2.7.4 ERB

A ERB (Estação Rádio Base) é o equipamento responsável pela comunicação entre o dispositivo móvel e a operadora de telefonia celular. A partir da ERB o dispositivo móvel comunica com a operadora. Ela transmite mensagens enviadas da operadora ao assinante e vice-versa. A ERB converte sinais de rádio em sinais de áudio e vice-versa. Informa ao dispositivo móvel sobre a sua área de cobertura, além de informar ao dispositivo móvel a qualidade do sinal, a presença de outros dispositivos móveis, e responder a comandos da operadora. (Subramanya e Yi, 2005) (3GPP, 2006).

2.7.5 SMSG

O SMSG (*Short Message Service Gateway*) é o *gateway* responsável por realizar as integrações entre as operadoras de telefonia celular. A partir desse *gateway*, é possível enviar mensagens de uma operadora para outra. Ou seja, quando um usuário deseja enviar mensagens para usuários de outras operadoras diferentes da operadora dele. Utiliza como meio de comunicação o protocolo TCP/IP. Quando as redes interconectadas não falam os mesmos protocolos, o *gateway* deve executar as traduções de protocolo (3GPP, 2006).

2.7.6 ESME

A ESME (*External Short Message Entity*) é a entidade responsável por realizar a integração da SMSC com os provedores de conteúdo. Normalmente está localizada na rede da operadora, mas também pode estar em um dispositivo móvel ou dentro dos provedores de conteúdo. Suas funções básicas são enviar e receber mensagens entre o celular e os provedores de conteúdo. Em algumas operadoras, as ESMEs também realizam o papel de cobrar por assinaturas de canais de conteúdo. Além do envio de mensagens as ESMEs podem realizar consultas sobre o estado das mensagens enviadas à SMSC, podendo cancelar uma mensagem, caso seja necessário. A SMSC pode enviar mensagens à ESME. Um exemplo típico dessa utilização é quando a SMSC avisa à ESME que determinada mensagem foi entregue ao usuário. O protocolo utilizado para a conexão entre a SMSC e a ESME é o SMPP (*Short Message Peer to Peer*) (3GPP, 1999).

2.7.7 Dispositivo Móvel

O dispositivo móvel ou celular, é a interface entre o usuário e o sistema. Transforma sinais de áudio em sinais RF e vice-versa, envia mensagens escritas pelo usuário para a operadora pela ERB, recebe mensagens enviadas pela ERB e mostra-as ao usuário, alerta usuário sobre chamadas recebidas e alerta ao sistema sobre tentativas de originação de chamadas (3GPP, 2006).

2.7.8 Provedor de Conteúdo

O Provedor de Conteúdo é a entidade que fornece o conteúdo e os aplicativos para dispositivos móveis. Por exemplo, quando um assinante envia uma mensagem de texto para recuperar informações ou participar de um programa de televisão, o provedor de conteúdo retorna as informações ou computa votos, dependendo do tipo de interatividade que o assinante estiver participando. Além disso, o provedor de conteúdo envia mensagens sobre determinados assuntos, de acordo com canais solicitados pelos assinantes das operadoras. Em

alguns casos os provedores de conteúdos são responsáveis pelos envios de spam (KRISHNAMURTHY, 2002).

2.8 Android

O Android é um projeto de uma arquitetura de desenvolvimento para celulares desenvolvido por um grupo de mais de 30 empresas. Dentre elas, podem ser citadas Google, HTC, LG Eletronics, Acer, Samsung, Sony Ericsson, Vodafone, ARM, Intel, MIPS Tecnologies, Qualcomm, Telefônica, T-Mobile, Garmin, Asus, entre outras. O projeto visa desenvolver a primeira plataforma completa e gratuita para telefones celulares. Essa plataforma é baseada na versão 2.6 do Linux, um sistema operacional que tem funções de segurança, gerenciamento de memória, gestão de processos, pilhas de rede e modelo de *drivers* para recursos. Além disso, o sistema operacional Linux atua com uma camada de abstração entre o *hardware* e a pilha de *software*. A Figura 3 apresenta a arquitetura do Android.

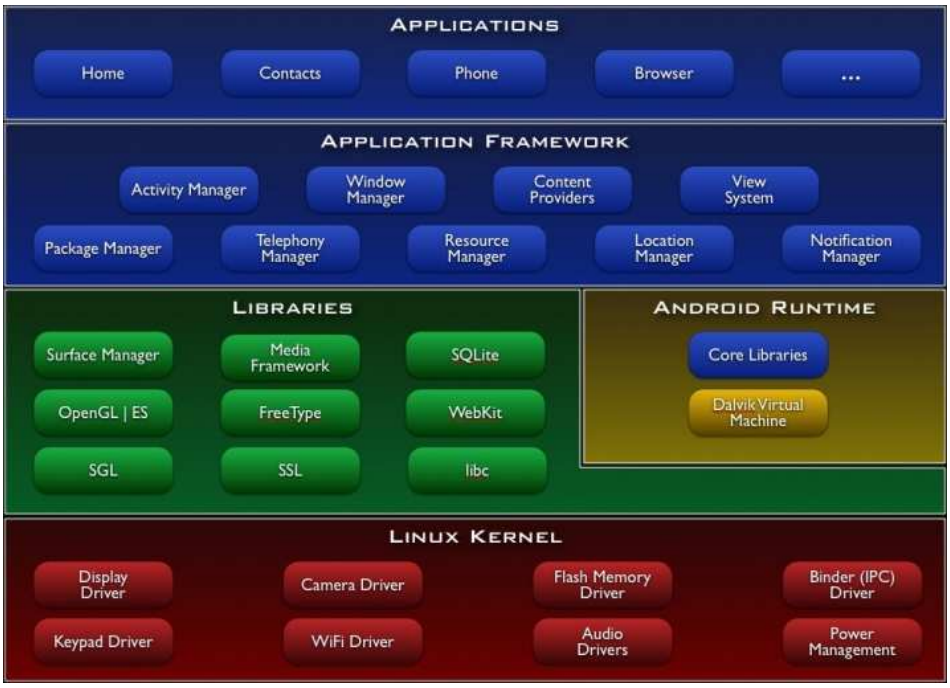

Figura 3. Arquitetura Android.
Fonte: Android (2009).

Na Figura 3 é possível perceber cinco partes principais do Android. A primeira parte chamada *Application*, é associada a um conjunto de aplicações, incluindo um cliente de e-mail, um programa SMS, calendário, mapas, navegador, contatos e outros. A segunda parte chamada *Application Framework,* onde os desenvolvedores possuem pleno acesso às API's do mesmo quadro usado pelos aplicativos essenciais. A arquitetura do aplicativo é projetada para simplificar a reutilização de componentes; qualquer aplicativo pode publicar as suas capacidades e qualquer outra aplicação pode então fazer uso dessas capacidades (sujeito a restrições de segurança impostas pelo *framework*). A terceira parte chamada *Libraries* é onde existe um conjunto de bibliotecas usadas por diversos componentes do sistema Android. A quarta parte chamada Android *Runtime*, corresponde a um conjunto de bibliotecas que fornece a maioria das funcionalidades disponíveis nas principais bibliotecas da linguagem de programação Java. Toda aplicação Android é executada em seu próprio processo, com a sua própria instância da máquina virtual Dalvik. Dalvik foi escrito de modo que um dispositivo pode executar várias VM's (*Virtual Machines*) eficientemente. O Dalvik VM executa os arquivos em executáveis Dalvik (DEX). Este formato é otimizado para consumo o mínimo de memória. A VM é baseada em registradores, e executa classes compiladas por um compilador da linguagem Java que foram transformadas para o formato. A VM Dalvik depende do *kernel* do Linux para funcionalidades subjacentes como o encadeamento de baixo nível e de gerenciamento de memória. E a quinta parte chamada, Linux *Kernel*, é onde existem sistemas de serviços essenciais como segurança, gerenciamento de memória, gestão de processos, rede de pilha, modelo e *driver*. O *kernel* também atua como uma camada de abstração entre o hardware e o resto da pilha de software.

Os desenvolvedores que utilizam o Android possuem todo acesso às API's (*Application Programming Interface*) utilizadas no desenvolvimento dos aplicativos essenciais. A arquitetura Android é projetada para simplificar a utilização de código, onde qualquer aplicativo pode publicar suas funções e quaisquer outras aplicações podem utilizá-las. Tudo isso é sujeito a restrições de segurança impostas pela arquitetura, conforme mencionado anteriormente.

Os recursos disponibilizados aos desenvolvedores de aplicações para o Android são: acesso ao *framework*, máquina virtual para desenvolvimento, integração a um navegador otimizado, bibliotecas gráficas 2D e 3D, SQLite, suporte a áudio e vídeo, tecnologia GSM, Bluetooth, EDGE, WiFi, câmera, GPS e acelerômetro.

3 TRABALHOS RELACIONADOS

Neste Capítulo, serão apresentados os trabalhos propostos por Deng e Peng (2006), Maier e Ferens (2009), Sirat e Susatyo (2008) e Silva (2009). Todos estes trabalhos apresentam propostas de filtragem de conteúdo de mensagens de e-mails e mensagens SMS. As propostas de Deng e Peng (2006) e de Sirat e Susatyo (2009) utilizam o Classificador Bayesiano como base para classificar as mensagens SMS. A proposta de Maier e Ferens (2009) realiza uma comparação entre a Classificação Bayesiana Multinomial, Classificação Bayesiana de Poisson e SVM (*Support Vector Machine*) para a classificação de mensagens SMS. E a proposta de Silva (2009) utiliza as Redes Neurais Aritificiais como base para classificar as mensagens de e-mails.

3.1 Research on a Naive Bayesian Based Short Message Filtering System

O trabalho apresentado por Deng e Peng (2006) propõe um filtro de spam baseado em Classificador Bayesiano para sistemas SMS. A proposta de Deng e Peng (2006) foi implementada em dois módulos apresentados na Figura 4, o do SMSC (*Short Message Service Center*) e do dispositivo móvel do assinante. O módulo do SMSC detecta as mensagens entendidas como spam e envia as atualizações do filtro aos dispositivos móveis. Ele é dividido em quatro partes: *Bayesian Learner, Blocked number list, Update Information Handler e Clients Reports Handler.* O *Bayesian Learner* é responsável pelos cálculos da Classificação Bayesiana e aprendizado. O *Bayesian Learner* recalcula as probabilidades a partir de novas mensagens coletadas e informações enviadas pelo módulo do dispositivo móvel. O *Blocked number list* é responsável por bloquear as mensagens enviadas pelos remetentes que estiverem na lista negra e liberar as mensagens dos remetentes que estiverem na lista branca. O *Update Information Handler* é responsável por enviar atualizações dos dados do filtro de conteúdo ao dispositivo móvel. O *Clients Reports Handler* é responsável por receber as informações reportadas pelos dispositivos móveis, como os erros e acertos de

classificação do filtro. O módulo do dispositivo móvel recebe as mensagens enviadas pelo módulo SMSC. Ao receber novas mensagens, o módulo do dispositivo móvel verifica as mensagens que foram consideradas spam e não spam pelo usuário reportando ao servidor as classificações realizadas e atualizando as informações novas baixadas do servidor para a base de dados interna do dispositivo móvel. Além desses módulos a Figura 4 apresenta o módulo *Junk SMS Sender*¸ que representa quem enviou a mensagem spam ao SMSC.

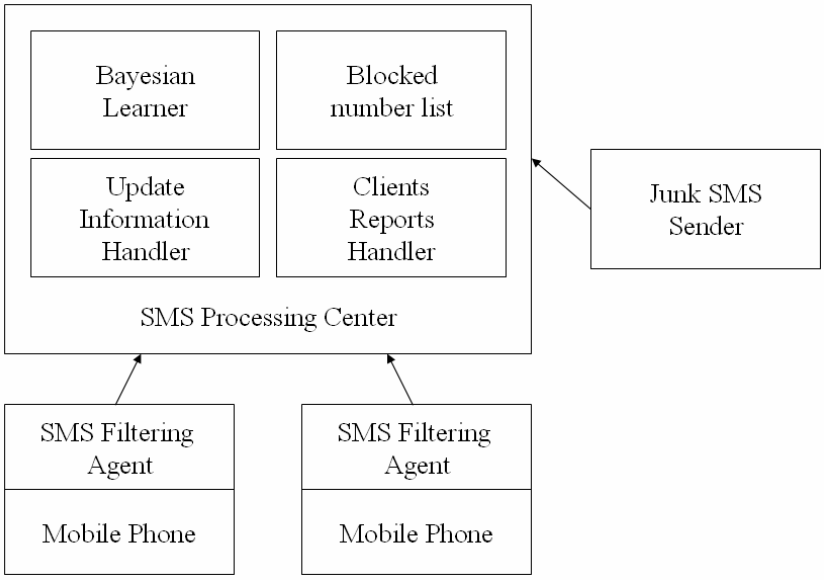

Figura 4. Arquitetura distribuída proposta por Deng e Peng (2006)
Fonte: Dados da pesquisa de Deng e Peng (2006).

Segundo Deng e Peng (2006), utilizando-se apenas o atributo palavra para a classificação de mensagens spam e não spam na Classificação Bayesiana, não se obtém um resultado satisfatório. Pela pesquisa realizada por eles, o acerto obtido é de 92% quando utiliza-se apenas o atributo palavra. Isso representa 8% de falsos positivos e negativos. Quando combinaram a solução do filtro com o atributo palavra, com o filtro com os atributos, telefone, URL, tamanho da mensagem, valores monetários, lista negra (*blacklist*) de remetentes a serem bloqueados e lista branca (*whitelist*) de remetentes confiáveis, o acerto chegou a 99,5 ao classificar mensagens spam e não spam, segundo Deng e Peng (2006).

A utilização do tamanho da mensagem como atributo, veio da percepção de que quanto maior a mensagem, maior a probabilidade da mensagem ser spam. A Figura 5 mostra

que 9% das mensagens não spam usam no máximo 10 caracteres, e as mensagens consideradas spam utilizam em média 55 a 60 caracteres.

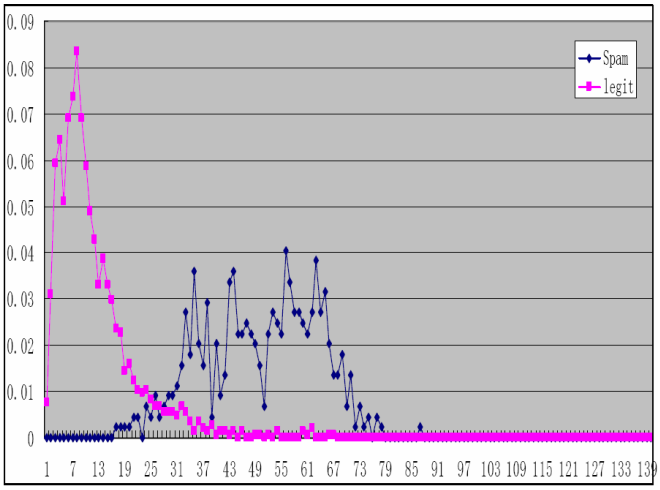

Figura 5. Porcentagem de utilização de caracteres nas mensagens
Fonte: Dados da pesquisa de Deng e Peng (2006).

Os atributos adicionados da proposta Deng e Peng (2006) obtiveram um percentual considerável na classificação das mensagens. A Tabela 3.1 apresenta o percentual obtido nos testes realizados por Deng e Peng (2006) para cada atributo utilizado. Nesses resultados é possível perceber que do total de mensagens testadas que possuíam o atributo telefone, 20,3% das mensagens foram classificadas como spam e 5,3% das mensagens foram classificadas como não spam. Para o atributo URL 3,1% das mensagens foram classificadas como spam e 1,1% das mensagens foram classificadas como não spam. Para o atributo valores monetários, 5,2% das mensagens foram classificadas como spam e 1,2% das mensagens foram classificadas como não spam. Para os remetentes que estavam na lista branca, nenhuma mensagem foi classificada como spam e 73,1% das mensagens foram classificadas como não spam. Para os remetentes que estavam na lista negra, 23,3% das mensagens foram classificadas como spam e nenhuma mensagem foi classificada como não spam. Para o atributo telefone, 65% das mensagens foram classificadas como spam e 100% das mensagens foram classificadas como não spam.

Tabela 3.1 – Atributos criados por Deng e Peng (2006).

Id do Atributo	Descrição do Atributo	% Spam	% Não Spam
R_1	Mensagens com telefone	20,3%	5,3%

R_2	Mensagens com URL	3,1%	1,1%
R_3	Mensagens com valores monetários	5,2%	1,2%
R_4	Mensagens enviadas dos telefones da lista branca	0%	73,1%
R_5	Mensagens enviadas dos telefones da lista negra	23,3%	0%
R_6	Remetentes enviam telefones formatados	65%	100%

Para a realização dos experimentos, Deng e Peng (2006) criaram dois índices para calcular a eficiência do filtro. O índice *SP*, apresentado em (1), calcula o número de mensagens spam, sobre o número de mensagens spam que foram consideradas spam, mais o número de mensagens não spam consideradas spam. O índice *SR*, apresentado em (2), calcula o número mensagens spam consideradas spam sobre o número de mensagens spam.

$$SP = \frac{n_{spam \to spam}}{n_{spam \to spam} + n_{não\_spam \to spam}} \qquad (1)$$

$$SR = \frac{n_{spam \to spam}}{n_{spam}} \qquad (2)$$

A Tabela 3.2 apresenta os resultados obtidos para os testes realizados sem adicionar os novos atributos propostos por Deng e Peng (2006). Nessa tabela, é possível perceber que, para 100 palavras, o filtro obteve um percentual de acerto de 93,33% das mensagens spam que foram consideradas spam pelo filtro, e 78,8% de acerto para as mensagens não spam. À medida que se aumentou o número de palavras treinadas, o percentual de acerto do filtro também aumentou, chegando a 96,5% de acerto para as mensagens spam consideradas spam pelo filtro e 81,8% de acerto para as mensagens não spam.

Tabela 3.2 – Cálculo sem adicionar as atributos novos

Quantidade de palavras	**SP**	**SR**	$S_{spam \to spam}$	$S_{não\_spam \to spam}$
100	0,933	0,788	1182	85
200	0,966	0,814	1221	43
500	0,965	0,818	1227	54

A Tabela 3.3 apresenta os resultados obtidos por Deng e Peng (2006) para os testes realizados adicionando todos os atributos, menos o atributo tamanho da mensagem. Nessa tabela é possível perceber que, para 100 palavras o filtro obteve um percentual de acerto de 97% das mensagens spam que foram consideradas spam pelo filtro, e 88,9% de acerto para as mensagens não spam. À medida em que se aumentou o número de palavras treinadas, o percentual de acerto do filtro também aumentou, chegando a 98,5% de acerto para as mensagens spam consideradas spam pelo filtro e 93,3% de acerto para as mensagens não spam.

Tabela 3.3 – Cálculo adicionando apenas os atributos sem tamanho

Quantidade de palavras	**SP**	**SR**	$S_{spam \rightarrow spam}$	$S_{não\_spam \rightarrow spam}$
100	0,97	0,889	1333	37
200	0,985	0,939	1409	22
500	0,985	0,933	1400	21

A Tabela 3.4 apresenta os resultados obtidos por Deng e Peng (2006) para os testes realizados adicionando todos os atributos. Nessa tabela é possível perceber que, para 100 palavras, o filtro obteve um percentual de acerto de 98,8% das mensagens spam que foram consideradas spam pelo filtro, e 95,4% de acerto para as mensagens não spam. À medida em que se aumentou o número de palavras treinadas, o percentual de acerto do filtro também aumentou, chegando a 99,1% de acerto para as mensagens spam consideradas spam pelo filtro e 96,4% de acerto para as mensagens não spam.

Tabela 3.4 – Cálculo adicionando todos os atributos

Quantidade de palavras	**SP**	**SR**	$S_{spam \rightarrow spam}$	$S_{não\_spam \rightarrow spam}$
100	0,988	0,954	1431	18
200	0,991	0,963	1445	13
500	0,991	0,964	1446	13

A conclusão apresentada por Deng e Peng (2006), no trabalho realizado, foi que, ao se adicionar os novos atributos propostos no cálculo, os resultados apresentaram uma precisão

maior para se classificar as mensagens como spam e não spam. Ou seja, o filtro com os novos atributos obteve um resultado superior quando se compara com o algoritmo sem os novos atributos, reduzindo o número de falsos positivos e falsos negativos.

3.2 Classification of english phrases and SMS text messages using Bayes and Support Vector Machine Classifiers

O trabalho apresentado por Maier e Ferens (2009) tem como objetivo realizar uma análise comparativa de três diferentes tipos de classificação: Classificação Bayesiana Multinomial, Classificação Bayesiana de Poisson e SVM (*Support Vector Machine*). Existem diversas variantes para a Classificação Bayesiana, dentre elas Bernoulli, Multinomial e de Poisson. No trabalho de Maier e Ferens (2009) foram escolhidas a Classificação Bayesiana Multinomial e a Classificação Bayesiana de Poisson. A Classificação Bayesiana de Bernoulli assume que as características de uma determinada mensagem existem ou não, ou seja, a abordagem Bernoulli considera apenas se a palavra existe ou não na mensagem. A Classificação Bayesiana Multinomial foi criada para melhorar a abordagem Bernoulli, que assume que as características de uma mensagem existem ou não, e não medir a freqüência dessas características ou a importância que uma determinada característica tem em uma mensagem. A abordagem Multimonial melhora o desempenho da classificação, multiplicando a freqüência de uma característica por um peso. A abordagem Poisson é uma generalização da abordagem Multinomial. Segundo Kim, Han, Rim e Myaeng (2006) a abordagem de Poisson é amplamente utilizada para modelar o número de ocorrências de um determinado fenômeno em um fixo período de tempo ou espaço, como, por exemplo, um número de telefone com as chamadas recebidas, em um quadro, durante um determinado período de tempo, ou um número de defeitos em um comprimento fixo de pano ou papel.

Além das comparações realizadas entre os tipos de classificação, Maier e Ferens (2009) adicionaram em suas análises um pré-processamento nos dados, para avaliar o ganho de desempenho na classificação para cada um dos tipos de classificação analisados. Segundo Maier e Ferens (2009) as frases das mensagens enviadas via SMS podem possuir várias abreviações de palavras, dificultando ainda mais a tarefa de identificação e classificação de cada uma delas. Por exemplo, em inglês, a abreviação de *who* para *w* ou *with* para *w*, que podem ser utilizados em ambos os casos dependendo do sentido da frase. Além disso, os

usuários de SMS utilizam o que é conhecido como *emoticons*, ou a derivação de *emotion* (emoção) mais *icon* (ícone), que é criado por uma seqüência de caracteres que representam uma palavra ou uma emoção, como por exemplo, :) (para uma carinha sorrindo), ou :((para uma carinha triste). Mas, segundo os autores Maier e Ferens (2009), esses efeitos de pré-processamento não produziram um ganho significativo.

Para se obter uma melhor compreensão de comunicação da mensagem SMS, Maier e Ferens estenderam o modelo de comunicação humana proposto por Berlo (1960). A Figura 6 apresenta o modelo proposto por Berlo (1960) para o processo de comunicação, adaptada por Maier e Ferens (2009) para a utilização em mensagens SMS. O modelo consiste em sete estágios. No primeiro estágio chamado por *Source*, utilizam-se atributos como habilidades de comunicação, atitude e conhecimento para construir uma mensagem. No segundo estágio chamado por *Message*, a mensagem pode ser construída por diferentes conteúdos, elementos e caracteres especiais. No terceiro estágio, o canal SMS, no caso seria ler a mensagem, já que a mensagem é transmitida usando caracteres de texto. No quarto estágio chamado por SMS *Encoder*, a mensagem é codificada no alfabeto SMS. No quinto estágio chamada de *Physical Channel*, a mensagem é transmitida pelo meio físico ou pelo canal físico. No sexto estágio chamado por *Channel*, a mensagem é recebida através do meio físico ou canal físico. No sétimo estágio chamado por *Receiver*, a mensagem é de decodificada utilizando os mesmos atributos semelhantes ao estágio *Source*.

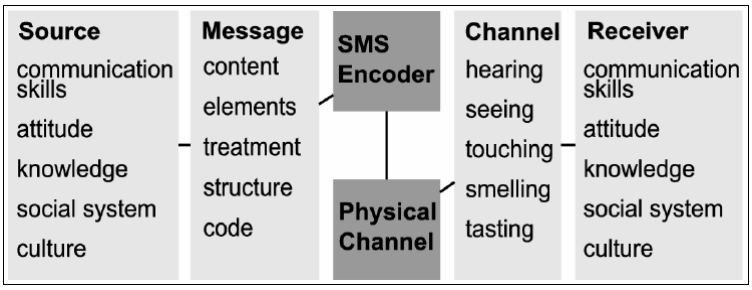

Figura 6. Modelo de Comunicação Humana
Fonte: Dados da pesquisa de Maier e Ferens (2009).

O trabalho de Maier e Ferens (2009) aborda a perda de informações que ocorre quando se trabalha com mensagens SMS. Durante o processo de decodificação, o receptor ou *Receiver*, utiliza de construções consideradas de alto nível. Nesse processo de decodificação são realizadas as habilidades de comunicação, atitude e conhecimento para entender uma

mensagem SMS. Para melhorar a precisão da decodificação da mensagem e aproximar os atributos do canal, Maier e Ferens (2009) adicionaram os três recursos na representação da mensagem. O primeiro recurso é a proporção de abreviaturas de uma mensagem SMS. O segundo recurso é a diferença de comprimento entre as mensagens SMS codificadas e decodificadas. O terceiro recurso é a quantidade de *emotions* com relação ao número absoluto de recursos da frase.

O software desenvolvido por Maier e Ferens (2009) é dividido em dois módulos: decodificação e classificação. O módulo de decodificação é dividido entre *tokenização*, processamento SMS, representação de recursos, ponderação e redução de recursos. O módulo de classificação é dividido entre as formas de classificação comparadas, Multinomial Bayes, Poisson Bayes e *Support Vector Machine* (SVM). A Figura 7 apresenta a arquitetura desenvolvida por Maier e Ferens (2009).

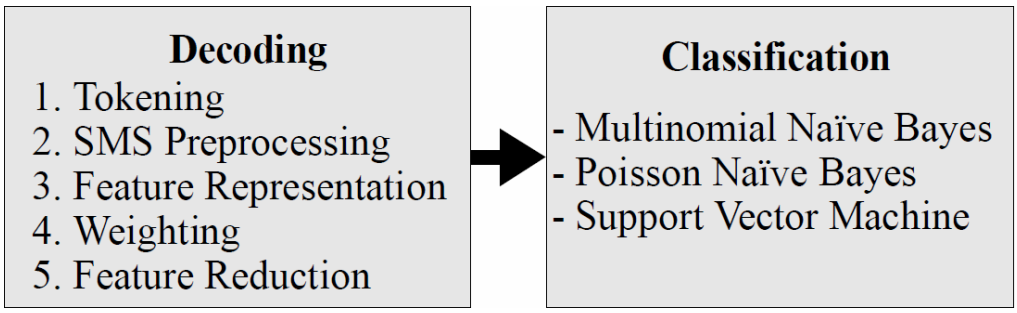

Figura 7. Arquitetura proposta Maier e Ferens
Fonte: Dados da pesquisa de Maier, Ferens (2009).

A pesquisa de Maier e Ferens (2009) apresenta os resultados entre os tipos de classificação baseado em 10.000 mensagens spam e 500 mensagens não spam. Além disso, foram utilizadas técnicas como *steamming*, que retira partes das palavras que possuem o mesmo sentido, como por exemplo compre e comprar, ficando apenas compr, e remoção palavras com menos de dois caracteres ou com apenas dois caracteres, além da remoção de *stopwords*.

A Tabela 3.5 apresenta os resultados obtidos nos experimentos realizados por Maier e Ferens (2009). Os algoritmos que possuem asterisco na frente do nome foram modificados conforme descritos na pesquisa de Maier e Ferens (2009). Com as alterações realizadas, os algoritmos não apresentaram nenhum ganho significativo, mas é possível perceber na tabela que o SVM e o Bayes Poisson obtiveram uma ligeira melhora.

Tabela 3.5 – Resultados obtidos na pesquisa de Maier e Ferens (2009)

Classificador	Precisão	Recall	F1-Measure
SVM	0,984	0,992	0,987
Multinomial NB	0,982	0,993	0,987
Poisson NB	0,980	0,992	0,985
*SVN	0,984	0,993	0,988
*Multinomial NB	0,982	0,993	0,987
*Poisson NB	0,984	0,991	0,987

O trabalho de Maier e Ferens (2009) apresentou uma comparação entre os algoritmos de classificação SMS: Multinomial NB, Poisson NB e SVM. As modificações realizadas no algoritmo não apresentaram nenhum ganho significativo no desempenho, apenas no espaço armazenado. Os próximos passos indicados para trabalhos futuros foram: obter uma base de mensagens maior, escolher esquemas de ponderação diferentes das realizadas nos testes e fazer uma análise sobre o tamanho do recurso utilizado para ambientes com recurso limitado.

3.3 Design and analysis of anti spamming SMS to prevent criminal deception and billing froud: Case Telkom Flexi

O trabalho apresentado por Asvial, Sirat e Susatyo (2008) tem como objetivo implementar uma barreira anti spam que realiza todo processo de análise, verificação e gerenciamento de listas negras de remetentes que enviam mensagens denominadas spam’s. Essa barreira minimiza o fluxo de mensagens SMS indesejadas na rede e fraudes de cobranças geradas por essas mensagens SMS. Segundo Asvial, Sirat e Susatyo (2008) as fraudes em sistemas SMS podem gerar milhões em prejuízo para as operadoras de telefonia celular. A estimativa proposta por Asvial, Sirat e Susatyo (2008) diminuiu em 87,32% das ações penais causadas por fraudes SMS, e em 95,35% nas perdas de faturamento.

3.4 Real-time Monitoring and Filtering System for Mobile SMS

O trabalho apresentado por Wu, Wu e Chen (2006) tem como objetivo apresentar uma proposta de filtragem de conteúdo em tempo real para mensagens SMS que utilizam o Mandarim padrão conhecido como Pinyn. Esta proposta divide o SMS em três categorias: mensagens SMS consideradas como não spam, mensagens SMS consideradas como spam e mensagens SMS consideradas como duvidosas. As mensagens trafegadas no filtro geram um dicionário de palavras-chave que são utilizadas para cálculos utilizando Classificador Bayesiano para gerar os dados estatísticos. A arquitetura proposta por Wu, Wu e Chen (2006), apresenta uma plataforma de filtragem de conteúdo das mensagens trafegadas no SMSC. Essa plataforma trabalha em paralelo com a criação dos dicionários de palavras-chave reduzindo, assim, o impacto do SMSC e a velocidade de processamento na classificação de mensagens.

3.5 Utilização de Redeis Neurais Artificiais para Classificação de Spam

No trabalho de Silva (2009), é apresentada a utilização de Redes Neurais Artificiais para a classificação de spam e o processo foi divido em três passos. O primeiro passo é a preparação das mensagens, onde foi realizada a remoção de *stopwords*, que consiste em descartar as palavras que pouco refletem o conteúdo de um documento ou são tão comuns que não distinguem nenhuma categoria dos documentos. Também foi realizada a técnica de *stemming*, onde várias palavras podem assumir formas variadas, como por exemplo: compra, comprar e comprei. O *stemming* consiste em remover os prefixos e sufixos. O resultado obtido é chamado de *Stem* (raiz ou radical) (Monteiro, Gomes e Oliveira, 2006). As palavras foram convertidas em minúsculas, os acentos foram removidos, e os *links*, imagens, anexos, endereços eletrônicos, moeda, porcentagem e palavras longas foram substituídos por uma palavra específica. As *tags* HTML (*HyperText Markup Language*) foram desconsideradas. Também foi realizado a uniformização do e-mail, realizando um processamento no HTML, realizando o processo de *tokenização* (processo de decompor a mensagem em termos) e reconhecimento de padrões (processo onde se identifica os padrões dos *spammers*).

O segundo passo foi a utilização de métodos estatísticos na extração das informações mais relevantes de um conjunto de dados, identificando as palavras que melhor representam uma categoria. A seleção das características utiliza o método chamado Distribuição por Freqüência. A Distribuição por Freqüência ou *DF* é considerada uma das técnicas mais

simples para redução da dimensionalidade. Ela possui uma complexidade computacional aproximadamente linear, possibilitando a utilização em grandes conjuntos de dados a um custo computacional relativamente pequeno.

O terceiro passo foi a criação de um vetor de características, onde foi empregado o método de Distribuição por Freqüência. O vetor de características é criado a partir de uma seleção de *n* características mais relevantes. No experimento de Silva (2009) foram gerados vetores com 25, 50 e 100 características. Cada característica corresponde a um nó RNA (Rede Neural Artificial) e cada mensagem é representada por um vetor $X = (x_1, x_2, ..., x_n)$, onde *n* é o número de características. Nos testes realizados foram analisadas as seguintes técnicas para compor o vetor: freqüência do termo, que indica o número de vezes que a palavra ocorreu no e-mail, peso binário, apresenta um para a palavra que aparece ao menos uma vez no e-mail e 0 para a que não aparecer, e peso normal, que indica o número de ocorrências da palavra que aparece mais vezes em um e-mail sendo escolhida como referência para o limite superior que é +0,5 e o limite inferior é -0,5. Com base nessas informações é feita a normalização dos valores do vetor característico.

3.6 Conclusão

Neste Capítulo foram introduzidas propostas envolvendo o uso de algoritmos na solução da filtragem de conteúdo de mensagens SMS e mensagens de e-mails.

Inicialmente foi apresentado a proposta de Deng e Peng (2006), que propõe um filtro de spam baseado em Classificador Bayesiano para sistemas SMS. Essa proposta é a que mais se assemelha a apresentada por este trabalho. A proposta de Maier e Ferens (2009) realiza uma comparação entre diferentes tipos de classificação. A proposta de Asvial, Sirat e Susatyo (2008), apresenta uma barreira anti spam que realiza o processo de análise da mensagem a partir de listas negras de remententes. Wu, Wu e Chen (2006), apresentam uma proposta de filtragem de conteúdo em tempo real para o Mandarim. E Silva (2009), utiliza Redes Neurais Artificiais para realizar o processo de identificação de spams em mensagens de e-mails.

4 IMPLEMENTAÇÃO DA CLASSIFICAÇÃO BAYESIANA EM UM SISTEMA SMS

4.1 Introdução

A proposta deste trabalho é a implementação de um filtro de conteúdo baseado em classificação Bayesiana para sistemas SMS e uma proposta de uma arquitetura para esse filtro. A Classificação Bayesiana foi escolhida devido à facilidade de implementação e de comparação com resultados de outros autores, Balinski(2009), Drucker, Donghui e Vapnik(1999), Cormack e Lynam(2009), Deepak e Sandeep(2005), Shahreza(2006), Lorena(2006), Wu, Wu e Chen(2006), Deng e Peng(2006), Andrade(2006), Ming, Yunchun e Wei(2007), He, Sun, Zheng e Wen(2008), Shahreza(2008), He, Wen e Zheng(2008), Asvial, Sirat e Susatyo(2008), Silva(2009), Zhang, Wen, He e Zheng(2009), Maier e Ferens(2009). O filtro implementado neste trabalho se difere dos propostos por outros autores por estar implementando dentro de uma ESME. Este trabalho propõe a utilização de apenas palavras e três atributos para a classificação dos filtros. A diferença entre esta proposta e o trabalho de Deng e Peng (2006), é a expansão do teste de probabilidades para agrupamento de palavras, para avaliação de escalabilidade.

4.2 Arquitetura proposta

A arquitetura proposta para esse trabalho é dividida em dois módulos. A Figura 8 apresenta o Módulo Filtro e o Módulo Dispositivo Móvel. O Módulo Filtro é dividido entre Filtro de Conteúdo, que realiza o filtro de cada uma das mensagens enviadas pela ESME, Aprendizado, que armazena os cálculos realizados no Filtro de Conteúdo, Treinamento, que é utilizado para realizar o treinamento das mensagens na fase de testes, e a Base de Dados, que armazena os dados enviadas pelo Aprendizado. O Módulo Dispositivo Móvel, filtra as novas mensagens enviadas ao dispositivo do cliente e envia as informações para o Módulo Filtro

melhorando cada vez mais as novas filtragens. Cada um desses Módulos serão explicados detalhadamente a seguir.

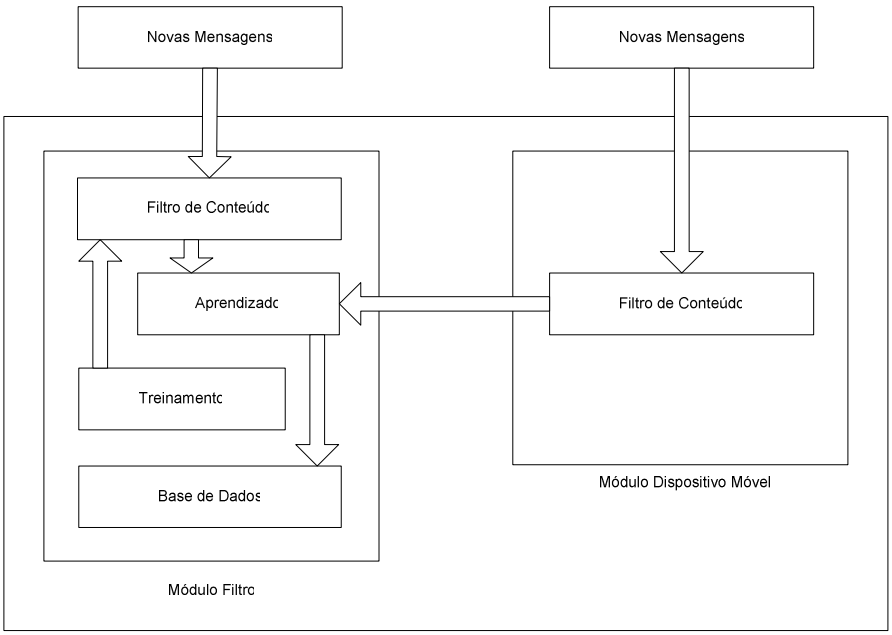

Figura 8. Arquitetura Proposta para o filtro de conteúdo

4.3 Implementação do Filtro

O filtro de spam deste trabalho, conforme mencionado anteriormente é semelhante ao proposto por Deng e Peng (2006). Resumidamente, ele passa por processos descritos a seguir: foi desenvolvida uma ESME e implantada em uma operadora de telefonia celular. Essa ESME recebe as mensagens enviadas pelos provedores de conteúdo e classifica as mensagens como spam e não spam antes de enviar à SMSC da operadora de telefonia celular. Caso a mensagem seja classificada como não spam, essa mensagem será encaminhada a SMSC da operadora que enviará essa mensagem ao dispositivo móvel do usuário. Foi desenvolvido também um aplicativo em Java para os dispositivos que suportam o sistema operacional Android, onde o usuário poderá classificar as mensagens como spam ou não spam que sua operadora enviou contribuindo com a base de dados do filtro.

O filtro implementado, trabalha sobre cada mensagem, para classifica-la como spam ou não spam, da seguinte forma: retiram-se os caracteres especiais, acentos, e *stopwords*,

transforma-se todos os caracteres em minúsculos. O conjunto de palavras restantes é armazenado em um vetor. Esse vetor é percorrido sequencialmente, pegando uma, duas, três, quatro, ou cinco palavras, de acordo com a opção feita antes de se iniciar o filtro. Todas as combinações de uma a cinco palavras são armazenadas no vetor. Cada elemento do vetor pode ser consequentemente uma, duas, três, quatro ou cinco palavras. Além disso, cala elemento pode conter um valor agregado, ou seja, um atributo. Para cada grupo de palavras (entre uma e cinco) o algoritmo do filtro, descrito na Seção 4.2.3, é executado para classificar a mensagem.

Inicialmente foram obtidas 120.000 mensagens de uma ESME implantada em uma operadora de telefonia celular. As 120.000 mensagens foram classificadas manualmente como spam e não spam de acordo com o caso. Todas as palavras de cada mensagem foram listadas e agrupadas até no máximo cinco palavras. Foi estabelecida uma probabilidade de cada uma dessas palavras e grupos de palavras aparecerem em uma mensagem spam e não spam, através da fórmula de Bayes. Por exemplo, ao se receber uma mensagem com a palavra “compre” a maioria dos usuários consideraria essa mensagem como spam, e, raramente, encontraria essa palavra em uma mensagem considerada não spam. Por isso foi necessário treinar o filtro para que fosse possível identificar a probabilidade de uma mensagem com a palavra “compre” spam. O mesmo foi realizado para o grupo de duas, três e quatro palavras, como por exemplo “compre agora”, “compre agora carro” e “compre agora carro imperdível”. Esse treino foi feito com 25.000 mensagens spam e 95.000 mensagens não spam, totalizando 1.405.285 grupos de uma a cinco palavras. Durante esse treino o filtro ajustou as probabilidades de cada uma das palavras e grupos encontrados nas mensagens, de acordo com a categoria, spam e não spam.

Foi realizada a remoção de *stopwords,* que consiste em descartar as palavras que pouco refletem o conteúdo de um documento ou são tão comuns que não distinguem nenhuma categoria dos documentos, como por exemplo, artigos e preposições. Cada idioma possui uma lista de palavras consideradas *stopwords*. A lista para o idioma português foi obtida em Balinski (Balinski 2002) e encontra-se no Anexo A. Para Marques (2009), nem todas as palavras são igualmente significativas para representar uma categoria. Algumas carregam mais significado que outras. Normalmente os substantivos, seguidos dos adjetivos e verbos carregam mais representatividade do que outras classes gramaticais como os pronomes, as conjunções e os artigos. A remoção das *stopwords* consiste em descartar as palavras que pouco refletem no conteúdo de um documento ou são tão comuns que não distinguem

nenhuma categoria dos documentos. Todas as palavras foram convertidas para minúsculas e acentos e caracteres especiais foram retirados.

Os atributos utilizados nessa pesquisa foram: grupos de uma a cinco palavras, telefones, URL's e valores monetários. Os demais atributos da pesquisa de Deng e Peng (2006) foram desconsiderados, como o tamanho da mensagem e o remetente. Esses atributos foram desconsiderados porque as mensagens utilizadas nessa pesquisa são mensagens enviadas por provedores de conteúdo. Essas mensagens possuem o tamanho maior quando se compara com mensagens enviadas de assinante para assinante, e os remetentes sempre são os mesmos. Cada um desses atributos selecionados contribui para o que se chama de probabilidade posterior no cálculo do teorema de Bayes. Em seguida, a probabilidade é calculada sobre todos os atributos uma mensagem. O cálculo é realizado para ambos os casos, tanto para a mensagem ser spam, quanto para a mensagem não ser spam. A classificação é determinada a partir do maior valor das probabilidades de spam e não spam. Além disso, o filtro estará em constante atualização, a partir da chegada de novas mensagens, as probabilidades de cada uma das palavras são atualizadas, deixando, assim, cada vez mais precisa a detecção de mensagens spam e não spam.

Abaixo será apresentado o trabalho realizado em uma mensagem classificada manualmente como não spam. O filtro separa cada uma das palavras em grupos de uma a cinco palavras.

"O ministro da Fazenda, Guido Mantega, avaliou que o crédito continuará restrito em função da atual crise financeira global."

A Tabela 4.1 apresenta um exemplo de grupos de até três palavras após a conversão para minúsculo, remoção de *stopwords*, acentos e caracteres especiais. Também é possível perceber nessa tabela o cálculo da probabilidade para cada um dos elementos do vetor de ser spam ou não spam. O resultado apresentado ao final da tabela determina como a mensagem foi classificada. Neste caso, como a probabilidade da mensagem ser não spam é maior que a probabilidade da mensagem ser spam, essa mensagem foi classificada como não spam.

Tabela 4.1 – Resultados obtidos na classificação de uma mensagem

Elemento	Probabilidade Spam	Probabilidade Não Spam
atual	0,177420	0,82258
atual crise	0,010000	0,99000
atual crise financeira	0,010000	0,99000
avaliou	0,010000	0,99000

Avaliou credito	0,010000	0,99000
avaliou credito continuara	0,010000	0,99000
continuara	0,010000	0,99000
continuara restrito	0,010000	0,99000
continuara restrito funcao	0,010000	0,99000
credito	0,372090	0,62791
credito continuara	0,010000	0,99000
credito continuara restrito	0,010000	0,99000
crise	0,175130	0,82487
crise financeira	0,096770	0,90323
crise financeira global	0,010000	0,99000
fazenda	0,262860	0,73714
fazenda guido	0,195120	0,80488
fazenda guido mantega	0,153850	0,84615
financeira	0,116070	0,88393
financeira global	0,010000	0,99000
funcao	0,310340	0,68966
funcao atual	0,010000	0,99000
funcao atual crise	0,010000	0,99000
global	0,209300	0,79070
guido	0,275860	0,72414
guido mantega	0,254550	0,74545
guido mantega avaliou	0,010000	0,99000
mantega	0,448280	0,55172
mantega avaliou	0,010000	0,99000
mantega avaliou credito	0,010000	0,99000
ministro	0,276020	0,72398
ministro fazenda	0,195120	0,80488
ministro fazenda guido	0,195120	0,80488
restrito	0,010000	0,99000
restrito funcao	0,010000	0,99000
restrito funcao atual	0,010000	0,99000
Resultado	**0,000000**	**0,01064**

A Tabela 4.2 apresenta os resultados para uma palavra, ideia semelhante à proposta por Deng e Peng (2006), utilizando a mesma mensagem. Para Deng e Peng (2006) o tamanho da mensagem é um atributo relevante. Ou seja, essa mensagem pode ser considerada spam por possuir uma quantidade de caracteres muito alta.

Tabela 4.2 – Resultados obtidos na classificação de uma mensagem segundo a pesquisa de Deng e Peng (2006)

Elemento	Probabilidade Spam	Probabilidade Não Spam
atual	0,17742	0,82258
avaliou	0,01000	0,99000
continuara	0,01000	0,99000
credito	0,37209	0,62791
crise	0,17513	0,82487
fazenda	0,26286	0,73714
financeira	0,11607	0,88393
funcao	0,31034	0,68966
global	0,20930	0,79070
guido	0,27586	0,72414
mantega	0,44828	0,55172
ministro	0,27602	0,72398
restrito	0,01000	0,99000
Resultado	**0,00000**	**0,04249**

4.3.1 Utilização do Teorema de Bayes

O teorema de Bayes foi utilizado várias vezes na análise de uma mensagem. Na primeira vez, foi usado para calcular a probabilidade de uma mensagem ser spam, sabendo-se que uma determinada palavra ou grupos de palavras aparecem na mensagem. Na segunda, foi usado para se calcular a probabilidade de uma mensagem ser spam, levando-se em consideração todas as palavras, grupos de palavras ou atributos da mesma. Na terceira, para se trabalhar com palavras raras. Nesse caso, foi atribuído o valor de 50% de chance de a palavra pertencer à classe spam e 50% para classe não spam.

Como exemplo, consideramos a palavra “compre”. Muitas pessoas, ao receber uma mensagem com essa palavra, a identificariam como spam, por associarem essa palavra a uma oferta ou propaganda sobre o produto. A partir da Fórmula (1), vamos calcular a probabilidade dela ser ou não spam.

$$P(S \mid W) = \frac{P(W \mid S) \cdot P(S)}{P(W)} \qquad (1)$$

Em (1), $P(S|W)$ é a probabilidade de a mensagem ser spam, sabendo-se que a palavra "Compre" está na mensagem, $P(W|S)$ é a probabilidade da palavra "Compre" aparecer em uma mensagem spam, $P(S)$ é a probabilidade de uma mensagem ser spam e $P(W)$ é a probabilidade de uma mensagem ser legítima.

Para o cálculo apenas da palavra "compre" numa mensagem, pode ocorrer em erro na identificação se a mensagem é spam ou não spam, por isso é necessário realizar o cálculo para várias palavras e atributos, obtendo-se, assim, um resultado mais confiável. Para se realizar o cálculo de várias palavras utilizamos a Fórmula anterior (1) calculando a probabilidade de cada palavra encontrada, e aplicamos o resultado na seguinte Fórmula (2):

$$P(S|W_i) = \prod_{t=1}^{n} P(s_t|W_i) \qquad (2)$$

Em (2), $P(S|W_i)$ representa o resultado que se quer encontrar, $P(s_t|W_i)$ a probabilidade de cada uma das palavras, grupos de palavras e atributos calculados pela Fórmula (1). No caso de palavras raras, ou seja, não encontradas na base de dados de palavras, o valor para $P(s_t|W_i)$ será de 50%. A definição de se uma mensagem é spam ou não spam, é realizada pelo maior resultado dos cálculos realizados para a mensagem ser spam ou não spam.

A Tabela 4.3 apresenta alguns exemplos de palavras e grupos de palavras classificadas no treinamento. Nessa tabela é possível perceber que a probabilidade da palavra "crise" não ser considerada spam vai aumentando à medida que se aumenta a quantidade de palavras, o mesmo ocorre na probabilidade da palavra "promocao" ser considerada como spam.

Tabela 4.3 – Probabilidade das palavras e grupos que mais apareceram em mensagens

Elementos	**Probabilidade Spam**	**Probabilidade Não Spam**
crise	0,132	0,868
crise financeira	0,097	0,903
crise financeira global	0,001	0,999
crise financeira global garantir	0,001	0,999
promocao	0,661	0,339
promocao geladeira	0,999	0,001
promocao geladeira recheada	0,999	0,001
promocao geladeira recheada loja	0,999	0,001

4.4 Descrição do Algoritmo implementado para o Sistema Operacional Android

O algoritmo desenvolvido para ser executado nos celulares que possuem o sistema operacional Android, desenvolvido na linguagem Java SDK 1.5 do Android, utilizando banco de dados SQLite. Nesta parte do projeto as pessoas poderão contribuir com as mensagens que recebem em seus celulares, indicando se são ou não spam's. Quando a aplicativo é baixado e instalado, o mesmo lê todas as mensagens na caixa de entrada do celular do cliente e envia ao servidor os dados atualizados para serem incluídos na base de dados central do filtro. A partir desse momento, a cada nova mensagem que chegar ao celular, novas probabilidades são calculadas e novos dados são enviados ao banco de dados do servidor. A Figura 9 apresenta o fluxograma da leitura de mensagens na caixa de entrada do celular do usuário. As mensagens lidas na caixa de entrada serão enviadas e classificadas como não spam para o servidor do filtro de conteúdo para melhoria do filtro, além de atualizar a lista de palavras e grupos de palavras considerados não spam do próprio celular.

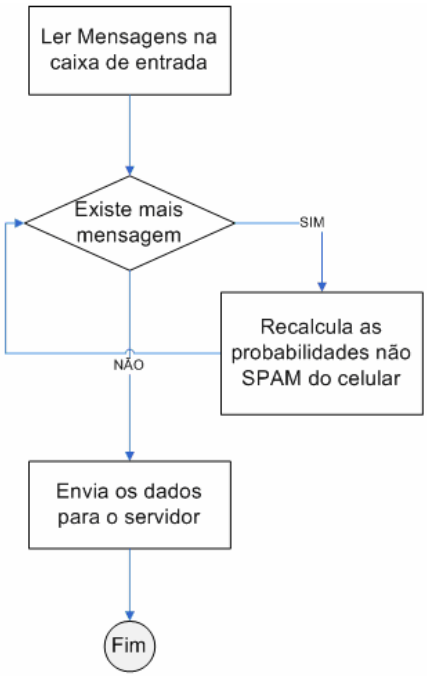

Figura 9. Fluxograma da leitura da caixa de entrada

A Figura 10 apresenta o fluxograma da leitura de mensagens na caixa de excluídos do celular do usuário. Ou seja, mensagens que os usuários leram e excluíram. As mensagens lidas na caixa de excluídos serão enviadas e classificadas como spam para o servidor do filtro de conteúdo para melhoria do filtro, além de atualizar a lista de palavras e grupos de palavras considerados não spam do próprio celular.

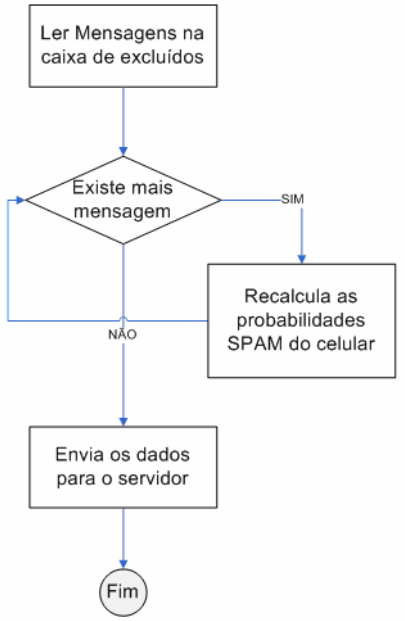

Figura 10. Fluxograma da leitura da caixa de excluídos

A Figura 11 apresenta o fluxograma da leitura de mensagens de novas mensagens do celular do usuário. As novas mensagens serão enviadas ao servidor do filtro e classificadas. O usuário será consultado sobre como ele classifica a mensagem para que seja confirmada a classificação do filtro. Após essa confirmação, os dados serão enviados para o servidor para que seja atualizada a base do filtro de conteúdo.

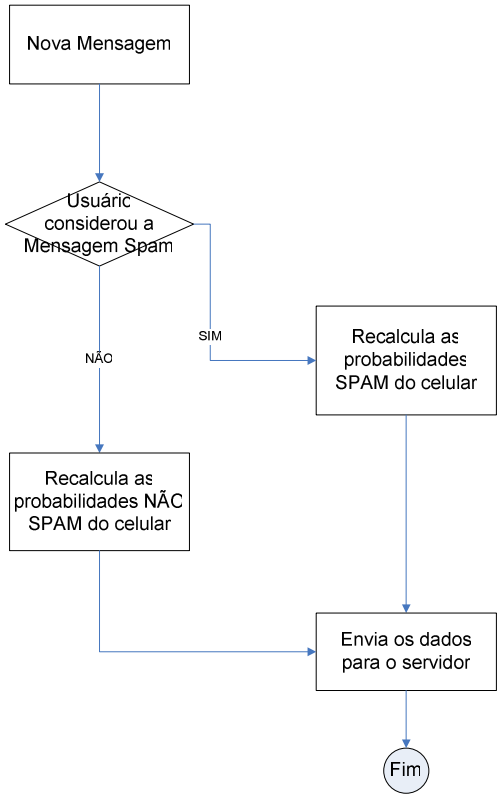

Figura 11. Fluxograma de novas mensagens no celular

4.5 Descrição do Algoritmo do Servidor

O algoritmo do servidor é a parte principal desse projeto. Ele contém toda a implementação do filtro de conteúdo. A cada nova mensagem que o algoritmo receber, são separados os grupos de uma a cinco palavras e atributos. Após essa separação é realizado o cálculo da probabilidade dessa mensagem ser ou não spam. Para cada grupo de palavras e atributos obtidos em uma mensagem, é realizada uma consulta na base de dados do filtro, identificando a incidência destes elementos como spam ou não spam. Posteriormente é realizado o cálculo, utilizando a Fórmula (1) a seguir:

$$P(S \mid W) = \frac{P(W \mid S) \cdot P(S)}{P(H)} \qquad (1)$$

A Fórmula (4) foi transformada nos seguintes trechos de código demonstrados nos Algoritmo 4.1 e Algoritmo 4.2 abaixo e explicados a seguir:

Algoritmo 4.1 – Cálculo para cada elemento da mensagem não ser spam

```
1: função calculaProbabilidadeNaoSpam(elemento)
2:         quantidadeSpam = consultaQuantidadeSpam(elemento);
3:         quantidadeNaoSpam = consultaQuantidadeNaoSpam(elemento);
4:         se quantidadeSpam = 0 e quantidadeNaoSpam = 0 então
5:                 retorna 0,5;
6:         senão
7:                 se quantidadeNaoSpam = 0 então
8:                         retorna 0,01;
9:                 senão
10:                        probabilidade = quantidadeNaoSpam / (quantidadeSpam +
                           quantidadeNaoSpam);
11:                        se probabilidae = 1 então
12:                                retorna 0,99;
13:                        senão
14:                                retorna probabilidade;
15:                        fim-se;
16:                fim-se;
17:        fim-se;
18: fim-função;
```

Algoritmo 4.2 – Cálculo para cada elemento da mensagem ser spam

```
1: função calculaProbabilidadeSpam(elemento)
2:         quantidadeSpam = consultaQuantidadeSpam(elemento);
3:         quantidadeNaoSpam = consultaQuantidadeNaoSpam(elemento);
4:         se quantidadeSpam = 0 e quantidadeNaoSpam = 0 então
5:                 retorna 0,5;
6:         senão
7:                 se quantidadeSpam = 0 então
8:                         retorna 0,01;
9:                 senão
10:                        probabilidade = quantidadeSpam / (quantidadeSpam + quantidadeNaoSpam);
11:                        se probabilidae = 1 então
12:                                retorna 0,99;
13:                        senão
14:                                retorna probabilidade;
```

```
15:                         fim-se;
16:                 fim-se;
17:         fim-se;
18: fim-função;
```

O Algoritmo 4.1 calcula a probabilidade de um elemento da mensagem não ser spam, caso o resultado for igual a 0, o valor retornado será 0,01 para não zerar o cálculo, e se o valor for igual a 1, o valor retornado será 0,99. O Algoritmo 4.2 calcula a probabilidade de um elemento da mensagem ser spam, caso o resultado for igual a 0, o valor retornado será 0,01 para não zerar o cálculo, e se o valor for igual a 1, o valor retornado será 0,99. Nos Algoritmos 4.1 e 4.2 foram desenvolvidos para tratar novos elementos, caso não exista na base de dados do filtro o retorno para a probabilidade será de 0,5.

Após realizado o cálculo de cada um dos elementos da mensagem é realizado o cálculo da mensagem inteira ser spam, utilizando a Fórmula (2) abaixo.

$$P(S \mid W_i) = \prod_{t=1}^{n} P(s_t \mid W_i) \qquad (2)$$

A Fórmula (2) é transformada no seguinte trecho de código demonstrado no Algoritmo 4.3 abaixo e explicado a seguir:

Algoritmo 4.3 – Verifica se mensagem é spam

```
1: função mensagemSpam(mensagem)
2:          vetor = transformaMensagem(mensagem);
3:          probabilidadeSpam = 1;
4:          probabilidadeNaoSpam = 1;
5:          para todo (elemento do vetor) faça
6:                  probabilidadeElemSpam = calculaProbabilidadeSpam(elemento);
7:                  probabilidadeElemNaoSpam = calculaProbabilidadeNaoSpam(elemento);
8:                  probabilidadeSpam = probabilidadeSpam * probabilidadeElemSpam;
9:                  probabilidadeNaoSpam = probabilidadeNaoSpam * probabilidadeElemNaoSpam;
10:         fim-para;
11:         retorno probabilidadeSpam > probabilidadeNaoSpam;
12: fim-função;
```

O Algoritmo 4.3 transforma inicialmente a mensagem em um vetor de elementos com grupos de palavras e atributos, após essa transformação é realizado um cálculo para cada um dos elementos, calculando a probabilidade de ser ou não spam. O resultado é obtido do valor que for maior. Ou seja, se o valor da probabilidade de da mensagem ser spam for maior que a probabilidade da mensagem não ser spam, então essa mensagem será considerada spam. Caso o valor da probabilidade da mensagem não ser spam for maior, essa mensagem será considerada não spam.

O desenvolvimento do algoritmo ficou dividido em três módulos. No primeiro módulo, denominado *Treinamento*, onde foram inseridas ao algoritmo uma lista de mensagens, spam e não spam. As probabilidades de cada um dos atributos e grupos de uma a cinco palavras são calculadas, classificando cada uma delas como spam ou não spam. No segundo módulo, denominado *Classificação*, o algoritmo recebe uma mensagem e a classifica como spam ou não spam. No terceiro módulo, denominado *Aprendizado*, os usuários que utilizarem o aplicativo no sistema operacional Android poderão contribuir com o filtro a partir das classificações realizadas por ele. O módulo *Treinamento* pode ser entendido melhor no fluxograma da Figura 12 e explicado melhor posteriormente.

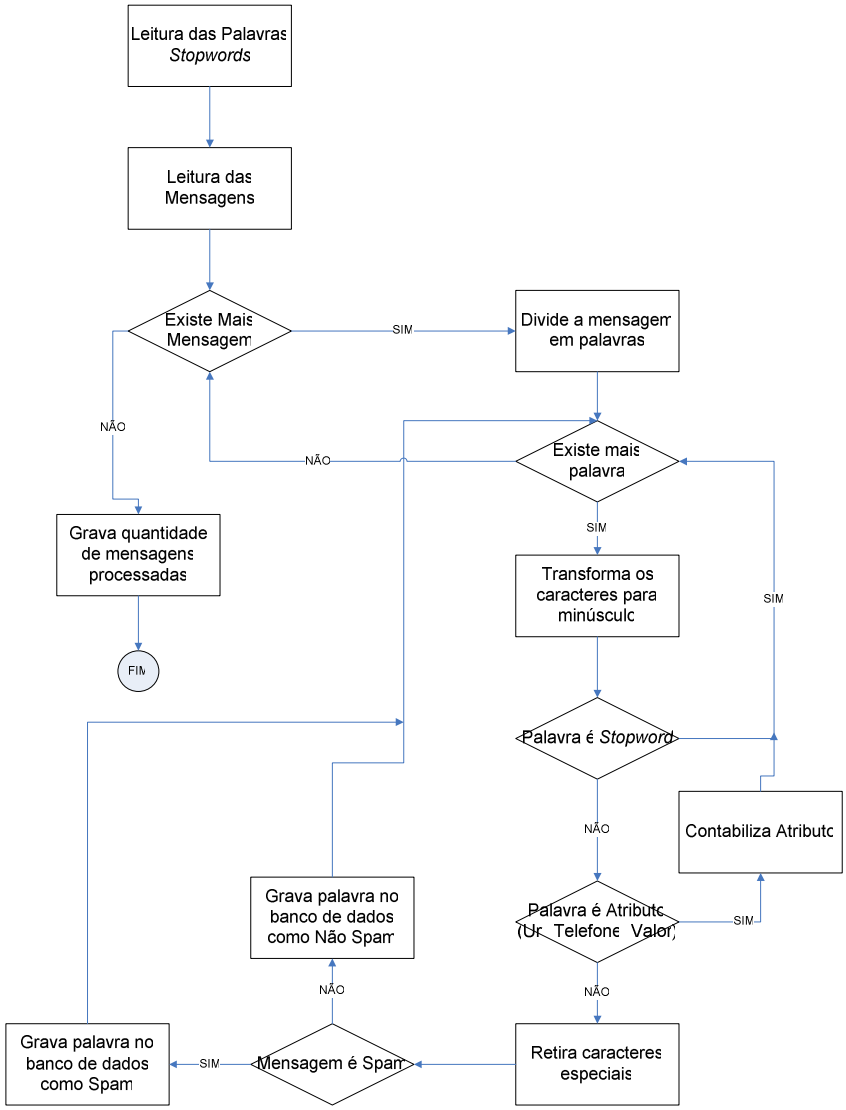

Figura 12. Fluxograma do módulo Treinamento

O módulo *Treinamento* apresentado na Figura 12 apresenta o fluxograma de treinamento das mensagens utilizadas. Inicialmente é realizado a carregamento dos *stopwords* na memória do servidor para utilização futura. Para cada mensagem lida é realizado o processo de preparação da mensagem, onde todos os caracteres especiais, acentos e *stopwords* são retirados, e é verificado se na mensagem existe algum atributo, como telefone, URL e valores monetários. Após a preparação da mensagem é utilizada a função descrita no Algoritmo 4.3, a partir do resultado dessa função os dados são gravados no banco de elementos do filtro.

O módulo *Classificação* e *Aprendizado* pode ser entendido melhor no fluxograma da Figura 13 e explicado melhor posteriormente.

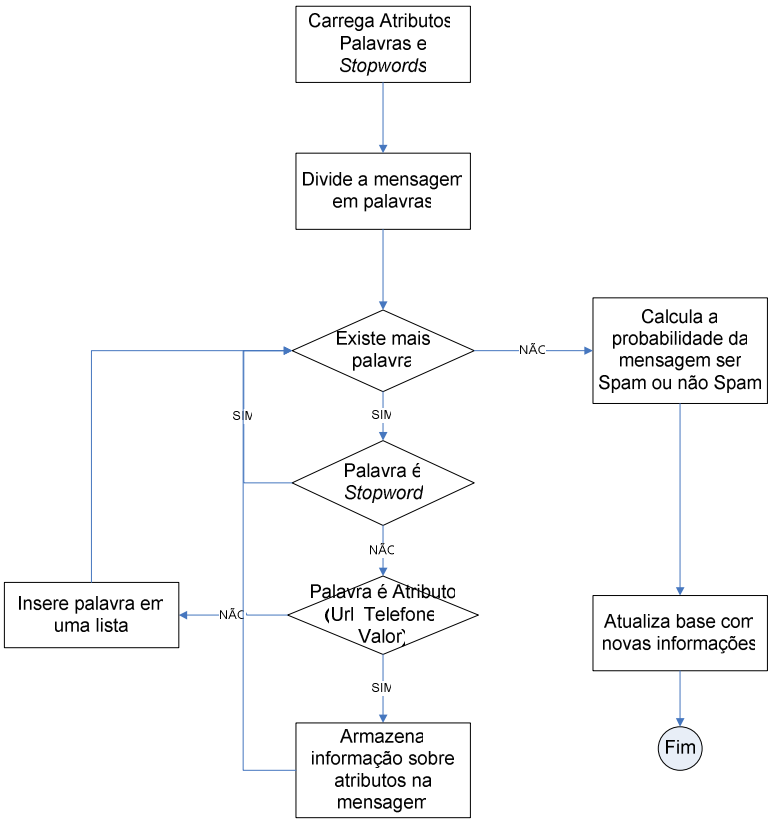

Figura 13. Fluxograma do módulo Classificação e Aprendizado

O módulo *Classificação* e *Aprendizado* apresentado na Figura 13 apresenta o fluxograma de classificação da mensagem enviada. Inicialmente é realizado a carregamento dos *stopwords*, atributos e elementos na memória do servidor para utilização futura. A mensagem é preparada da mesma forma como demonstrado no processo da Figura 12, onde todos os caracteres especiais, acentos e *stopwords* são retirados, e é verificado se na mensagem existe algum atributo, como telefone, URL e valores monetários. Após a preparação da mensagem é utilizada a função descrita no Algoritmo 4.3, a partir do resultado dessa função os dados são gravados no banco de elementos do filtro.

5 EXPERIMENTOS E RESULTADOS

5.1 Introdução

Os experimentos foram realizados no filtro de conteúdo após a carga de 120.000 mensagens que foi realizado no processo de *Treinamento*. Dessas 120.000 mensagens, 25.000 mensagens foram classificadas manualmente como spam e 95.000 mensagens foram classificadas manualmente como não spam. Essa carga realizada gerou 1.405.285 elementos, além dos atributos telefone, URL e valores monetários. Para realizar os experimentos foram obtidas mais 8.687 mensagens. Essas mensagens também foram classificadas manualmente, e dentre elas 8008 mensagens foram classificadas como não spam e 636 mensagens foram classificadas como spam. Essa classificação manual foi realizada para que após a obtenção dos resultados os mesmos possam ser conferidos e verificando assim o percentual de acerto do algoritmo. Os experimentos foram divididos em cinco partes. Na primeira parte dos testes as 8.687 mensagens foram classificadas considerando apenas uma palavra e os atributos. Na segunda parte dos testes as 8.687 mensagens foram classificadas considerando o grupo de uma a duas palavras, e os atributos. Na terceira parte dos testes as 8.687 mensagens foram classificadas considerando o grupo de uma a três palavras, e os atributos. Na quarta parte dos testes as 8.687 mensagens foram classificadas considerando o grupo de uma a quatro palavras, e os atributos. Na quinta parte dos testes as 8.687 mensagens foram classificadas considerando o grupo de uma a cinco palavras, e os atributos.

Para realizar os experimentos do filtro de spam SMS, foi utilizado o equipamento representado na Tabela 5.1.

Tabela 5.1 – Características principais do equipamento dos testes.

Característica	**Valor**
Modelo do processador	Intel(R) Xeon(R)
Quantidade de processadores	2
Freqüência de operação	2.0 GHz
Memória principal	4 GB

Memória secundária	350 GB
Sistema operacional	Ubuntu 9.04
Banco de dados	PostgreSQL 8.4

5.2 Resultados e Análises

A Tabela 5.2 apresenta os resultados para classificação de uma palavra e atributos. O percentual de falsos positivos foi de 0,045% e o percentual de falsos negativos foi 0,044%. A performance total do algoritmo foi de 99,955% de acerto. Nessa parte do testes foram utilizados 69.515 grupos de uma palavra. O espaço utilizado para esses grupos foi de 0,49MB.

Tabela 5.2 – Resultados para classificação de uma palavra e atributos

Tempo em segundos do teste	53,815
Tempo gasto por mensagem em segundos	0,006
Número de mensagens	8.687
Quantidade de mensagens spam	8.008
Quantidade de mensagens não spam	636
Falso positivo	29
Falso negativo	360
Grupos utilizados	69.515
Memória (MB)	0,49

A Tabela 5.3 apresenta os resultados para classificação do grupo de uma a duas palavras e atributos. O percentual de falsos positivos foi de 0,191% e o percentual de falsos negativos foi 0,007%. A performance total do algoritmo foi de 99,978% de acerto. Nessa parte do testes foram utilizados 366.910 grupos de uma a duas palavras. O espaço utilizado para esses grupos foi de 4,38MB.

Tabela 5.3 – Resultados para classificação do grupo de uma a duas palavras e atributos

Tempo em segundos do teste	99,041

Tempo gasto por mensagem em segundos	0,011
Número de mensagens	8.687
Quantidade de mensagens spam	8.008
Quantidade de mensagens não spam	636
Falso positivo	122
Falso negativo	64
Grupos utilizados	366.910
Memória (MB)	4,38

A Tabela 5.4 apresenta os resultados para classificação do grupo de uma a três palavras e atributos. O percentual de falsos positivos foi de 0,209% e o percentual de falsos negativos foi 0,005%. A performance total do algoritmo foi de 99,9799% de acerto. Nessa parte do testes foram utilizados 717.550 grupos de uma a três palavras. O espaço utilizado para esses grupos foi de 11,05MB.

Tabela 5.4 – Resultados para classificação do grupo de uma a três palavras e atributos

Tempo em segundos do teste	131,117
Tempo gasto por mensagem em segundos	0,015
Número de mensagens	8.687
Quantidade de mensagens spam	8.008
Quantidade de mensagens não spam	636
Falso positivo	133
Falso negativo	42
Grupos utilizados	717.550
Memória (MB)	11,05

A Tabela 5.5 apresenta os resultados para classificação do grupo de uma a quatro palavras e atributos. O percentual de falsos positivos foi de 0,218% e o percentual de falsos negativos foi 0,005%. A performance total do algoritmo foi de 99,9792% de acerto. Nessa parte do testes foram utilizados 1.073.356 grupos de uma a quatro palavras. O espaço utilizado para esses grupos foi de 19,95MB.

Tabela 5.5 – Resultados para classificação do grupo de uma a quatro palavras e atributos

Tempo em segundos do teste	148,856
Tempo gasto por mensagem em segundos	0,017

Número de mensagens	8.687
Quantidade de mensagens spam	8.008
Quantidade de mensagens não spam	636
Falso positivo	139
Falso negativo	42
Grupos utilizados	1.073.356
Memória (MB)	19,95

A Tabela 5.6 apresenta os resultados para classificação do grupo de uma a cinco palavras e atributos. O percentual de falsos positivos foi de 0,220% e o percentual de falsos negativos foi 0,005%. A performance total do algoritmo foi de 99,9790% de acerto. Nessa parte do testes foram utilizados 1.405.285 grupos de uma a quatro palavras. O espaço utilizado para esses grupos foi de 30,32MB.

Tabela 5.6 – Resultados para classificação do grupo de uma a cinco palavras e atributos

Tempo em segundos do teste	181,827
Tempo gasto por mensagem em segundos	0,020
Número de mensagens	8.687
Quantidade de mensagens spam	8.008
Quantidade de mensagens não spam	636
Falso positivo	140
Falso negativo	42
Grupos utilizados	1.405.285
Memória (MB)	30,32

Com esses dados é possível perceber que quanto maior o grupo de palavras utilizadas nos testes, maior o tempo gasto para realizar as classificações das mensagens. No gráfico da Figura 14 é possível perceber esse aumento do tempo gasto para realizar a classificação a partir dos grupos de palavras utilizados.

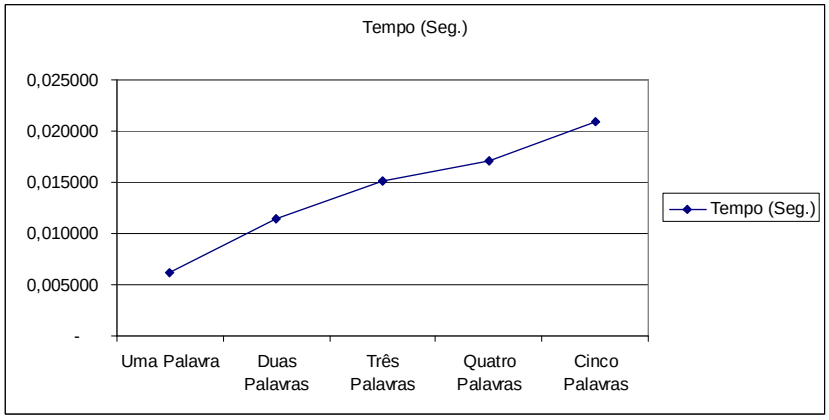

Figura 14. Tempo gasto para grupos de uma a cinco palavras por mensagem

Um outro resultado que é possível perceber nesses números é o aumento de falsos positivos à medida que se aumenta a quantidade de palavras. A Figura 15 apresenta esses resultados.

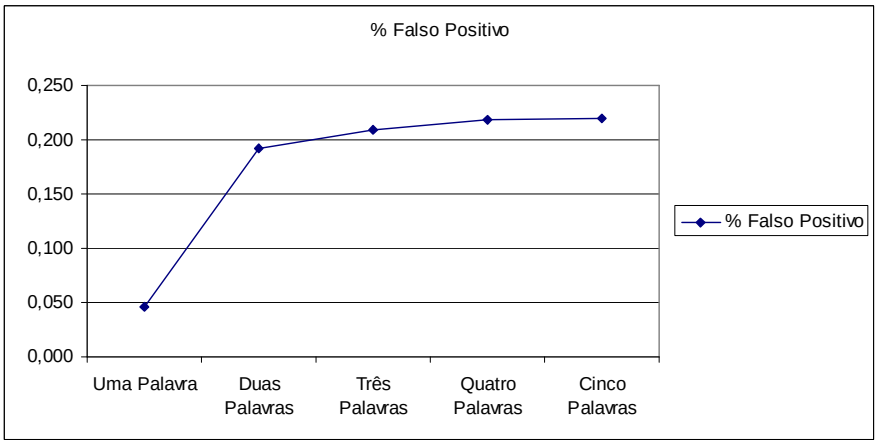

Figura 15. Percentual de falsos positivos na classificação do grupo de uma a cinco palavras

O número de falsos negativos diminui a medida que o número de palavras aumenta estabilizando no grupo de três, quatro e cinco palavras. A Figura 16 representa esses números.

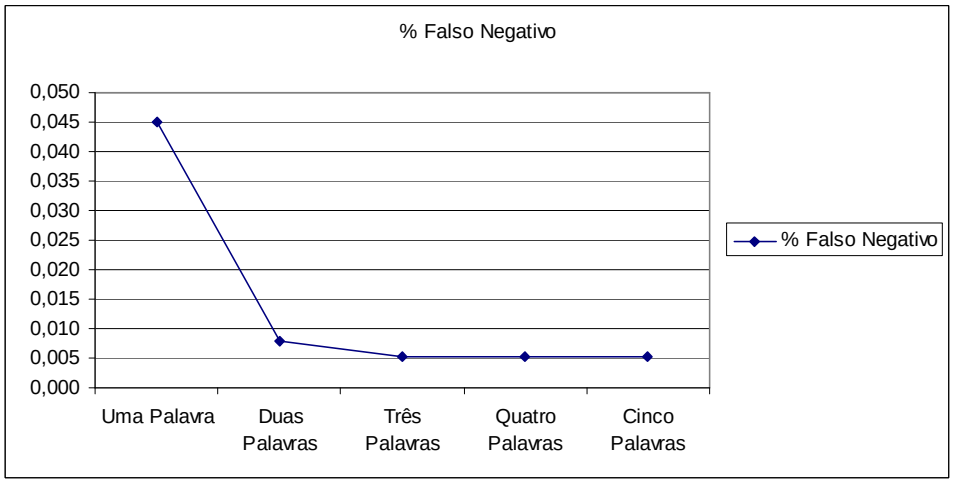

Figura 16. Percentual de falsos negativos na classificação do grupo de uma a cinco palavras

Conforme demonstrado na Figura 17, é possível perceber que a performance do filtro foi mais eficiente quando foi utilizado do grupo de até três palavras e atributos. Também vale destacar a diferença na utilização do grupo de uma palavra e o grupo de até duas palavras, onde o percentual de acerto aumentou significativamente. Quando foi utilizado o grupo de até quatro e cinco, a performance teve uma pequena queda.

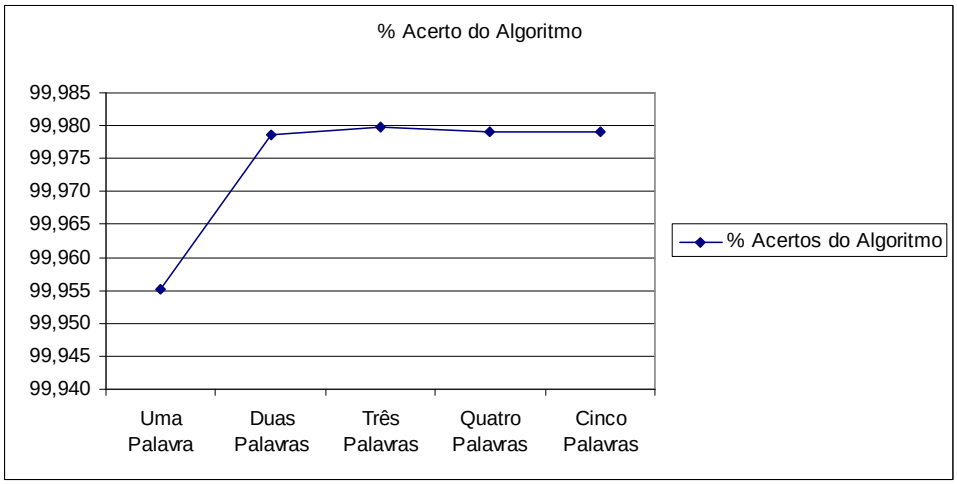

Figura 17. Percentual de acerto do algoritmo para o grupo de uma a cinco palavras

Além dos testes realizandos com 120.000 mensagens, foram realizados testes com 10.000 mensagens para uma comparação com a quantidade de mensagens utilizadas por Deng e Peng (2006). Dentre as 10.000 mensagens, 5.000 mensagens são spam e 5.000 mensagens

não spam. A Figura 18, apresenta os resultados obtidos com o treinamento com 10.000 mensagens e testes com 8.687 mensagens.

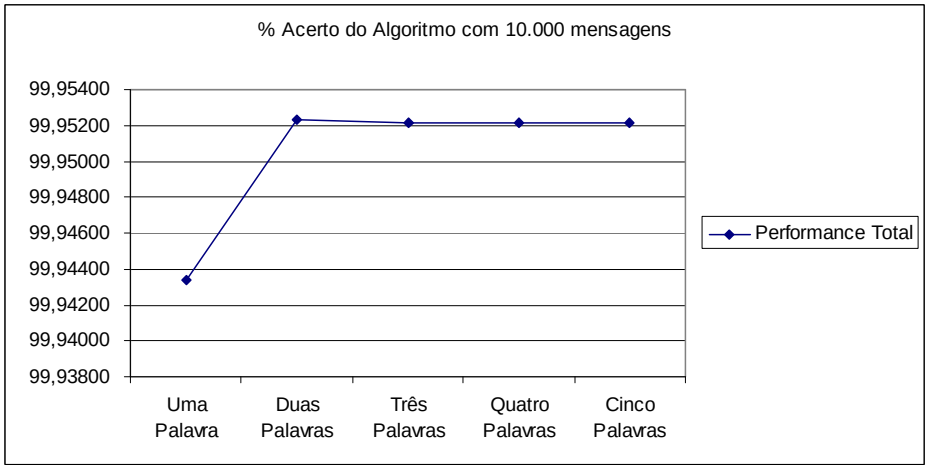

Figura 18. Percentual de acerto do algoritmo para o grupo de uma a cinco palavras com testes realizados com 10.000 mensagens

6 CONCLUSÕES E TRABALHOS FUTUROS

A principal contribuição dessa pesquisa foi a implementação de um filtro de conteúdo para sistemas SMS baseado em Classificador Bayesiano com agrupamento de palavras. Com este tipo de agrupamento de palavras, foi possível melhorar o percentual de acerto do algoritmo em comparação com a proposta de Deng e Peng (2006). Este trabalho foi motivado, principalmente, pela necessidade de se detectar e bloquear spam's em ambientes reais de operadoras de telefonia celular. Como os testes foram realizados sobre mensagens reais, de uma operadora de telefonia celular, os resultados indicam que o filtro apresentado é altamente eficaz e consegue identificar e bloquear spam's com quase 100% de acerto. Consideram-se os resultados alcançados plenamente satisfatórios e o filtro totalmente possível de ser implementado em redes reais. Como o tempo de processamento do filtro proposto é irrisório, tendo vista que se trata de um filtro de conteúdo para uma ESME de uma operadora de telefonia celular. Sendo assim, o custo computacional para um operadora de celular será mínimo, enquanto os benefícios serão inumeráveis.

O filtro de conteúdo proposto visa detectar spam's em mensagens SMS. Ele é baseado no trabalho de Deng e Peng (2006). A diferença da proposta está na implementação do algoritmo. Enquanto Deng e Peng (2006) usam apenas uma palavra, o filtro proposto permite a combinação de uma a cinco palavras. Os testes para detecção de spam's em um ambiente real foram feitos para 120.000 mensagens, enquanto Deng e Peng (2006) testaram apenas para 10.000 mensagens.

Os resultados mostram que a utilização de grupos de palavras tornou o filtro mais eficiente na classificação de mensagens spam e não spam, principalmente para grupos de duas ou três palavras. A utilização de quatro e cinco palavras não obteve um resultado superior. Comparando os resultados com a pesquisa de Deng e Peng (2006) o filtro desenvolvido para o grupo três palavras obteve um melhor resultado. O índice de acerto do filtro implementado chegou a 99,9799%, enquanto a proposta de Deng e Peng (2006) obteve 99,1% de acerto.

Quatro trabalhos futuros podem ser citados: (1) Extensão do filtro para todas as mensagens trafegadas na SMSC da operadora com o agrupamento de palavras, adicionando os atributos remetente e tamanho da mensagem, propostos por Deng e Peng (2006). (2) Implementação de um filtro de conteúdo com diversas classes. Dessa forma, o filtro ao invés

de classificar as mensagens com apenas as classes spam e não spam poderá classificar as mensagens por assunto, como por exemplo: promoção, esporte, financias, humor, políticas e religiosas. Dessa forma o usuário poderá liberar ou bloquear um determinado assunto. (3) Implementação de um filtro de conteúdo com classes por faixa etária. Dessa forma, o filtro, ao invés de classificar as mensagens com apenas as classes spam e não spam, poderá classificar as mensagens em faixas etárias dos usuários dos dispositivos móveis. Assim, os usuários dos dispositivos móveis que possuem idade abaixo dos 18 anos, por exemplo, não receberão mensagens com conteúdo ofensivo ou pornográfico. (4) Implementação de um filtro de conteúdo que faça uma tarifação diferenciada para o envio de mensagens promocionais ou propaganda. Nesse caso, a operadora, poderá tarifar dos provedores de conteúdo que enviam mensagens com propaganda de forma diferenciada.

REFERÊNCIAS

3GPP TS 22.039. **Interface Protocols for the Connection of Short Message Service Centers (SMSCs) to Short Message Entities (SMEs)**. 3rd Generation Partnership Project. TS 22.039 v.2.0.0 jun. 1999.

3GPP TS 22.002. **Technical Specification Group Services and Systems Aspects; Network architecture**. 3rd Generation Partnership Project. TS 22.002 v.7.1.0 mar. 2006.

3GPP TS 22.003. **Technical Specification Group Core Network; Numbering, addressing and identification**. 3rd Generation Partnership Project. TS 22.003 v.5.11.0 jun. 2006.

3GPP TS 22.040. **Technical Specification Group Core Network and Terminals; Technical realization of the Short Message Service (SMS)**. 3rd Generation Partnership Project. TS 22.040 v.6.8.1 out. 2006.

ANDRADE, Lélia Maria de. **Análise Comparativa de Técnicas de Inteligência Computacional para a Detecção de Spam**. 2006. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Minas Gerais, Belo Horizonte.

Android Developers. **What is Android?** Disponível em <http://developer.android.com/guide/basics/what-is-android.html>. Acesso em: 23 out. 2009.

ASSIS, João Marinho de Castro. **Detecção de E-mails Spam Utilizando Redes Neurais Artificiais**. 2006. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Itajubá, Itajuba.

ASVIAL, M. SIRAT, D. SUSATYO, B. **Design and analysis of anti spamming SMS to prevent criminal deception and billing froud: Case Telkom Flexi**. Management of Innovation and Technology, 2008. ICMIT 2008. 4th IEEE International Conference on, p. 928-933, sep. 2008.

BOAS, Roberta de Matos Vilas. **Faturamento com SMS deve crescer globalmente, mas no Brasil preço alto dificulta uso**. Disponível em <http://web.infomoney.com.br/templates/news/view.asp?codigo=1241084&path=/suasfinancas/estilo/tecnologia/> Acesso em: 19 out. 2008.

BALINSKI, Ricardo. **Filtragem de Informações no Ambiente do Direto**. 2002. Dissertação (Mestrado em Informática) - Universidade Federal do Rio Grande do Sul, Porto Alegre.

BERLO, D. The **Process of Communication: An Introduction to Theory and Practice**. San Francisco: Rinehart Press, 1960.

BRAGA, A. P.; CARVALHO, A. C.; LUDERMIR, T. B. **Redes Neurais Artificiais: Teoria e aplicações.** 1a. ed. Rio de Janeiro: LTC, 2000.

BURTON, B. **SpamProbe - Bayesian Spam Filtering Tweaks**. Disponível em <http://spamprobe.sourceforge.net/paper.html> Acesso em: 30 mar 2009.

CARRERAS, X. MARQUEZ, L. **Boosting trees for clause splitting**. In: ConLL '01: Proceedings of the 2001 workshop on Computational Natural Language Learning. Morristown, NJ, USA: Association for Computational Linguistics, p. 73-75, 2001.

Cellsigns. **Text Message Statistics**. Disponível em <http://www.cellsigns.com/industry.shtml> Acesso em: 07 nov. 2009.

CORMACK, G.; LYNAM, T. **Spam corpus creation for TREC**. In: Proceedings of the Second Conference on Email and Anti-Spam. Mountain View, CA, USA: CEAS, 2005. Disponível em: <http://www.ceas.cc/papers-2005/162.pdf>. Acesso em: 05 nov. 2009.

CRANOR, L. F. LAMACCHIA, B. A. **Spam!** In: Commun. ACM. New York, NY, USA: ACM, 1998. v. 41, p. 74–83. ISSN 0001-0782.

DEEPAK, P. SANDEEP, P. **Spam filtering using spam mail communities.** Applications and the Internet, 2005. Proceedings. The 2005 Symposium on, p. 377-383, feb. 2005.

DENG, Wei-Wei., PENG, Hong. **Research on a Naive Bayesian Based Short Message Filtering System**. Machine Learning and Cybernetics, 2006 International Conference on, p. 1233-1237, aug. 2006.

DROSSOS, D., GIAGLIS, G.M. LEKAKOS, G. **An Empirical Assessment of Factors that Influence the Effectiveness of SMS Advertising**. System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on, p. 58-58, jan. 2007.

DRUCKER, H. DONGHUI, W. VAPNIK, V. **Support vector machines for spam categorization**. Neural Networks, IEEE Transactions on Volume 10, Issue 5, p. 1048-1054, sep. 1999.

FOSTER, M. **Avoiding latch formation in regular expression recognizers**. Computers, IEEE Transactions on Volume 38, Issue 5, p. 754-756, may. 1989.

GARNER, Philip, MULLINS, Ian, EDWARDS, Reuben e COULTON, Paul. **Mobile Terminated SMS Billing – Exploits and Security Analysis**. Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on, p. 1319-1324, jun. 2008.

GRAHAM, P. **A plan for spam.** Disponível em <http://www.paulgraham.com/spam.html>, Acesso em: 30 mar 2009.

HE, P. SUN, Y. ZHENG, W. WEN, X, **Filtering Short Message Spam of Group Sending Using CAPTCHA**, Knowledge Discovery and Data Mining, 2008. WKDD 2008. International Workshop on, p. 558-561, jan. 2008.

HE, Peizhou, WEN, Xiangming e ZHENG Wei. **A Novel Method for Filtering Group Sending Short Message Spam**. Convergence and Hybrid Information Technology, 2008. ICHIT '08. International Conference on, p. 60-65, aug. 2008.

KANTARDZIC, M, **Data Minimg, Concepts, Models, Methods and Algorithms.** J. B. Speed Scientific School University of Louisvile, IEEE Computer Society, Sponser 2003.

KIM, S., HAN, K.; RIM, H., MYAENG, S. **Some Effective Techniques for Naive Bayes Text Classification**. Knowledge and Data Engineering, IEEE Transactions on, p. 1457-1466, nov. 2006.

KRISHNAMURTHY, N. **Using SMS to deliver location-based services.** Personal Wireless Communications, 2002 IEEE International Conference on, p. 177-181, dec. 2002.

Lógica. **Plataforma SMSC**. Disponível em <http://opensmpp.logica.com/CommonPart/Download/download2.html> Acesso em: 31 abr 2009.

LORENA, Ana Carolina. **Investigação de estratégias para a geração de máquinas de vetores de suporte multiclasses.** 2006. Tese (Doutorado em Ciências Matemáticas) - Instituto de Ciências Matemáticas e de Computação (ICMC) USP, São Carlos.

MAIER, J. FERENS, K. **Classification of english phrases and SMS text messages using Bayes and Support Vector Machine classifiers**. Electrical and Computer Engineering, 2009. CCECE '09. Canadian Conference on, p. 415-418, may. 2009.

MING L, YUNCHUN L. WEI L. **Spam Filtering by Stages**. Convergence Information Technology, 2007. International Conference on, p. 2209-2213, nov. 2007.

Mobilen50ar. Disponível em <http://www.mobilen50ar.se/eng/FaktabladENGFinal.pdf>. Acesso em: 23 out. 2009.

PRIETO, A.G. COSENZA, R. STADLER, R. **Policy-based congestion management for an SMS gateway**. Policies for Distributed Systems and Networks, 2004. POLICY 2004. Proceedings. Fifth IEEE International Workshop on, p. 215-218, jun 2004.

SAHAMI, M. DUMAIS, S. HECKERMAN, D. HORVITZ E. **A Bayesian Approach to Filtering Junk E-mail,** AAAI Workshop on Learning for Text Categorization, July 1998, Madison, Wisconsin. AAAI Technical Report WS-98-05.

SILVA, Alission Marques da. **Utilização de Redeis Neurais Artificiais para Classificação de Spam**. 2009. Dissertação (Mestrado em Modelagem Matemática e Computacional) - Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte.

SHAHREZA, M. **Verifyin Spam SMS y Arabic CAPTCHA**, Information and Communication Technologies, 2006. ICTTA '06. 2nd, p. 78-83, 2006.

SHAHREZA, S. M.H., **An Anti-SMS-Spam Using CAPTCHA**. Computing, Communication, Control, and Management, 2008. CCCM '08. ISECS International Colloquium on, p. 318-321, aug. 2008.

SUBRAMANYA, S.R., YI, B.K. **Mobile communications - an overview.** Potentials, IEEE Volume 24, Issue 5, p. 36-40, dec, 2005.

The Economist. **Television, meet text message**. Disponível em <http://www.economist.com/business/displayStory.cfm?story_id=1392699> Acesso em: 19 out. 2008.

TOORANI, Mohsen e SHIRAZI, Ali Asghar Beheshti. **SSMS - A Secure SMS Messaging Protocol for the M-Payment Systems**. Computers and Communications, 2008. ISCC 2008. IEEE Symposium on, p. 700-705, jul. 2008.

WAADT, A. BRUCK, G. JUNG, P. KOWALZIK, M. TRAPP, T. BEGALL, C. **QoS Monitoring for Professional Short-Message-Services in Mobile Networks**. Wireless Communication Systems, 2005. 2nd International Symposium on, p. 228-232, sep. 2005.

WU, Ningning, WU, Mingguang, CHEN, Siguo. **Real-time monitoring and filtering system for mobile SMS**. Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on, p. 294-299, apr. 2006.

ZURADA, J. M. **Introduction to Artificial Neural Systems.** Boston: PWS Publising Company, p. 721, 1992.

ZHANG, H. WEN, X; HE, P. ZHENG, W. **Dealing with Telephone Fraud Using CAPTCHA**, Computer and Information Science, 2009. ICIS 2009. Eighth IEEE/ACIS International Conference on, p. 1096-1099, jun 2009.

Anexo A

Lista de *Stopwords*

A.1 Advérbios de afirmação

sim	efetivamente	realmente
deveras	certamente	deverás

A.2 Advérbios de dúvida

talvez	porventura	possivelmente
quiçá	acaso	provavelmente
quica		

A.3 Advérbios de lugar

ali	aqui

A.4 Advérbios de modo

mal	bem	melhor
corretamente	debalde	pior
calmamente	depressa	fielmente

assim	devagar	levemente

A.5 Advérbios de negação

não	jamais	nunca
absolutamente	nao	

A.6 Advérbios de ordem

primeiramente	ultimamente	depois

A.7 Advérbios de tempo

hoje	depois	cedo
amanhã	ainda	então
sempre	anteontem	já
logo	antes	jamais
agora	breve	nunca
ontem	tarde	outrora
amanhã	já	então

A.8 Artigos definidos

a	o	as

os

A.9 Artigos indefinidos

uns	uma	uns
umas		

A.10 Artigos

o	no	nas
os	pelo	pelas
a	da	num
as	na	numa
um	pela	nuns
uns	aos	numas
uma	dos	dum
umas	nos	duma
ao	pelos	duns
do	das	dumas

A.11 Consoantes

b	c	d
f	g	h
j	k	l
m	n	p
q	r	s

t	v	x
y	z	w

A.12 Conjunções

que	porém	ou
mas	porem	e

A.13 Interjeições

ah	coragem	bravo
oh	eia	viva
oba	vamos	oxalá
opa	bis	ai
avante	bem	ui
chi	hem	psit
ih	alo	silencio
eu	o	alto
puxa	ola	basta
hum	psiu	uh

A.14 Numerais cardinais

um	dois	três
quatro	cinco	seis
sete	oito	nove
dez	onze	doze

treze	quatorze	catorze
quinze	dezesseis	dezessete
dezoito	dezenove	vinte
trinta	quarenta	cinquenta
sessenta	setenta	oitenta
noventa	cem	duzentos
trezentos	quatrocentos	quinhentos
seiscentos	setecentos	oitocentos
novecentos	mil	milhão
bilhao		

A.15 Numerais ordinais

primeiro	segundo	terceiro
quarto	quinto	sexto
sétimo	oitavo	nono
décimo	vigésimo	trigésimo
quadragésimo	qüinquagésimo	sexagésimo
septuagésimo	octogésimo	nonagésimo
centésimo	ducentésimo	trecentésimo
quadrigentesimo	qüingentésimo	seiscentesimo
sexcentesimo	septingentesimo	octingentésimo
nogentesimo	milesimo	milésimos
milionésimo	milionesimos	bilionésimo
bilionésimos	primeira	segunda
terceira	quarta	quinta
sexta	sétima	oitava
decima		

A.16 Preposições

a	de	sobre
ante	desde	sob
até	por	trás
após	para	trás
com	perante	ate
contra	sem	após
pára	no	duma
ao	na	duns
à	nos	dumas
aos	nas	num
às	pelo	numa
do	pela	nuns
da	pelos	numas
dos	pelas	das
dum	em	

A.17 Pronomes de tratamento

você	senhora	sua
voce	vossa	senhorita
senhor	senhoria	

A.18 Pronomes demonstrativos

este	esta	isto
esse	essa	isso

aquele	aquela	aquilo
estes	estas	tal
esses	essas	tais
aqueles	aquelas	

A.19 Pronomes indefinidos

alguém	todo	toda
ninguém	todos	todas
algo	outros	outra
nada	muitos	outras
tudo	poucos	muita
cada	certo	muitas
qualquer	certos	pouca
algum	vario	poucas
outro	tanto	certa
nenhum	tantos	certas
muito	quanto	varia
pouco	quantos	varias
mais	quaisquer	tanta
menos	alguma	tantas
vários	algumas	quanta
alguns	nenhuma	quantas
nenhuns	nenhumas	

A.20 Pronomes interrogativos

que	quem	qual
quanto	quantos	quais

A.21 Pronomes pessoais oblíquos átonos

me	lhe	os
te	se	as
o	nos	lhes
a	vos	

A.22 Pronomes pessoais oblíquos tônicos

mim	contigo	nos
ti	conosco	vos
si	convosco	comigo
consigo		

A.23 Pronomes pessoais reto

eu	ela	nós
tu	eles	vós
ele	elas	

A.24 Pronomes possessivos

meu	teus	nossos

teu	seus	vossos
seu	nosso	meus
vosso		

A.25 Pronomes relativos

qual	onde	aonde
quais	como	cuja
que	cujo	quem
quanto		

A.26 Vogais

a	e	i
o	u	