Jackson José de Souza

Identificação de alertas de segurança virtual veiculados em redes sociais

São Paulo - Brasil

2 de março de 2015

Jackson José de Souza

Identificação de alertas de segurança virtual veiculados em redes sociais

Universidade de São Paulo – USP
Instituto de Matemática e Estatística – IME-USP
Trabalho de formatura

Orientador: Daniel M. Batista
Coorientador: Elisabeti Kira

São Paulo - Brasil
2 de março de 2015

Agradecimentos

Como este trabalho simboliza o encerramento de um ciclo tão especial que é a graduação apesar de ainda não finalizado eu resolvi aproveitar para lembrar de todos que fizeram parte da minha educação desde o meu nascimento até o momento.

Aos meus pais por todo o apoio, incentivo e amor incondicional durante toda a vida.

À sociedade por financiar os meus estudos desde o ensino fundamental até o ensino superior e ao meu pai por arcar com os custos do cursinho pré-vestibular para corrigir a formação deficitária adquirida ao longo da minha formação e também por custear meus estudos de inglês na Nova Zelândia que além de me permitir melhorar a minha proficiência na língua inglesa também me proporcionou muitas experiências novas e inesquecíveis.

Ao professor Daniel eu dedico muita gratidão por toda a atenção, ajuda, paciência e conselhos dispendidos na orientação do trabalho e por disponibilizar toda a infraestrutura necessária para realizá-lo.

Ao Rodrigo e Luiz pelas colaborações, disponibilização dos tuítes e discussões sobre o trabalho.

À professora Elisabeti Kira por toda a atenção e pelos valiosos ensinamentos na orientação da escrita da pesquisa sobre a identificação de ASVs.

Ao Samu por ter me emprestado um de seus notebooks, o que me ajudou bastante a continuar o trabalho depois de eu ter perdido o meu notebook.

Ao professor JEF e à professora Nina pela disponibilidade e conselhos sobre abordagens que poderiam ser realizadas no trabalho.

À universidade por toda a infraestrutura disponibilizada para proporcionar um ambiente de discussões e convivência com a comunidade, principalmente com os estudantes.

Aos funcionários em funções não-docentes do IME pela dedicação e apoio ao ensino dentre os quais gostaria de ressaltar aqui alguns funcionários e setores que foram mais importantes na minha vida acadêmica como o Sérgio e o Nilson do setor de audiovisual, à Daniela assistente acadêmica, à Patty assistente administrativa, ao Vilemar assistente financeiro, à Rosana do serviço e contabilidade, à Ana Carla, Adenilza e Edna da secretaria do DCC, ao professor Clodoaldo atual diretor do IME-USP e ao professor Roberto Marcondes anterior chefe do DCC.

A todos os professores do DCC por toda a dedicação ao ensino, por serem tão

receptivos, abertos e pelo carinhos com os alunos além de tornarem o relacionamento entre professor e aluno bastante respeitoso e saudável.

Também dedico agradecimento especial a alguns professores do DCC:

Ao Carlinhos por todas as dicas, conselhos, carinho e apoio durante todos esses anos. Pessoas pela qual tenho muita admiração respeito e carinho.

Ao Coelho por todo o aprendizado adquirido ao longo do tempo em que fui bolsista do projeto Apoio ao BCC e por insistir que não se deve usar sujeito indeterminado apesar de eu discordar disso.

Ao Gubi por toda a atenção, conversas e conselhos ao longo do curso, como também por fiar a minha participação no FISL por meio de verba do IME e pelas minhas matrículas em algumas disciplinas.

À Nina, por quem eu tenho bastante apreço, por diversas conversas no IME.

Ao Coelho, Felipe, Goroba, Haruki, Henrique, Jé, Jeff, Manzo, Miojo, Omar, Paulo, Renato, Samu, Su, Wall, Wil, colegas de turma que eu tive o prazer de conhecer, pelos quais eu desenvolvi estima e carinho durante a longa jornada que foi a graduação e dentre os quais fiz várias amizades que creio que serão para toda a vida.

Aos outros colegas de USP, principalmente do BCC, com os quais também convivi ao longo destes anos e dentre os quais gostaria destacar o Solfer e Will que considero como bons amigos.

Entre estes principais amigos que fiz no IME e que fazem parte da minha turma eu gostaria de destacar:

Felipe, umas das pessoas mais presentes durante a convivência no IME e que sempre estava disposto a me ajudar com as matérias.

Paulo, que foi quem me recebeu quando eu fui calouro e sempre me deu várias dicas, conselhos, me ajudou em diversos momentos e por quem eu tenho bastante carinho e sempre ficará na minha lembrança.

Samu, uma pessoa muito legal, compreensiva, de muita sabedoria e com quem as discussões sobre qualquer coisa rendem bastante e resultam em uma troca de conhecimentos muito prazerosa.

Wall, amigo leal, sempre convidando para vários jogos de futebol e um bom ombro amigo.

Su, amiga devotada, disposta a me animar para o que quer que seja, quem mais se aproxima pra mim do ideal de amigo pra todas as horas e que está seu lado a qualquer momento, sempre brincalhona me dando vários sustos ao fazer cócegas em mim a qualquer hora e em qualquer lugar. Fonte de admiração pelo ser que ela é e fonte de gratidão infinita

por toda a amizade oferecida desde que nos conhecemos.

Finalmente agradeço a todas as pessoas que de alguma forma fizeram ou fazem parte da minha vida e foram especiais para mim, mas que não foram mencionadas aqui.

Resumo

Palavras-chaves: segurança computacional, redes sociais, Twitter, aprendizado de máquina.

Lista de ilustrações

Lista de tabelas

Sumário

Parte I

Parte objetiva

1 Introdução

O alto grau de domínio tecnológico na fabricação de computadores e a diminuição do seu custo possibilitou a popularização do seu uso pela sociedade. Essa popularização tem provocado, nas últimas décadas, transformações na forma como as pessoas consomem, se comunicam, se relacionam com empresas, se entretêm etc. Como várias destas atividades necessitam da Internet para serem realizadas, a disseminação do seu uso acompanhou naturalmente o aumento do uso dos computadores.

Dessa forma, a Internet tornou-se um ambiente ubíquo e passou a ser acessado não só por computadores de mesa ou portáteis, mas também por dispositivos como vídeo-games, celulares, relógios, etc. Por sua vez, as redes sociais online[1] atraíram milhões de usuários desde a sua introdução devido ao massivo uso da Internet. As interações possibilitadas por uma rede social entre usuários são bastante variadas e entre as mais comuns estão a comunicação e o compartilhamento de informações sobre assuntos de interesse em comum. Usuários podem ser pessoas, organizações, instituições, etc. As redes sociais online se tornaram tão populares que 3 delas estão entre as 10 páginas mais acessadas na internet[2]. As redes sociais online serão chamadas neste trabalho de redes sociais para a leitura deste termo não cansar o leitor.

O fato de 10 bilhões[3] de dispositivos estarem conectados à Internet mostra a abrangência do seu uso e a sua importância como meio de comunicação. Como em todo meio de comunicação, a segurança das informações transmitidas é fundamental. Afinal, apenas em um meio de comunicação seguro é possível viabilizar e assegurar a disponibilidade, a integridade, a confidencialidade e a autenticidade das informações. Porém, não há um controle rígido sobre o tráfego de dados que passa pela Internet, o que facilita que informações sejam extraídas de computadores sem o conhecimento dos seus donos. Isso demonstra que a liberdade de comunicação proporcionada pela Internet aliada às falhas de segurança presentes nos softwares que a utilizam revelam um risco à segurança de pessoas, empresas, instituições e etc. Este risco é grave porque as brechas na segurança virtual[4] podem provocar danos morais e materiais no mundo real. Existem alguns softwares desenvolvidos para proteger os computadores contra falhas de segurança como anti-vírus, *firewalls*, *anti-adwares* entre outros. Alguns deles, inclusive, utilizam heurísticas para detectar ameaças que não tenham sido identificadas e catalogadas. Apesar da existência de tais softwares de segurança, ataques e invasões continuam causando grandes prejuízos. Estudos apontam

---

1 Redes sociais online são sites que permitem interação entre os usuários cadastrados na página.

2 Informação obtida no dia 13/09/2013 em <http://www.alexa.com/topsites>

3 <http://gigaom.com/2011/10/13/internet-of-things-will-have-24-billion-devices-by-2020/>

4 Que é feito ou simulado através de meios electrónicos. “virtual”, in Dicionário Priberam da Língua Portuguesa [em linha], 2008-2013, http://www.priberam.pt/DLPO/virtual [consultado em 30-01-2015].

que as perdas com crimes virtuais alcançam a casa das centenas de bilhões de dólares em prejuízos sofridos por usuários e empresas a cada ano (STRATEGIC; STUDIES, 2013).

Portanto, existem muitas falhas de segurança que os mecanismos de combate e prevenção não conseguem solucionar, pelo menos antes delas serem descobertas. Por isso, é necessário identificar e corrigir tais brechas o quanto antes. Entre as várias formas de se descobrir falhas de segurança em um software, por exemplo, existe a identificação de alertas de segurança virtual (ASV) veiculados pela rede em sites de segurança, fóruns etc. Em (SANTOS et al., 2012) foi mostrado que é possível utilizar redes sociais para detecção de ASVs, como o Twitter[5]. Contudo, os ASVs não são agrupados em uma categoria específica pelo Twitter e não é fácil encontrá-los usando as ferramentas de busca disponibilizadas pelas redes sociais. Assim, percebeu-se que esse é um problema interessante de se abordar e é dele que este trabalho trata. Relacionado com o que é apresentado neste trabalho há um serviço chamado OpenCalais[6] que extrai informação semântica de textos não estruturados[7] e retorna um metadado com informações sobre entidades presentes no texto. Por exemplo, pessoas, lugares, fatos entre outros. As entidades são extraídas utilizando o contexto presente no texto, mas é necessário que as entidades tenham um determinado grau de associação em um dado contexto. Contudo, o OpenCalais não gera conhecimento das informações que ele extrai a partir das entidades que ele identifica. Em outras palavras, o serviço é capaz de extrair conhecimento das entidades, mas ele não consegue gerar metaconhecimento nem agrupar as informações por classes como é feito neste trabalho.

1.1 Motivação do trabalho

Entre o segundo trimestre de 2011 e o início de 2012 houve um aumento nas atividades hacktivistas autopromovidas no Brasil. Em maio de 2011 ocorreu um grande volume de ataques feitos pelo Anonymous[8] a empresas privadas e agências governamentais brasileiras e em janeiro de 2012 nove grandes bancos ou agências governamentais foram atacados. O motivo do hacktivismo[9] no Brasil sofrer um crescimento tão intenso deve-se ao Twitter. A disseminação do uso do Twitter no Brasil, país cuja comunidade de usuários é uma das mais assíduas[10], tornou o Twitter uma boa plataforma de recrutamento para o Hacktivismo. Por sua vez o grande uso do Twitter por brasileiros foi impulsionado pela democratização do acesso aos computadores e à Internet, que tem sido realizada no Brasil.

[5] <https://www.twitter.com>
[6] <http://www.opencalais.com/>
[7] Dados não estruturados são dados que não possuem uma estrutura clara ou semanticamente evidente como um modelo de dados e possuem irregularidades ou ambiguidades. Um exemplo de dados estruturados são bancos de dados relacionais.
[8] Definição em: http://pt.wikipedia.org/wiki/Anonymous
[9] https://pt.wikipedia.org/wiki/Hacktivismo acessado em 27/07/2014
[10] http://www.socialmediatoday.com/content/brazil-social-media-marketers-gold-mine

Hoje, até mesmo uma parte da população que mora nas áreas mais pobres do país têm e-mail, acesso a redes sociais, entre outros serviços disponíveis na Internet.

Quando o hacktivismo começou, ele atraiu a atenção de muitos brasileiros. Dentre estes, vários acreditam que alguns alvos do hacktivismo mereciam sofrer os ataques e na mente dos brasileiros tais ataques não são crime. Dessa forma, o hacktivismo acabou por tocar em ponto nevrálgico, a insatisfação da população contra governos e empresas. Ao tornar ataques de negação de serviço (*DDoS*[11]) acessíveis às massas sem realizar grandes esforços, um alvo pode parar de funcionar com uma revolta popular promovida na rede. Este cenário é apenas um entre vários outros que criam o interesse de empresas e governos em monitorar redes sociais. (IMPERVA, 2012)

A necessidade de monitorar redes sociais coloca em foco outra atividade cujo objetivo é o combate e a prevenção dos ataques: a identificação de ASVs. O problema da identificação de ASVs ainda não foi explorado o suficiente e várias pessoas dentro da própria comunidade da área da computação não sabem definir adequadamente o que caracteriza um ASV. Isto foi concluído após a análise dos dados da pesquisa de identificação de ASVs do Twitter no Apêndice A, realizada com o objetivo de avaliar a percepção que as pessoas possuem sobre ASVs. Alguns participantes da pesquisa manifestaram ter sentido dificuldade em fazer a classificação dos tuítes. Para exemplificar a dificuldade seguem abaixo as opiniões enviadas por 2 participantes da pesquisa:
"Muitas possibilidades para definir o que é segurança virtual. Em um universo de expressões infinitas. Virus Definitions Update Download -> Definitions Update Download is a Virus. São muito próximas as expressões, mas diferentes. A questão é o que muda nas duas?"
"É meio dificil separar o que é "alerta" mesmo (urgente, corra para se proteger/atualizar algo específico) do que é notícia relacionada com segurança (algo mais genérico, como a história dos plugins de browser), mas todos são relevantes no aspecto de seguranca digital. Claro que tem que separar notícias que realmente falam de segurança daquelas que não tem nenhum conteúdo relevante nesse aspecto (ex: a da venda do exploit)."
A dificuldade de identificar alertas de segurança também se revelou nas classificações. Alguns tuítes foram classificados como alerta de segurança virtual por aproximadamente 50% dos participantes enquanto os outros cerca de 50% os classificaram como não sendo alertas de segurança virtual. Além da tarefa de identificar um ASV em publicações de redes sociais não ser simples, é inviável a um ser humano olhar cada possível alerta e classificá-lo manualmente, dado que 9.100 tuítes por segundo são postados no Twitter[12].

[11] Um ataque de negação de serviço (também conhecido como DoS Attack, um acrônimo em inglês para Denial of Service), é uma tentativa de tornar recursos de um sistema indisponíveis para seus utilizadores. Alvos típicos são servidores web, e o ataque tenta tornar as páginas hospedadas indisponíveis na WWW. Não se trata de uma invasão do sistema, e sim da sua invalidação por sobrecarga. Mais detalhes em: <https://pt.wikipedia.org/wiki/Ataque_de_nega%C3%A7%C3%A3o_de_servi%C3%A7o>

[12] <http://www.statisticbrain.com/twitter-statistics/>

Por isso, uma solução para o problema é desenvolver e utilizar um sistema para identificar os alertas de segurança automaticamente.

1.2 Objetivo

Neste trabalho são realizadas algumas análises das mensagens de segurança no Twitter escritas na língua inglesa para detectar alertas de segurança virtual. As análises são feitas utilizando recuperação de informação, processamento de linguagem natural e são utilizadas técnicas de aprendizagem computacional para identificar as mensagens que contêm alertas de segurança virtual com o auxílio da Weka, um software de mineração de dados. Este estudo serve de apoio às teses de doutorado do Luiz A. F. Santos e do Rodrigo Campiolo que estão relacionadas com a detecção antecipada de anomalias em redes de computadores.

O estudo também está diretamente relacionado ao projeto "GT-EWS: Mecanismos para um Sistema de Alerta Antecipado" [13] financiado pela Rede Nacional de Ensino e Pesquisa (RNP) no edital de 2014/2015 de grupos de trabalho. As conclusões acerca dos mecanismos de classificação de tuítes implementados neste trabalho serão usadas para guiar a equipe do projeto no desenvolvimento de ferramentas que buscarão alertas antecipados sobre segurança virtual em redes sociais.

1.3 Estrutura do trabalho

O trabalho está dividido da seguinte forma:

- Conceitos básicos
  Introdução teórica a conceitos importantes para a compreensão do desenvolvimento e experimentos realizados no trabalho

- Desenvolvimento
  Descrição do processo de coleta e rotulação dos tuítes, apresentação das classes do problema e também é explicado como os dados foram filtrados e pré-processados para serem utilizados na classificação.

- Experimentos, resultados e discussões
  Apresentação da Weka, descrição e discussão dos experimentos realizados, apresentação e interpretação dos resultados.

[13] <https://www.pop-ba.rnp.br/GTEWS/>. Último acesso em 12/02/2015.

- Conclusão

  Aponta contribuições do trabalho e oferece ideias para a realização de melhorias que podem aumentam a acurácia dos classificadores.

2 Conceitos básicos

Este capítulo apresenta conceitos e fundamentos que sustentam o desenvolvimento do trabalho. A seção 2.1 apresenta uma definição de rede social, explica como elas funcionam e apresenta o Twitter, suas características principais e como os usuários podem compartilhar conteúdo nele em suas várias formas. A seção 2.2 define os conceitos de segurança adotados neste trabalho e a seção 2.3 define os conceitos envolvendo especificamente segurança virtual para que se entenda como foram escolhidas as classes dos tuítes. A seção 2.5 apresenta conceitos envolvendo leitura de documentos, extração de termos e a posterior remoção, redução, contagem de termos e transformação dos dados para a extração das características dos tuítes. A seção 2.6 apresenta conceitos necessários para a compreensão da vinculação entre atributos, ou características, e classes. A seção 2.7 introduz conceitos relacionados à aprendizagem de máquina para permitir o entendimento do processo de classificação dos tuítes e os resultados do processo.

2.1 Redes sociais e Twitter

Esta seção define o que são redes sociais, quais são os seus propósitos e como funcionam, colocando em destaque o Twitter, fonte dos dados utilizados no trabalho, explicando o que é um tuíte, qual o seu conteúdo, e como ele se propaga dentro do Twitter. As fontes utilizadas na escrita desta seção são (BOYD; ELLISON, 2007; TWITTER..., 2013).

Redes sociais são serviços hospedados na Internet que permitem a indivíduos construir um perfil, publicar conteúdo no seu perfil e em grupos, visualizar sua lista de conexões e participar de outras listas que tenham sido criadas por outros. As conexões podem ser os amigos em uma rede social e as listas podem ser conjuntos de integrantes de um grupo, de evento etc. Vale mencionar que a classificação de um relacionamento em uma rede social como sendo o de amizade não significa que os usuários realmente sejam amigos no sentido denotativo da palavra.

A visibilidade do conteúdo gerado ou compartilhado por usuários depende das restrições impostas pela rede social e pelos usuários. Por exemplo, o perfil do usuário pode ser total ou parcialmente público e as informações do perfil e das publicações dele podem ser abertas a todos os usuários da internet, ou apenas a usuários cadastrados na rede social ou também podem ser visíveis ou fechadas a apenas determinadas conexões do divulgador do conteúdo.

As conexões podem ser estabelecidas de forma unidirecional e bidirecional. Ou

seja, uma conexão bidirecional depende da aprovação de ambos os usuários envolvidos nela a respeito do status do relacionamento enquanto que a unidirecional depende apenas da vontade de um usuário. Em algumas redes sociais, para duas pessoas serem amigas é necessário o consentimento de ambas, mas um usuário pode liberar o acesso do conteúdo que ele publica a outros usuários sem que estes façam o mesmo.

As comunicações entre conexões podem ser feitas de várias formas. Entre elas, existe a troca de mensagens visível a todos os usuários da rede, aos usuários conectados a pelo menos um dos participantes da troca de mensagens e apenas entre os participantes da troca de mensagens. O tipo de conteúdo publicado varia desde textos em língua natural a fotos, áudios, vídeos entre outros tipos de conteúdo. Finalmente, como as redes sociais possuem o intuito de serem o mais acessíveis possível, elas possibilitam o seu uso por meio de computadores de mesa, notebooks, *smartphones* e até aparelhos celulares comuns.

O que torna as redes sociais únicas é o fato de que elas não apenas permitem que indivíduos conheçam estranhos, mas também possibilitam aos usuários se comunicarem e tornarem visíveis seus grupos de conexões. Ao fazer alguma publicação dentro da rede social o conteúdo gerado pelo usuário pode ser difundido entre todos os seus grupos de conexões e ser compartilhado por estas conexões a outros grupos e usuários com os quais o publicante original não possui conexão. Isto possibilita que um conteúdo tenha alcance ilimitado dentro da rede social e permite a criação de conexões entre indivíduos, que não seriam possíveis de outra forma. Apesar de, em geral, não existir o objetivo de se criar tais conexões, a publicação de conteúdo na rede social faz com que as comunicações estabelecidas com outros usuários da rede social tenham como consequência a criação de novas conexões na plataforma devido à existência de interesse em comum entre indivíduos que não se conhecem pessoalmente.

Porém, vale ressaltar que os usuários, em várias das redes sociais, não estão necessariamente buscando fazer troca de conhecimento com pessoas que possuem interesses em comum nem buscando criar novas conexões. Na verdade, elas podem apenas ter a intenção de se comunicar na plataforma com contatos que elas já possuem no mundo físico.

O Twitter é uma rede social que funciona também como microblog permitindo aos usuários lerem e enviarem tuítes, que são mensagens de texto com até 140 caracteres. Os tuítes podem ser enviados por meio de aplicativos para *smartphones*, página na Internet ou, em alguns países, por SMS. Os tuítes dos usuários, por padrão, são visíveis a qualquer um que tenha acesso a Internet, mas seu acesso também pode ser limitado apenas aos usuários conectados ao usuário que envia tais mensagens, os chamados seguidores. Um seguidor no Twitter é uma conexão que possui permissão para ler as mensagens de um dado usuário e permite a esse usuário enviar 'mensagens diretas'[1] ao seguidor. Se um

[1] 'Mensagens diretas' são tuítes que apenas o remetente e o destinatário podem ver. Usuários podem ser pessoas, empresas, instituições etc.

usuário quiser ele pode deixar de seguir alguém ou bloquear algum seguidor.

Os tuítes utilizam alguns metadados[2] que permitem realizar algumas operações na plataforma. As *hashtags* - palavras ou frases precedidas de uma cerquilha '#' - são usadas para agrupar tuítes por assunto. Da mesma forma, é possível mencionar um usuário em um tuíte usando um @ sucedido do nome de um usuário (sem espaços), o que permite que o usuário mencionado no tuíte possa lê-lo. Uma resposta é um caso particular de menção em que o tuíte começa com o @ seguido do nome do usuário ao qual se está respondendo. Além disso, também é possível compartilhar um tuíte contanto que o seu dono não tenha limitado seu acesso apenas a seus seguidores. Os tuítes compartilhados possuem o acrônimo *RT* (*retweet*) seguido pela menção do usuário que originalmente escreveu o tuíte. Outra característica comum nos tuítes é o uso de URLs curtas[3], em substituição às URLs originais, devido ao limite do número de caracteres de um tuíte. Caso o usuário esteja escrevendo um tuíte com uma URL não encurtada (mais de 20 caracteres) o Twitter utiliza seu próprio encurtador.
A Figura 1 mostra a captura de tela de um tuíte que reúne todos os tipos de metadados supracitados:

Figura 1 – Exemplo de tuíte com os tipos de metadados mais comuns no Twitter

Além do tuíte clássico de 140 caracteres também é possível publicar tuítes expandidos. Estes tuítes podem conter fotos, vídeos e cartões. Estes conteúdos multimídia

[2] Metadados são dados sobre dados. De outra forma, podemos dizer que um metadado possui informações relativas a um dado. Exemplo: um tuíte pode ser armazenado como uma estrutura de dados que contém uma mensagem (também chamada de tuíte) em texto, vídeo ou imagem e contém campos com informações referentes a esta mensagem como local onde foi escrita, língua, data de criação, etc. Estas informações sobre o tuíte escrito constituem o metadado dele.

[3] URL curta (*short URL*) é um endereço reduzido de uma página que costuma ser utilizado para referenciar o endereço original de um site em textos cujo limite de caracteres para escrita é reduzido como o Twitter. Quando se deseja obter uma URL curta pode-se usar um serviço de encurtamento de URLs como o <http://tinyurl.com/>.

podem ser adicionados usando a própria plataforma do Twitter, para fazer o *upload* de fotos, ou aplicativos como o <https://vine.co/> para inserir vídeos no tuíte. O conteúdo multimídia é disponibilizado via um link, respeitando o limite de caracteres de um tuíte. Um cartão de tuíte ou *tweet card* é um conteúdo multimídia expandido utilizado para exibir fotos, vídeos, propaganda, resumo de notícias etc. O cartão é gerado a partir da inserção de alguns metadados no tuíte que permitem visualizar o conteúdo. Note que este é um meio adicional utilizado para publicas fotos e vídeos no Twitter, mas não é o único.

2.2 Segurança

Neste trabalho entende-se por segurança a ação ou o resultado da promoção da proteção de um bem, seja ele material ou imaterial, e compreende também o combate e prevenção de ameaças a tal bem. Um bem material pode ser uma pessoa, uma casa. Um bem imaterial seria o conhecimento ou a cultura de um povo, por exemplo.

A preocupação das sociedades, instituições e países em proteger vários tipos de bens inspirou a divisão de tais bens em diversas categorias. Dentre as mais importantes para este trabalho estão:

- **Segurança do interior**
  Esta categoria, também chamada em inglês de *Homeland security*, se refere aos esforços nacionais para prevenir ataques terroristas, reduzir a vulnerabilidade de um país ao terrorismo e minimizar os danos consequentes de ataques e desastres naturais que ocorrerem. Esta categoria foi adaptada à preocupação atual dos EUA com o risco de ataques terroristas em seu país.

- **Segurança pública**
  "A Segurança Pública é uma atividade pertinente aos órgãos estatais e à comunidade como um todo, realizada com o fito de proteger a cidadania, prevenindo e controlando manifestações da criminalidade e da violência, efetivas ou potenciais, garantindo o exercício pleno da cidadania nos limites da lei." (MINISTéRIO..., 2013).

- **Segurança nacional**
  Trata-se do estado mensurável da capacidade de uma nação superar as múltiplas ameaças ao aparente bem estar da sua população e sua sobrevivência como um Estado-nação, a qualquer momento. Isto se faz pelo balanceamento da política de estado através da governança, que pode ser guiada pela computação, empiricamente ou de outra forma, e é extensível à segurança global por variáveis externas ao governo (PALERI, 2008).

- **Segurança física**
  Segurança física compreende as medidas adotadas para negar acesso não autorizado a instalações, equipamentos e recursos, e proteger o pessoal e propriedade contra perdas e danos provocados por espionagem, roubo, ataques terroristas e desastres naturais. Traduzido e adaptado de (HEADQUARTERS, 2001).

- **Segurança pessoal**
  Segurança pessoal é um conjunto de ações preventivas, adotadas com vistas a assegurar a integridade física, mental ou moral de si ou de outro (WIKIPEDIA..., b).

Além das categorias citadas acima também existe o que se decidiu chamar de segurança virtual e é a esta categoria de segurança que é dedicada a seção 2.3.

2.3 Segurança virtual

Esta seção aborda alguns conceitos e possui definições envolvendo segurança virtual que serão utilizados na classificação dos tuítes. Alguns dos conceitos e definições desta seção são baseados em (FUTURE, 2011; SECURITY, 2010; WILSHUSEN, 2013; CERT.PT; CSIRT, 2012; SHIRLEY, 2007; WILSHUSEN, 2011; GLOSSARY..., 2009).

Segurança virtual consiste na prevenção de dano, uso não autorizado, exploração e também envolve a restauração de dados, sistemas de comunicação e de informação para garantir confidencialidade, integridade, autenticidade e acessibilidade de dados e programas de computador.

Um incidente de segurança virtual (ISV) pode ser considerado como um evento adverso, confirmado ou sob suspeita, que tem por consequência o acesso, extração, manipulação ou corrompimento da integridade, confidencialidade, segurança ou acessibilidade de dados ou programas de computador, sejam públicos ou privados, sem autorização legal. Um evento pode ser causado intencional ou não intencionalmente, ter um alvo restrito ou amplo, e pode fazer uso de variadas técnicas. Ele pode surgir a partir de diferentes fontes, incluindo um país fazendo espionagem ou guerra de informações contra outros países, criminosos, *crackers*, programadores de vírus, terroristas entre outros.

ISVs não intencionais podem ser causados por erro ou omissão humana e falhas de equipamentos, como por exemplo, a operação de um sistema por funcionários displicentes ou sem treinamento, atualizações defeituosas de programa de computador, realização de manutenções entre outros. ISVs não intencionais podem corromper dados ou provocar interrupções ou mau funcionamento de sistemas. ISVs intencionais são provocados por um ente inteligente, como um *cracker* ou organização criminosa, e incluem ataques com alvo restrito ou amplo. Um ataque com alvo restrito ocorre quando um grupo de pessoas

ou um único indivíduo realiza um ataque contra um sistema de infraestrutura crítica, organização ou pessoa. Um ataque de alvo amplo ocorre quando o alvo definido para a realização do ataque é um número grande de pessoas, empresas, organizações e etc.

No contexto de segurança virtual, um ataque consiste na tentativa de destruir, expor, alterar ou incapacitar algum software, sistema e/ou dados contidos neles, ou produzir qualquer outra falha de segurança em dispositivos eletrônicos (GLOSSARY. . . , 2009 apud STANDARDIZATION, 2006).

Há algumas formas de estruturar a classificação dos eventos e incidentes como em (WILSHUSEN, 2013; CERT.PT; CSIRT, 2012). Neste trabalho vamos adotar a classificação de ISV utilizada em (CERT.PT; CSIRT, 2012). Existem várias classes e tipos de incidentes que agrupam tipos de eventos. Para conhecer mais os tipos de eventos veja (CERT.PT; CSIRT, 2012).

Define-se um alerta de segurança virtual (ASV) como um aviso, geralmente de caráter urgente, sobre a ameaça, ocorrência, uma notícia de solução para, uso de ferramenta para, ou a explicação de como gerar um ISV.

Um exemplo de ASV pode ser visto no tuíte[4] da Figura 2:

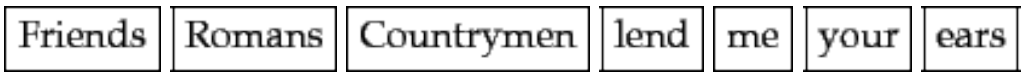
Figura 2 – Exemplo de Alerta de segurança virtual

2.4 Processamento de linguagem natural

Para classificar as mensagens compartilhadas no Twitter é necessário realizar processamento de linguagem natural. Esta seção possui alguns conceitos para entender como esse processamento é realizado.

[4] Todos os tuítes deste trabalho seguem as regras de publicação de tuítes em trabalhos segundo o Twitter. Ver <https://twitter.com/logo> seção: Offline (static uses and publications)

Tokenização é o ato de decompor um documento em peças chamadas *tokens*. As peças são ocorrências específicas de cadeias de caracteres e são separadas por caracteres chamados delimitadores. Em geral, no processo de tokenização alguns caracteres são descartados como os delimitadores de tokens. Eis um exemplo:
Documento: Friends, Romans, Countrymen, lend me your ears;
Delimitadores (entre aspas simples): ‘,’ ‘;’ ‘ ’

Tokens: 

Note que o caractere espaço (‘ ’) é também um delimitador de token.

Normalização de tokens é a tarefa de converter em uma forma canônica tokens formados por conjuntos distintos de caracteres mas que possuem um mesmo significado. Assim, temos o token canônico que dá nome à sua classe de equivalência e quaisquer tokens que sejam convertíveis ao canônico pertencem à mesma classe de equivalência e são representantes dela.
Exemplos:
naive => Naive, Naïve, naïve, NAÏVE, NAIVE
usa => U.S.A., USA

Neste trabalho *tipo* é uma classe de todos os tokens que possuem exatamente a mesma cadeia de caracteres e que faz parte do dicionário de classificação. Em outras palavras, um token é uma cópia de um tipo. Um documento pode conter várias cópias (tokens) do seu representante, mas um tipo é único.

Termo é um tipo eventualmente normalizado presente no dicionário de um sistema de recuperação de informação.

Stemming ou stemização é o processo de reduzir palavras flexionadas ou derivadas à uma forma básica comum, o radical. O radical neste caso não precisa ser igual ao seu homônimo linguístico.
Exemplo: real, reais, realizar, realizável, realista => rea

Vale ressaltar que os vários conceitos de tratamento mencionados acima podem ser aplicados de várias formas e isso depende do programa de computador utilizado. Ou seja, há mais de uma forma de se fazer *stemming*. A forma utilizada depende dos objetivos a serem alcançados com o conteúdo do texto que está sendo analisado.

2.5 Recuperação de informação

Esta seção possui alguns conceitos de quantificação de informações em um documento e de similaridade entre documentos.

Stop words, ou palavras de parada, são, em geral, palavras muito frequentes em textos que não trazem informação relevante na análise de um documento e por isso são removidas do vocabulário de termos em um sistema de recuperação de informação (RI) . A lista de *stop words* pode variar conforme as necessidades e particularidades do sistema de RI.

O modelo sacola de palavras é uma representação simplificada de documentos utilizada em processamento de linguagem natural e em problemas de RI. No modelo, cada documento é representado por uma coleção de termos (ou palavras) ignorando a ordem de ocorrência deles no documento. Os termos de uma coleção de documentos formam o vocabulário utilizado em problemas que usam o modelo sacola de palavras.

O modelo de língua unigrama (*unigram language model*) considera que a probabilidade da ocorrência de cada termo é independente da ocorrência de quaisquer outros termos e a ordem com que eles ocorrem também não importa.
Então, temos: $P(t_1 \cap t_2 \cap \cdots \cap t_n) = P(t_1) \times P(t_2) \times \cdots \times P(t_n)$, onde $P(x)$ denota a probabilidade do termo x ocorrer e $t_1, t_2, \cdots, t_n$ representam termos. A Tabela 1 apresenta um exemplo de cálculo de probabilidade usando o modelo de língua unigrama.

Termo	Probabilidade
sacola	0.3
palavras	0.2
simples	0.15

Tabela 1 – Exemplo de um modelo sacola de palavras com a lista de termos do vocabulário e as respectivas probabilidades deles.

Dado o vocabulário da tabela 1 e o seguinte documento: “O modelo sacola de palavras é bem simples, pois ele lida apenas com palavras.” temos que a probabilidade da ocorrência do documento acima dada a sacola de palavras é dada por P(sacola, palavras, simples, palavras) $= 0.3 \times 0.2 \times 0.15 \times 0.2$

Frequência de um termo, ou *term frequency*, é o número de vezes que um termo t ocorre em um documento d e é denotada por $tf_{t,d}$.

Frequência de um documento, ou *document frequency*, é o número de documentos em uma coleção que contém um termo t e é denotada por df_t

Considerando uma coleção com N documentos define-se a frequência inversa de um documento (idf), ou *inverse document frequency*,

$$idf_t = log\frac{N}{df_t}. \tag{2.1}$$

O método de ponderação $tf\text{-}idf$ atribui a um termo t um peso no documento d dado por

$$tf\text{-}idf_{t,d} = tf_{t,d} \times idf_t. \tag{2.2}$$

O modelo de espaço vetorial, ou *vector space model*, é uma representação de um conjunto de documentos como vetores em um espaço vetorial e cada termo de um dicionário é representado como um eixo. Os valores de cada coordenada da representação do documento no espaço vetorial podem ser obtidos utilizando medidas que quantifiquem a presença de cada termo do espaço no documento como $tf\text{-}idf$ ou $tf_{t,d}$.

Similaridade de cossenos é uma medida que quantifica a similaridade entre dois documentos d_1 e d_2 utilizando as suas representações $\vec{V}(d_1)$ e $\vec{V}(d_2)$ no espaço vetorial:

$$sim(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{\mid \vec{V}(d_1) \mid\mid \vec{V}(d_2) \mid}. \tag{2.3}$$

2.6 Teoria da Informação

Entropia é uma medida da quantidade de informação obtida na ocorrência de um evento em um conjunto de eventos possíveis, no caso deste trabalho, os termos. O objetivo da entropia é caracterizar a incerteza ou pureza da fonte de informação (WIKIPEDIA..., c). Pode-se dizer que quanto maior a entropia mais uniforme é a distribuição da ocorrência dos eventos e menos informação ela traz. Por outro lado, se a entropia é baixa a ocorrência dos eventos é mais previsível e mais informativa sobre os próprios eventos. A entropia pode ser calculada por:

$$H(V) = -\sum_{t \in |V|} P(t) \log P(t), \quad t : termo, \tag{2.4}$$

onde $\mathbb{V}$ é o vocabulário de termos.

De acordo com (MITCHELL, 1997) *information gain*, ou redução da incerteza, é a redução esperada na entropia causada pelo particionamento de exemplos de acordo com um atributo. Primeiro calcula-se a entropia, por exemplo, de uma classe c. A utilidade dessa medida é aferir a quantidade de informação provida pelo atributo sobre um evento. Em outras palavras, se $IG(c|V)$ tiver um valor alto em relação a $H(c|V)$ significa que o atributo está vinculado a uma determinada classe c. Abaixo a fórmula para *information gain* segundo (MOORE, 1994).

$$H(c|V) = \sum_{k=1}^{|V|} P(V = t_k) H(c|V = t_k), \tag{2.5}$$

onde |V| = número de termos no vocabulário.

$$IG(c|V) = H(c) - H(c|V), \tag{2.6}$$

onde $H(c|V)$ é a entropia de uma classe c dado o vocabulário $\mathbb{V}$ e $IG(c|V)$ é a redução da incerteza da classe c para um vocabulário $\mathbb{V}$.

2.7 Aprendizagem Computacional

Aprendizagem de Máquina é uma subárea de Inteligência Artificial cujo objetivo é construir sistemas que são treinados a executar uma dada tarefa aperfeiçoando o seu desempenho conforme ganham experiência em realizar tal tarefa. Aprendizagem de Máquina utiliza técnicas em que se busca encontrar padrões nos dados relacionados à tarefa de interesse e são definidas regras ou maneiras de identificar e extrair tais dados para que o sistema possa executar a tarefa de forma satisfatória. O aprendizado é bem sucedido se ele se aperfeiçoa conforme aumenta a exposição aos dados relacionados ao problema que deve ser resolvido ou atinge uma alta taxa de acerto na realização da tarefa.

Também podemos definir aprendizagem de máquina como:

> *Um programa de computador aprende a partir da experiência E com respeito a uma classe de tarefas T e medida de desempenho P, se a sua performance em T, segundo a medição P, melhora com a experiência E.*
> Tradução livre de (MITCHELL, 1997)

Por exemplo, um sistema que utiliza Aprendizagem de Máquina pode ser treinado para distinguir mensagens enviadas por e-mail e separá-las em mensagens spam e não spam. Conforme aumenta o número de exemplos de mensagens spam e não-spam o sistema consegue separar de forma cada vez mais próxima do ideal as mensagens spam das não-spam. Após os resultados do treinamento serem considerados satisfatórios o sistema está pronto para ser aplicado na realização da tarefa que ele aprendeu. Ou seja, no exemplo dado ele já pode ser usado para separar, ou classificar, as mensagens de e-mail recebidas por um usuário que irão para a caixa de entrada e as que irão para a pasta de spam.

Para o problema de aprendizagem ficar melhor definido considere as suas características:

- Tarefa T: separar mensagens spam de mensagens não-spam
- Medição P: porcentagens separadas nas categorias corretas
- Experiência E: exemplos de mensagens spam e não-spam

O ato de identificar a categoria a qual um dado pertence é chamado de classificação, que também podemos definir da seguinte forma:
Dado um conjunto de classes busca-se determinar a qual classe um objeto pertence. Um problema de classificação possui 2 ou mais classes e embora na maioria dos problemas um objeto pertença a apenas uma classe, também é possível atribuir mais de uma classe a cada objeto. Um algoritmo de aprendizagem de máquina que faz a classificação de objetos é chamado de classificador (WIKIPEDIA..., d).

Há varias formas de aprendizagem, mas vamos nos limitar ao escopo deste trabalho que é a aprendizagem supervisionada. Na aprendizagem supervisionada o sistema recebe um conjunto de exemplos de dados rotulados[5] para realizar o treinamento. Cada exemplo possui um dado de entrada associado a uma categoria. Um algoritmo de aprendizagem supervisionada analisa os dados de treinamento e produz uma função inferida que pode ser usada para mapear novos exemplos. O cenário ótimo é aquele em que o algoritmo classificador determina a classe para instâncias, ou dados, não conhecidos. Para isto o algoritmo precisa desenvolver uma boa generalização para dados não conhecidos (WIKIPEDIA. . . , e).

O processo de aprendizagem pode ser visto com um número variável de etapas dependendo da forma como ele for realizado. No contexto deste trabalho o processo de classificação será dividido nas etapas a seguir, representadas também no fluxograma da Figura 3.

1. Aquisição dos dados
2. Rotulação dos dados
3. Pré-processamento
4. Extração e seleção das características
5. Seleção do algoritmo de aprendizagem
6. Treinamento
7. Validação
8. Teste

1 Aquisição dos dados é a atividade de coletar os dados na sua forma bruta, ou seja, como são observados, e remover instâncias de dados que não se aplicam ao problema.

2 Rotulação é o ato de categorizar (atribuir uma classe às) instâncias de dados para construir os conjuntos de treinamento, validação e teste. Em muitos casos o ato de rotular os dados exige um alto grau de conhecimento sobre o domínio e por isso precisa ser realizado por um especialista para evitar que os dados sejam rotulados incorretamente e o classificador fique pouco confiável.

3 Pré-processamento é a fase em que os dados são tratados para remover informações que não são de interesse, remover ruídos, isolar padrões de interesse considerando o contexto do problema, filtrar informações e aplicar correções ou alterações nos dados para

[5] Rotulação é o ato de atribuir uma classe a cada objeto utilizado na construção do classificador, o que envolve o conjunto de treinamento, validação e teste do classificador.

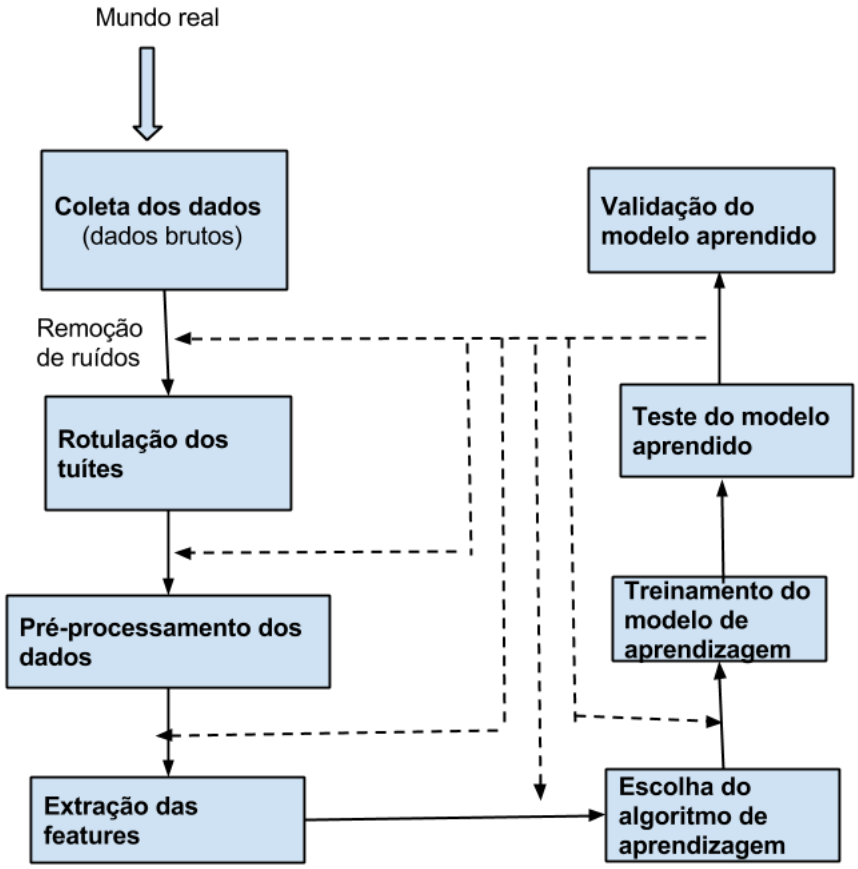


Figura 3 – Fluxograma de aprendizagem

melhorar o desempenho do classificador. Para pré-processar os dados é necessário conhecer a sua representação e significado. Ou seja, saber como são representados, o significado das características presentes neles (por exemplo, metadados) e, se possível, conhecer os tipos de ruído que eles possuem para que o pré-processamento melhore a aprendizagem da tarefa de interesse.

4 Na extração e seleção das características é realizada a representação do documento em termos de características e são selecionadas as características, ou atributos, a serem utilizados para o algoritmo escolhido aprender a classificar os dados.

6 Treinamento do classificador é a fase em que o modelo adotado irá aprender os critérios de decisão para realizar a classificação e auxiliar o ajuste dos parâmetros do algoritmo e a seleção das características.

7 A fase de validação serve para estimar o erro do classificador e verificar o desempenho dos ajustes realizados no algoritmo e seleção das características utilizando o conjunto de dados de validação.

8 A fase de teste simula a avaliação do classificador no mundo real. O desempenho do classificador em relação ao conjunto de teste serve para estimar como o classificador

irá se comportar em relação a dados que não foram observados pelo classificador e para aperfeiçoar o ajuste dos atributos do algoritmo e a extração e seleção de características. O conjunto de dados de teste precisa ser disjunto em relação ao conjunto de teste e de treinamento.

O processo de aprendizagem é cíclico como mostra o fluxograma da Figura 3, pois os resultados de algumas etapas podem exigir uma melhora na seleção das características ou nos parâmetros dos algoritmos classificadores.

Para realizar a fase de treinamento e de validação são utilizadas algumas técnicas de amostragem e serão descritas aqui apenas as que são utilizadas no trabalho. Considerando que a amostra utilizada no treinamento e validação é previamente separada da amostra de teste, dentre as técnicas de amostragem algumas das mais comuns são:

- *Holdout*
- Validação cruzada (amostragem sem reposição)

O *Holdout* é feito dividindo a amostra (A) em conjunto de treinamento (S) e conjunto de validação (T) e a proporção mais comum adotada é 2:1. A amostra S é utilizada para treinar o classificador e a amostra T para estimar o erro do mesmo. Se o conjunto A for pequeno, isso significa que S e T são menores ainda e as estimações realizadas são pouco confiáveis.

A técnica de validação cruzada busca compensar este problema fazendo uso da reamostragem. São consideradas k partições (S_i, T_i) e para cada partição (S_i, T_i) é realizado o treinamento com S_i e estima-se o erro sobre T_i o que resulta em k erros $ErroT_i$. O erro de validação cruzada é dado pela média dos erros $ErroT_i$:

$$Erro = \frac{1}{k} \sum_{i=1}^{k} ErroT_i \tag{2.7}$$

Na validação cruzado do tipo *k-fold* a amostra é dividida em k partes de tamanhos (aproximadamente) iguais. São repetidas k rodadas de treinamento, deixando alternadamente uma das partes para validação em cada rodada.

Para avaliarmos o desempenho de um classificador são usadas algumas métricas dentre as quais as principais estão presentes abaixo.

A matriz de confusão (WIKIPEDIA..., a), também conhecida como tabela de contingência, é uma tabela específica que permite visualizar o desempenho de um algoritmo de classificação. A Tabela 2 apresenta um exemplo. O conteúdo de cada célula deve ser a quantidade de dados que se encaixam naquela classificação (correta ou incorreta).

Tabela 2 – Matriz de confusão

		Classe inferida	
		Sim	Não
Classe do tuíte	Sim	verdadeiro positivo - vp	falso positivo - fp
	Não	falso negativo - fn	verdadeiro negativo - vn

Precisão corresponde à fração de documentos retornados que pertencem a uma classe em relação ao número total de documentos retornados,

$$\frac{vp}{vp + fp}. \tag{2.8}$$

Cobertura, ou (*Recall*), corresponde à fração de documentos retornados que pertencem a uma classe em relação ao número total de documentos existentes na classe,

$$\frac{vp}{vp + fn}. \tag{2.9}$$

Acurácia é a fração de documentos classificados corretamente,

$$\frac{vp + vn}{vp + fp + fn + vn}. \tag{2.10}$$

F-measure é uma média harmônica utilizada para ponderar cobertura e precisão a problemas de RI. O α é fixado de acordo com a medida que se julga mais ou menos importante na recuperação de documentos,

$$\frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}}, \tag{2.11}$$

onde P é a precisão e R é a cobertura

2.8 Classificadores e modelo de língua

Nesta seção são apresentados os algoritmos de classificação utilizados nos experimentos do trabalho e os modelos de dados que os algoritmos utilizam para fazer a classificação. Ao final da seção são comparados os consumos de tempo dos algoritmos para cada método de cruzamento das classes. A principal bibliografia adotada nesta seção é (MANNING; RAGHAVAN; SCHüTZE, 2009).

2.8.1 *Naive Bayes*

O classificador *naive Bayes*, ou Bayes ingênuo, é um método de aprendizagem probabilístico baseado no modelo de língua unigrama multinomial. Considerando o modelo sacola de palavras, a ordem de ocorrência dos termos é ignorada, e o modelo unigrama estabelece que as probabilidades de ocorrência dos termos são independentes entre si.

Apesar da ordem dos termos ser ignorada no modelo unigrama, ou seja, as probabilidades de quaisquer ordenações deles serem iguais, é preciso considerar que as ordenações dos termos existem e como consequência a distribuição dos termos é multinomial.

Portanto, temos o algoritmo *naive Bayes* multinomial que calcula a probabilidade de um documento d pertencer a uma classe c como segue:

$$P(c|d) \propto P(c) \prod_{1 \leqslant k \leqslant n_d} P(t_k|c), \tag{2.12}$$

onde n_d é o número de termos em um documento.

Para obtermos as probabilidades $\hat{P}(c)$ e $\hat{P}(t_k|c)$, utilizamos o estimador de máxima verossimilhança (EMV) que corresponde à frequência relativa (o valor mais provável de cada parâmetro dado considerando o conjunto de treinamento). Assim, temos:

$$\hat{P}(c) = \frac{N_c}{N}, \tag{2.13}$$

onde N_c é o número de documentos na classe c e N é o número total de documentos no conjunto de treinamento.

$$\hat{P}(t_k|c) = \frac{T_{ct}}{\sum\limits_{t' \in V} T_{ct'}}, \tag{2.14}$$

onde $T_{ct'}$ é o número de ocorrências do termo t nos documentos de treinamento da classe c, incluindo múltiplas ocorrências de um termo em um mesmo documento.

Em classificação o objetivo é encontrar a melhor classe para o documento. Ou seja, a classe mais provável, ou *maximum a posteriori* (MAP) c_{map} :

$$c_{map} = \underset{c \in \mathbb{C}}{arg\,max}\ \hat{P}(c|d) = \underset{c \in \mathbb{C}}{arg\,max}\ \hat{P}(c) \prod_{1 \leqslant k \leqslant n_d} P(t_k|c), \tag{2.15}$$

onde $\mathbb{C}$ é o conjunto das classes do problema.

Na fórmula acima foi descartado o fator correspondente às combinações possíveis para cada termo por simplicidade, e diminuição da complexidade computacional, pois ele se mantém igual no cálculo da probabilidade de cada classe possível. Além disso, para evitar problemas de precisão de ponto flutuante, o algoritmo naive Bayes substitui produto de probabilidades pela soma de logaritmos de probabilidades.

$$c_{map} = \underset{c \in \mathbb{C}}{arg\,max}\ [\hat{P}(c) + \sum_{1 \leqslant k \leqslant n_d} P(t_k|c)], \tag{2.16}$$

Como os dados disponíveis para treinamento não são grandes o suficiente para representar adequadamente a frequência de eventos raros, ou seja, o espaço de dados

disponível é esparso, alguns termos terão probabilidade 0 de ocorrer. Para evitar a ocorrência dos zeros é utilizada a suavização de Laplace para contar a frequência de cada termo presente nos documentos,

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V}(T_{ct'} + 1)} = \frac{T_{ct} + 1}{((\sum_{t' \in V} T_{ct'}) + B')}, \tag{2.17}$$

onde $B = |V|$ é o número de termos no vocabulário.

O algoritmo de Bayes é chamado de ingênuo por adotar a hipótese de independência entre termos que é bastante equivocada, porque existe bastante dependência entre a ocorrência das palavras numa frase porque elas precisam obedecer a uma estrutura gramatical e possuem contexto. Apesar disso as decisões tomadas por ele são surpreendentemente boas e por isso em geral ele alcança uma boa acurácia nas classificações. No espaço logarítmico ele é linear, o que significa que ele é bom para classificar documentos cujas classes são linearmente separáveis e pelo fato dele ser linear, ele é pouco influenciado por *noise features*[6].

2.8.2 *Support Vector Machines (SVM)*

SVM é um tipo de classificador de margem largo: ele é um método de aprendizagem computacional baseado em um espaço vetorial onde o objetivo é encontrar uma fronteira de decisão entre duas classes que maximiza a distância de qualquer ponto no espaço do conjunto de treinamento, possivelmente desconsiderando alguns pontos considerados ruídos ou *outliers*[7]

A representação de documentos adotada pelo SVM é o modelo de espaço vetorial já discutido anteriormente. Cada documento é um ponto no espaço vetorial e o valor de cada coordenada é obtido por meio de algum método de ponderação de termos como o tf-idf. Assim, o espaço vetorial possui dimensão $\mathbb{R}^{|V|}$. O modelo de espaço vetorial adota a hipótese de contiguidade segundo a qual documentos da mesma classe formam uma região contígua e regiões de diferentes classes não se sobrepõem. Isso significa que os documentos devem ser representados de forma adequada no espaço vetorial para não quebrar a hipótese de contiguidade. Por isso para representar os documentos no espaço vetorial é necessário normalizar o tamanho dos documentos e utilizar usar bons métodos de ponderação como o tf-idf.

A princípio será explicado o SVM para problemas de duas classes e depois ele será estendido para problemas com mais de duas classes usando o método *one-against-one*. O SVM busca encontrar uma superfície de decisão maximalmente longe de quaisquer

[6] *Noise features* são ocorrências de instâncias raras de dados que aumentam o erro do classificador porque elas introduzem um viés nas predições e provocam *overfitting*.

[7] *Outliers* são pontos que se situam distantes de quaisquer outros documentos e portanto não se encaixam bem em nenhuma classe.

pontos e a distância dos pontos mais próximos da superfície determinam a margem do classificador que separa a superfície de decisão das regiões de cada classe. O método de construção da função de decisão é especificada por um conjunto pequeno de pontos que definem a posição do separador. Isso faz com que tais pontos sejam chamados de *support vectors*, pois um ponto pode ser visto como um vetor entre a origem e o próprio ponto. Uma margem grande faz boas decisões de classificação porque erros sutis ou pequenas variações nos documentos não provocam uma classificação incorreta dos documentos.

Formalizando a ideia definimos um hiperplano de decisão que intercepta uma constante b e um vetor $\vec{\omega}$ normal (chamado de vetor ponderado ou *weight vector*) perpendicular ao hiperplano. A constante b determina todos os pontos $\vec{x}$ do hiperplano separador tais que $\vec{w}^T\vec{x} = -b$. O conjunto de dados de treinamento é dado por $\mathbb{D} = \{(\vec{x_i}, \vec{y_i})\}$. Cada dado é representado pelo ponto $\vec{x_i}$ e um rótulo $\vec{y_i} \in \{1, -1\}$ representando uma classe. Assim, temos que o classificador linear é dado por,

$$f(\vec{x}) = sign(\vec{w}^T\vec{x} + b), \tag{2.18}$$

onde o valor da função (1 ou -1) prediz a classe do documento.

A decisão é confiável se o ponto ficar longe o suficiente da fronteira de decisão. Para medir a distância de um ponto ao hiperplano define-se margem funcional como o i-ésimo exemplo $\vec{x}$ em relação ao hiperplano $\langle\vec{\omega}, b\rangle$ com o valor $y_i(\vec{w}^T\vec{x_i}+b)$. Para limitar o tamanho da margem funcional, adota-se a distância euclidiana do ponto $\vec{x}$ mais próximo da fronteira de decisão e chamamos essa distância de r. Temos que a distância mais curta entre um ponto e o hiperplano é perpendicular ao plano e, por isso, paralela a $\vec{\omega}$. Um vetor unitário nesta direção é $\vec{\omega}/|\vec{\omega}|$. Considerando o vetor $r\vec{\omega}/|\vec{\omega}|$ e chamando de $\vec{x}'$ o ponto do hiperplano mais próximo de $\vec{x}$, temos:

$$\vec{x}' = \vec{x} - yr\frac{\vec{\omega}}{|\vec{\omega}|}, \tag{2.19}$$

sendo que multiplicar por y apenas muda o sinal para os casos de $\vec{x}$ estar em quaisquer dos dois lados da superfície de decisão. Inclusive, $\vec{x}'$ situa-se sobre a fronteira de decisão e por isso satisfaz $\vec{w}^T\vec{x}' + b = 0$

$$\vec{w}^T(\vec{x} - yr\frac{\vec{\omega}}{|\vec{\omega}|}) + b = 0 \tag{2.20}$$

Isolando r tem-se:

$$r = y\frac{\vec{w}^T\vec{x} + b}{|\vec{\omega}|} \tag{2.21}$$

Considera-se como margem geométrica a largura máxima da faixa que pode ser desenhada separando os *support vectors* das duas classes. Ou seja, a margem geométrica é duas vezes o valor mínimo sobre os pontos para r dada a equação (2.21). A margem geométrica é invariante ao produto por escalares, pois ela é implicitamente normalizada pelo tamanho de $\vec{\omega}$.

Define-se para todos os dados que,

$$y_i(\vec{w}^T\vec{x_i} + b) \geq 1, \tag{2.22}$$

tal que existem alguns pontos, os *support vectors*, que são os pontos que satisfazem a igualdade. Dessa forma, para qualquer dado a distância do hiperplano é dada por $r_i = y_i(\vec{w}^T\vec{x_i} + b)/|\vec{\omega}|$ e a margem geométrica é $\rho = 2/|\vec{\omega}|$. O objetivo é maximizar a margem geométrica. Logo deseja-se encontrar $\vec{\omega}$ e b de modo a:

- maximizar $\rho = 2/|\vec{\omega}|$
- Para todo $(\vec{x_i}, y_i) \in \mathbb{D}$, $y_i(\vec{w}^T\vec{x_i} + b) \geq 1$

Maximizar $2/|\vec{\omega}|$ é o mesmo que minimizar $|\vec{\omega}|/2$, o que permite ver a formulação do SVM como o seguinte problema de minimização:

Encontrar $\vec{\omega}$ e b de modo a:

- minimizar $\frac{1}{2}\vec{w}^T\vec{w}$
- Para todo $\{(\vec{x_i}, y_i)\}$, $y_i(\vec{w}^T\vec{x_i} + b) \geq 1$

Em geral os dados não são linearmente separáveis quando o número de dimensões do espaço não é muito grande em problemas de classificação de texto, o que provoca a realização de predições incorretas. Porém, o mais importante é que a maior parte dos dados sejam separáveis, então alguns erros na predição são admissíveis, mas isto exige que seja feita uma alteração no modelo para permitir estes erros. Em programação linear quando se necessita remover uma desigualdade em uma das restrições adiciona-se *variáveis residuais* que representam o custo de transformar desigualdades em igualdades e a função de minimização paga por isso um custo proporcional às variáveis residuais. No problema de minimização do SVM adicionam-se variáveis residuais ζ_i a cada dado para representar o custo de $\vec{x_i}$ violar a margem geométrica. Portanto, a formulação do problema de minimização do SVM fica:

Encontrar $\vec{\omega}$, b e $\zeta_i \leq 0$ de modo a:

- minimizar $\frac{1}{2}\vec{w}^T\vec{w} + C\sum_i \zeta_i$

- Para todo $\{(\vec{x_i}, y_i)\}$, $y_i(\vec{w}^T\vec{x_i} + b) \geq 1 - \zeta_i$

Assim, a margem pode ser menor que 1 para um ponto $\vec{x_i}$ fazendo $\zeta_i > 0$ e a função de minimização "paga uma pena" de $C\zeta_i$ por isso. O parâmetro C chamado de termo de regularização é utilizado para controlar *overfitting* e funciona da seguinte forma: conforme C se torna grande é desvantajoso desobedecer a margem porque o custo fica alto e quando C fica pequeno alguns dados podem violar a margem sem grandes prejuízos, ou diminuição, para a função de minimização e ainda é possível manter uma margem larga para separar uma grande parte dos dados. A forma dual do problema, que busca encontrar o hiperplano que minimiza o número de *support vectors* foi omitida, pois o objetivo da apresentação é entender como ele funciona sem fazer uma exposição muito extensa dele.

O SVM foi pensado para duas classes e ainda existe muita discussão sobre a melhor forma de estendê-lo para mais que duas classes, que é o caso do problema deste trabalho. As formas mais comuns de estendê-lo é utilizando o método *todos-contra-um* e o método *um-contra-um*. Neste trabalho será utilizado o método um-contra-um (HSU; LIN, 2002). O SVM um-contra-um é uma forma de utilizar o SVM de duas classes de forma que cada classe é confrontada contra todas as outras, uma por vez, e a classe determinada para um documento é aquela que for escolhida, ou votada, mais vezes dentre todas os problemas de duas classes. Caso mais de uma classe receba o maior número de votos é escolhida a classe de menor índice. Logo, o SVM do método um-contra-um resolve $|\mathbb{C}|(|\mathbb{C}| - 1)$ problemas SVM de duas classes.

Classificador	Modo	método	Complexidade
Naive Bayes	treinamento		$\Theta(\|\mathbb{D}\|L_{médio} + \|\mathbb{C}\|\|\mathbb{V}\|)$
Naive Bayes	teste		$\Theta(L_a + \|\mathbb{C}\|M_a) = \Theta(\|\mathbb{C}\|M_a)$
SVM	treinamento	convencional	$O(\|\mathbb{C}\|\|\mathbb{D}\|^3 M_{médio})$
SVM	teste	convencional	$O(\|\mathbb{C}\|M_a)$
SVM	treinamento	um-contra-um	$O(\|\mathbb{C}\|(\|\mathbb{C}\| - 1)\|\mathbb{D}\|^3 M_{médio}))$
SVM	teste	um-contra-um	$O(\|\mathbb{C}\|(\|\mathbb{C}\| - 1)M_a)$

Tabela 3 – Complexidade dos algoritmos de classificação para realizar o treinamento e teste dos modelos de aprendizagem computacional obtida de (MANNING; RAGHAVAN; SCHüTZE, 2009).

A tabela 4 exibe uma comparação do tempo gasto nas etapas de treinamento, validação (mesma complexidade da fase de teste) e teste dos classificadores SVM e *Naive Bayes*. Treinamento é o tempo gasto pelos métodos de aprendizagem para aprender um classificador sobre $\mathbb{D}$ (conjunto de documentos). A complexidade da fase de teste é o que se gasta para classificar apenas um documento. $L_{médio}$ é o número médio de tokens por documento, $M_{médio}$ é o tamanho médio do vocabulário de um documento (número de termos pertencentes ao vocabulário que estão presentes no documento). L_a e M_a são as quantidades de tokens e termos (ou tipos), respectivamente, presentes em um documento.

Nota-se que o tempo gasto pelo *Naive Bayes* para treinar um conjunto de documentos aumenta linearmente em função do número de documentos. Como o vocabulário $\mathbb{V}$ é constante ao longo do treinamento o tamanho dele apenas influencia o tempo gasto no treinamento se $|\mathbb{V}|$ for absurdamente alto. No caso do SVM o tempo da fase de treinamento aumenta cubicamente em função do número de documentos no conjunto de treinamento e quadraticamente em função do número de classes. Portanto, o aumento do conjunto de treinamento é bastante limitado por estar condicionado a um rápido aumento no tempo gasto para treinar o classificador. Na fase de validação tanto *Naive Bayes* e o SVM de duas classes possuem o mesmo tempo para classificar um único documento, enquanto que o custo de classificação do SVM multiclasse pode ser alto por variar quadraticamente em função do número de classes do problema.

3 Desenvolvimento

Este capítulo detalha o processo de coleta rotulação e pré-processamento dos dados que serão utilizados nos experimentos descritos no capítulo seguinte.

3.1 Coleta dos dados

A coleta dos dados no Twitter foi feita por Rodrigo Campiolo e Luiz Artur usando uma API do Twitter em Java. Foi utilizado um filtro para recuperar apenas os tuítes que respeitassem a seguinte busca: *security AND (virus OR worm OR attack OR intrusion OR invasion OR ddos OR hacker OR cracker OR exploit OR malware)*.

Os tuítes foram coletados em três fases compondo assim três bases de dados: A 1ª base de dados possui 260440 tuítes coletados entre 28/04/2012 a 10/02/2013. A 2ª base de dados possui 40307 tuítes coletados entre 14/06/2013 a 31/07/2013. A 3ª base de dados possui 37422 tuítes coletados em outubro de 2014. Todas as bases de dados foram armazenadas em arquivos com formato json.

3.2 Classes do problema

A definição das classes do problema baseou-se em uma análise realizada no conteúdo dos tuítes coletados. Este problema foi definido como sendo do tipo *one-of*, ou seja, cada tuíte rotulado ou classificado pertence necessariamente e exclusivamente a uma das classes abaixo.

Alerta de segurança virtual: Notícia sobre a ocorrência, ameaça ou descoberta recente para correção de um ISV.

Notícia de segurança virtual: Mensagens sobre artigos, notícias, *reviews*, *reports*, informações ou discussões sobre assuntos e pessoas relacionadas à computação e segurança virtual e que não possuem o objetivo explícito de promover um determinado produto, serviço ou publicação.

Notícia de segurança geral: Nesta classe enquadram-se as mensagens em que o assunto principal é o que concerne a segurança de países, organizações, instituições, cidades, pessoas e política, mas que não se enquadrem na categoria de segurança virtual. Também pertencem a essa categoria ações ou medidas dos poderes executivo, legislativo e judiciário de países, exceto aquelas que produzam ou corrijam ISVs.

Spam: Mensagens cujo teor não foi identificado, ou que se tratam de comentários ou afirmações que não estão diretamente ligadas a uma notícia ou reportagem identificada,

ou que tem por objetivo a promoção explícita de produtos e serviços e publicações que não tem a função de publicar conteúdo periódico sobre segurança virtual ou que não está relacionado ao conteúdo das outras classes.

3.3 Rotulação

Para rotular um tuíte é necessário ler o seu conteúdo e identificar a qual classe ele pertence. Para simular a aleatoriedade na escolha de um tuíte também é necessário sortear um número de 1 ao número de tuítes na base de dados correspondente ao tuíte a ser lido (n). Isso é importante, pois há tuítes repetidos nas bases de dados coletadas devido ao fato de as pessoas compartilharem a mesma mensagem e isso faria com que alguns tuítes se repetissem muitas vezes nos conjuntos de dados rotulados.

Para realizar a tarefa foi necessário escrever um *script*, pois os tuítes estavam armazenados como uma estrutura de dados em json com muitas chaves. Além disso, para rotular vários tuítes foi necessário abrir os links na mensagem, o que o *script* fazia de forma automatizada. Por fim, sortear um número e copiar um tuíte do arquivo original e colar no arquivo correspondente à classe a qual ele foi atribuído é muito cansativo e sujeito a erros. Uma simulação mostrou que a semi-automatização da rotulação para 100 tuítes leva cerca de 60 minutos enquanto que a rotulação automatizada dura aproximadamente 30 minutos.

O *script label_tweets.rb* escrito em ruby armazena todos os tuítes de uma base de dados e abre os arquivos correspondentes às classes do problema. É sorteado um número de 1 a n, o tuíte é decodificado (pois ele contém entidades codificadas em html), a mensagem do tuíte é exibida na tela e são abertos os links presentes no tuíte. Então deve-se selecionar a classe a qual pertence o tuíte ou ignorá-lo e escolher outro para classificar. Se o tuíte é rotulado, ele é copiado no arquivo correspondente à classe escolhida, é apagado da base de dados para evitar que uma cópia da mesma instância de dado possa ser atribuída à mesma classe no futuro e é escolhido um novo tuíte aleatoriamente. Também é exibido na tela o número de tuítes já rotulados na execução atual do *script*.

Há outras duas operações importantes realizadas na rotulação dos tuítes. A primeira, foi a filtragem realizada pelo *script* que permitia rotular apenas tuítes com 35 ou mais caracteres, já descontando os metadados (exceto as *hashtags*) e links presentes neles. Essa filtragem foi pensada considerando-se que um tuíte precisa ter informação suficiente para ser classificado e foi baseada no critério utilizado em (SANTOS et al., 2012) para fazer o *clustering* dos tuítes. A filtragem dos tuítes segundo o critério do tamanho foi possível por meio da utilização de um módulo chamado *filter_tweet_data.rb* escrito em Ruby para fazer a filtragem dos tuítes e remoção de alguns metadados, símbolos e caracteres que não agregam informação ao tuíte na classificação.

A segunda foi a eliminação de tuítes em outras línguas, que não a inglesa, o que caracteriza ruído. Por algum motivo não esclarecido, foram coletados tuítes em outras línguas, que não o inglês, apesar da língua inglesa ter sido especificada na coleta dos dados realizada pelo Rodrigo e pelo Artur. A princípio tentou-se escrever um classificador de línguas em Perl para remover os tuítes que não estavam escritos em inglês antes da rotulação dos dados, mas o *script* não funciona para línguas com caracteres não latinos. Então, a eliminação destes tuítes foi realizada de forma manual.

A 1ª e a 2ª bases de dados foram agrupadas em um único arquivo json do qual são extraídos os tuítes para as fases de treinamento e validação e metade da fase de teste. A outra metade da fase de teste foi obtida da 3ª base de dados.

Inicialmente foram rotulados exatamente 12010 tuítes. O conjunto de treinamento e validação totalizaram 9332 tuítes, rotulados entre outubro de 2013 e junho de 2014. A 1ª e a 2ª metade do conjunto de teste totalizaram 1518 e 1160 tuítes respectivamente, que foram rotulados em janeiro de 2015.

3.4 Pré-processamento dos dados

Para pré-processar os tuítes é preciso conhecer o domínio. Porém, antes de disponibilizar os dados pré-processados para extrair e selecionar as características é necessário remover os tuítes repetidos nos conjuntos rotulados.

Assim, primeiro é necessário pré-processar os tuítes, remover as instâncias repetidas para depois preparar os dados para serem utilizados pelos classificadores. Remover as instâncias repetidas é importante para evitar super ajuste (*overfitting*[1]).

Para fazer o pré-processamento foi escrito um *script* Ruby chamado *filter_tweet_data.rb* disponibilizado como um módulo para ser utilizado por outros *scripts*. Porém, antes de começar o pré-processamento dos tuítes é utilizado um analisador sintático (*parser*) da estrutura de dados do tuíte em json para um objeto em Ruby e então o texto é decodificado. Após isso o pré-processamento começa removendo todos os metadados presentes em um tuíte, exceto as *hashtags* que podem ser utilizadas como características no trabalho. Depois são removidos os links, as aspas simples e duplas são literalizadas para evitar colisão de delimitadores de strings, é aplicado *feature engineering*[2] por meio do uso de algumas expressões regulares para isolar alguns padrões encontrados nos dados, então são removidos dos dados alguns caracteres inúteis na extração e seleção de características e são removidos espaços excedentes.

[1] *Overfitting* ocorre quando uma função é bastante ajustada para maximizar a acurácia na predição dos dados observados. *Overfitting* costuma ser causado por uma amostra viesada ou não representativa dos dados e pela ocorrência de instâncias raras de dados.

[2] Processo de transformar dado bruto em características que representem melhor o contexto do problema

Para remover os tuítes repetidos foi escrito um *script* Ruby (*remove_repeated_tweets.rb*). O *script* primeiro realiza a análise sintática dos tuítes, para depois decodificá-los e então pré-processá-los. Após serem pré-processados os tuítes são tokenizados e os seus termos são utilizados para construir o modelo de espaço vetorial para representar os tuítes (documentos). É construída a representação dos documentos no espaço vetorial em função do método de ponderação *tf-idf* e é calculado o valor da similaridade de cossenos ($sim(d_i, d_j)$) entre todos os documentos de um conjunto de documentos. Dado um documento d_i, $i\,e\,j \in [1, N]$, N = número de documentos no conjunto, são removidos os documentos d_j tais que $sim(d_i, d_j) > 0.8$. O limite inferior para 0.8 para $sim(d_i, d_j)$ foi obtido por inspeção a olho nu dos tuítes comparados e se mostrou acurado na detecção de tuítes iguais.

A Tabela 3 apresenta o número de tuítes antes e depois da execução do *script* de remoção de tuítes para cada um dos conjuntos de tuítes rotulados.

Execução do *script*	Antes	Depois	Após ajuste no tamanho dos conjuntos
Treinamento e validação	9332	7333	7422
Teste parte 1	1518	1138	1012
Teste parte 2	1160	1012	1012

Tabela 4 – Tabela com o tamanho de cada um dos conjuntos de tuítes em cada etapa da remoção de tuítes repetidos entre todos os conjuntos de dados.

A tabela 3 descreve os conjuntos de tuítes antes, depois da execução do *script* sobre cada um dos conjuntos de tuítes e após ajustes no tamanho deles. Após terem sido removidos os tuítes repetidos de cada classe o *script* de remoção de tuítes repetidos foi executado para o conjunto de treinamento e validação contra o conjunto de teste parte 1, pois ambos os tuítes foram obtidos da mesma base de dados. Depois disso foi necessário mover alguns tuítes da parte 1 para o conjunto de treinamento e validação para fazer uma comparação estatisticamente mais confiável do desempenho do classificador entre os conjunto de teste parte 1 e parte 2 pois a parte 1 possuía mais tuítes que a parte 2 do conjunto de teste.

Após a remoção dos tuítes repetidos os tuítes restantes podem ser pré-processados e ficarem prontos para serem usados na classificação. Para isso, foi escrito o *script* Ruby (*arff_document_generator.rb*) que realiza a análise sintática dos tuítes, depois os decodifica, os pré-processa e então os tuítes são escritos em um arquivo no formato usado pelo software Weka, associados aos respectivos nomes de usuário que os postaram. Isso é feito separadamente para cada conjunto de dados descritos na tabela dos conjuntos de tuítes.

4 Experimentos, resultados e discussões

Na primeira seção deste capítulo é feita uma breve apresentação da ferramenta utilizada para realizar o processo de classificação dos tuítes e são descritos os passos para efetuar a classificação na ferramenta. Na seção seguinte os experimentos são categorizados, são descritos todos os passos da sua realização e os respectivos parâmetros utilizados em cada passo. A última seção deste capítulo apresenta os resultados acompanhados de discussões sobre seus aspectos positivos e negativos além das suas possíveis causas.

Para a realização dos experimentos foram utilizados o manual oficial da Weka anexo ao software e alguns artigos de (HIDALGO, 2013) além do uso de (PARANAGAMA, 2013) para semiautomatizar a execução dos experimentos.

4.1 Uso da Weka para classificação

A Weka é um conjunto de algoritmos de aprendizagem computacional para tarefas de mineração de dados. Os algoritmos podem ser aplicados diretamente a um conjunto de dados ou chamados utilizando código na linguagem Java. A Weka contém ferramentas para pré-processamento, classificação, regressão, clusterização, regras de associação, e visualização. Ela também é adaptada para desenvolver novas formas de aprendizagem computacional (HALL et al., 2009).

A figura 4 mostra um exemplo de uso da Weka no qual foram utilizados alguns filtros para selecionar as características, dentre as quais algumas podem ser vistas no quadro “Atributes” da figura.

Para realizar os experimentos é necessário iniciar a Weka e abrir o ambiente *explorer* , um ambiente de exploração de dados. Depois é necessário abrir o arquivo no formato usado pelo software Weka na aba “preprocess” com os tuítes que foram pré-processados anteriormente. Então é necessário utilizar um filtro para pré-processar os tuítes na ferramenta e extrair algumas medidas dos dados e as características deles (no caso, construir o vocabulário da sacola de palavras) e pode-se refinar e filtrar as características usando outros filtros ou algoritmos de seleção de características. Depois deve-se abrir a aba “classify” para escolher o classificador e os seus parâmetros e escolher a técnica de amostragem para realizar o treinamento e validação do classificador. Após a realização de alguns ajustes nos parâmetros dos filtros utilizados para selecionar e ordenar as características são definidos os modelos de classificação que serão utilizados sobre os conjuntos de teste para aferir o desempenho do classificador aprendido.

Os experimentos realizados na Weka exigiram conhecimentos de aprendizado de

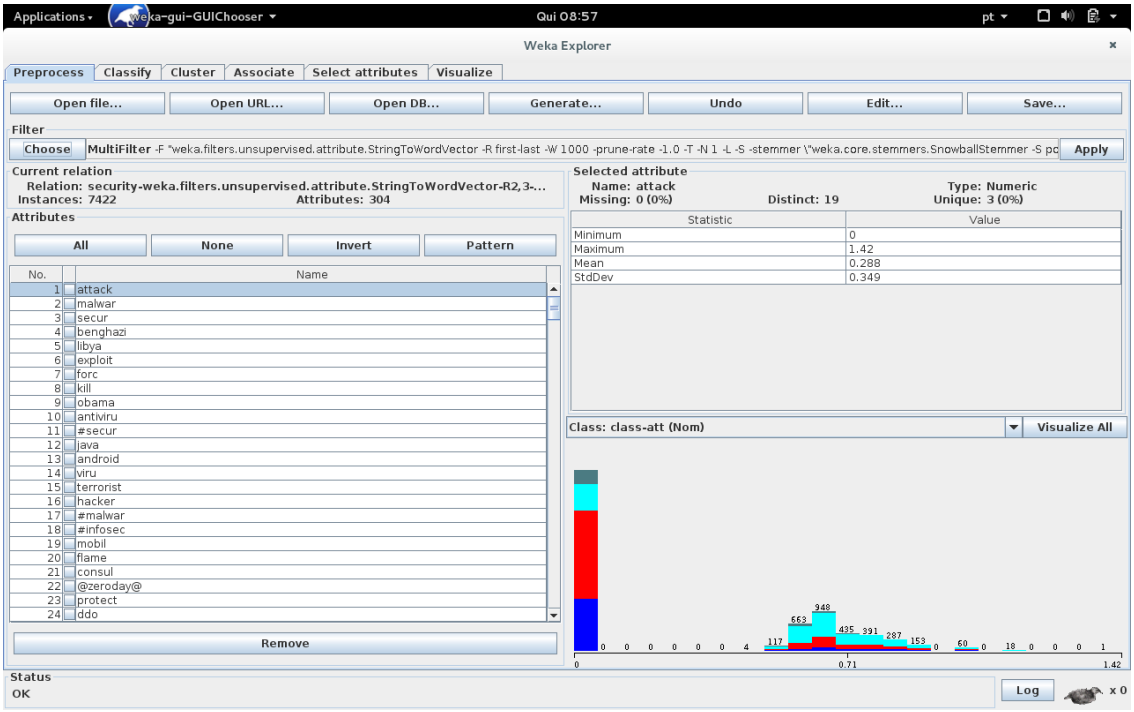


Figura 4 – Exemplo de tela do explorer da Weka

máquina de (MITCHELL, 1997), de processamento de linguagem natural e recuperação de informação presentes em (MANNING; RAGHAVAN; SCHüTZE, 2009) e do domínio do problema (segurança virtual) e de outros conceitos aprendidos durante o trabalho por meio de outras bibliografias.

4.2 Experimentos e resultados

Para realizar a fase de treinamento e de validação foi utilizado o conjunto de treinamento e validação com 7422 tuítes já pré-processados e armazenados em um arquivo no formato usado pelo software Weka. Foi utilizado o filtro "StringToWordVector" que transforma os documentos em um dicionário de palavras. Após a realização de experimentos utilizando os classificadores SVM e Naive Bayes a configuração dos parâmetros do filtro "StringToWordVector" ficaram como segue abaixo:

- TFTransform = true (Uso do método de ponderação $tf = log(1 + ftd)$)
- attributeIndices = first-last (primeiro ao último termo)
- lowerCaseTokens = true (transformar todas as letras em letras minúsculas)
- minTermFreq = 1 (para NB) e minTermFreq = 0 (para SVM)
  (minTermFreq é número mínimo da ocorrência de um termo no documento de um termo presente no dicionário. Utilizar minTermFreq = 1 equivale a utilizar a suavização de Laplace.)

- normalizeDocLength = allData (normalizar o tamanho dos documentos dos dados de alidação e treinamento)
- stemmer = Snowball (PORTER, 2001)
- stopwords (lista padrão de *stop words* da Weka. Ela foi omitida por ser grande.)
- Tokenizador padrão da Weka usando os tokens " `\n\r\t`.,;:?!()"
- useStoplist = true
- wordsToKeep = 1000 (número máximo de termos que podem ser mantidos por cada classe)

A biblioteca Snowball foi adicionada à Weka para utilizar o algoritmo de Porter recomendado por (MANNING; RAGHAVAN; SCHüTZE, 2009).

Para selecionar as características que contribuirão para separar melhor as classes foi utilizada a métrica *information gain*. Ela seleciona as características que melhor separam as classes umas das outras, elimina características não informativas, evita a ocorrência de *noise features* e diminui a chance de ocorrer *overfitting* na classificação dos dados. Essa métrica foi avaliada como uma das melhores para seleção de características segundo (ROGATI; YANG, 2002).

Para aplicar métrica *information gain* foi utilizado o seletor de atributos "InfoGainAttributeEval" com as opções padrão associadas e o método de busca de atributos chamado "Ranker" foi associado à métrica *information gain* para selecionar as características que serão mantidas no dicionário. Ele seleciona as características que possuem um valor mínimo (parâmetro "threshold") que é obtido utilizando "InfoGainAttributeEval", ordena as característica de acordo com essa métrica e seleciona as n maiores (parâmetro "num to select") características.

Em "Ranker" foram adotadas duas configurações distintas para fazer uma comparação de experimentos. As configurações utilizadas foram as seguintes:

Configuração 1:

- threshold = 0.001
- num to select = 310

Configuração 2:

- threshold = 0.001
- num to select = 1000

Depois de aplicar o "Ranker" em ambas as configurações de processamento dos dados são removidos alguns termos que são considerados causadores de *overfitting*.

Assim, temos um conjunto com 7422 tuítes e um vocabulário de 304 termos e o mesmo conjunto de tuítes com um vocabulário de 990 tuítes.

Após realizar alguns experimentos em ambos os conjuntos de dados os experimentos com o classificador *Naive Bayes* (NB) que obtiveram melhor desempenho foram.

- NB com 310 características e *holdout* de 85%:15%
- NB com 310 características e com 11 *folds*
- NB com 1000 características e *holdout* de 85%:15%
- NB com 1000 características e com 10 *folds*

Para utilizar o classificador SVM foi utilizada a biblioteca libsvm (CHANG; LIN, 2011) adicionada à Weka. Os parâmetros relevantes configurados para executar o classificador SVM são:

- Tipo de SVM = C-SVC (corresponde ao modelo SVM apresentado no trabalho)
- Tipo de *kernel function* = linear: $u' * v$
- Normalizar dados de entrada = true
- C = 1 (termo de regularização)

Os experimentos realizados com o SVM que obtiveram melhor desempenho foram:

- SVM com 310 características e *holdout* de 85%:15%
- SVM com 310 características e com 4 *folds*
- SVM com 1000 características e *holdout* de 85%:15%
- SVM com 1000 características e com 8 *folds*

Após definir os experimentos os classificadores foram treinados e as estatísticas mais significativas dos resultados seguem na tabela 5:

A tabela 5 exibe uma comparação de desempenho entre os classificadores na fase de validação dos classificadores treinados. A primeira coluna possui os nomes dos classificadores, tamanho do dicionário, método de amostragem e porcentagem de dados usados na fase de treinamento (no caso do *holdout*) ou número de *folds* utilizados (no caso de uso da validação cruzada sem reposição). Os valores de cobertura (verdadeiro positivo),

Fase de treinamento e validação	Acurácia	Cobertura (ASV)	Precisão (ASV)	Cobertura	Precisão	*F-measure*
NB-310-85%-holdout	75.3819%	0.825	0.716	0.754	0.743	0.739
NB-310-11-fold	75.1954%	0.855	0.713	0.752	0.742	0.736
NB-1000-85%-holdout	79.425%	0.855	0.762	0.794	0.792	0.788
NB-1000-10-fold	79.6281%	0.882	0.764	0.796	0.793	0.786
SVM-310-85%-holdout	75.4717%	0.818	0.723	0.755	0.746	0.746
SVM-310-4-fold	75.1684%	0.837	0.723	0.752	0.742	0.739
SVM-1000-85%-holdout	79.6047%	0.834	0.782	0.796	0.79	0.791
SVM-1000-8-fold	79.4395%	0.849	0.783	0.794	0.789	0.789

Tabela 5 – Comparação de desempenho dos classificadores SVM e NB na fase de treinamento e validação com tamanhos de dicionário e tipos de validação diferentes

precisão e da *F-measure* são o resultado das médias dos respectivos valores de cada classe ponderadas pelo número de documentos das respectivas classes.

A comparação da tabela mostra que para determinado classificador com mesmo número de termos e diferentes técnicas de amostragem a acurácia é aproximadamente igual e que o aumento no tamanho do vocabulário aumenta a acurácia do classificador independentemente da técnica de amostragem utilizada. Não há uma diferença significativa de acurácia entre classificadores diferentes, mas com mesmo tamanho de vocabulário. A cobertura possui valores entre 0.820 e 0.850 a precisão fica entre 0.710 e 0.780 para os ASVs. Isso mostra que a classe dos ASVs possui um bom grau de separação entre as classes entre razoável a bom. Também nota-se que a precisão varia mais que a cobertura e que ambas aumentam com um aumento do tamanho do dicionário. A cobertura ponderada pelas classes do problema possui a mesma variância da precisão dos ASVs, mas é menor entre 0.4 e 0.7 que a cobertura dos ASVs o que mostra que há classes com baixa cobertura, e portanto mal definidas. A precisão ponderada é menor que a dos ASVs, o que mostra que há classes mais bem definidas na fase de validação que a dos ASVs. A *F-measure* aumenta com o aumento do vocabulário utilizado e ela é semelhante entre NB e SVM para vocabulários do mesmo tamanho.

Fase de treinamento e validação	a	b	c	d
Notícia de segurança virtual = a	978	528	43	15
Alerta de segurança virtual (ASV) = b	230	2568	99	13
Notícia de segurança geral = c	49	174	2233	20
Spam = d	132	92	117	131

Tabela 6 – Matriz de confusão da fase de treinamento e validação com o classificador NB com $|\mathbb{V}| = 1000$ e uso de amostragem por validação cruzada com 10 *folds*

A dispersão dos dados em todos os experimentos realizados possui dispersão de dados semelhante a da matriz do algoritmo NB com $|\mathbb{V}| = 1000$ e uso de amostragem por validação cruzada com 10 *folds* da tabela 6. Nota-se que a classe de notícia de segurança geral e ASVs são as mais bem definidas do problema, enquanto que os documentos da classe spam estão distribuídos entre todas as classes e que muitos documentos da classe

notícia de segurança geral são identificados como ASVs.

A fase de teste foi dividida em 2 partes, cada uma delas com 1012 tuítes, seguindo a divisão da rotulação da fase de teste. Os resultados da parte 1 da fase de teste estão na Tabela 7:

Fase de teste parte 1	Acurácia	Cobertura (ASV)	Precisão (ASV)	Cobertura	Precisão	*F-measure*
NB-310	70.6522%	0.848	0.648	0.707	0.705	0.688
NB-1000	73.6166%	0.864	0.68	0.736	0.743	0.721
SVM-310	71.8379%	0.848	0.665	0.718	0.714	0.7
SVM-1000	74.5059%	0.861	0.704	0.745	0.74	0.73

Tabela 7 – Comparação de desempenho dos classificadores SVM e NB da primeira parte da fase de teste com tamanhos de dicionário de tamanho 310 e 1000.

A tabela 7 exibe uma comparação de desempenho da parte 1 da fase de teste entre os classificadores com melhor desempenho na fase de validação (todos com amostragem por *holdout*). A primeira coluna possui os nomes dos classificadores e tamanho do dicionário usados. Os valores de cobertura (verdadeiro positivo), precisão e da *F-measure* são o resultado das médias dos respectivos valores de cada classe ponderadas pelo número de documentos das respectivas classes.

A acurácia dos classificadores diminuiu entre 4% e 5% em comparação com a fase de teste, mas ela continua maior para classificadores com maior número de características. A acurácia dos classificadores SVM é sutilmente maior (cerca de 1%) do que a acurácia dos classificadores NB. A taxa de cobertura dos ASVs também continua um pouco maior para classificadores de vocabulário maior, mas é igual entre SVM e NB com mesmo número de características e ela continua a mesma em comparação com a fase de validação. Porém, a cobertura geral e a precisão dos ASVs e geral diminuíram aproximadamente de 0.5 em comparação com a fase de validação para todos os classificadores na fase de teste. Isso significa que a sobreposição entre as classes começou a aumentar. No caso dos ASVs em particular, a mesma porcentagem de ASVs continua sendo corretamente identificada, mas aumentou a porcentagem de tuítes de outras classes que são equivocadamente considerados ASVs.

Fase de teste parte 1	a	b	c	d
Notícia de segurança virtual = a	131	94	11	2
Alerta de segurança virtual (ASV) = b	31	311	16	3
Notícia de segurança geral = c	4	20	285	8
Spam = d	26	17	26	27

Tabela 8 – Matriz de confusão da fase de teste parte 1 do classificador SVM com $|\mathbb{V}| = 1000$ e uso de amostragem por validação cruzada com 8 *folds*

A matriz de confusão da tabela 8 é representativa da distribuição dos dados entre as classes para todos os classificadores na fase de teste parte 1. As características dela

são bastante semelhantes às da matriz de confusão da fase de teste e a única diferença significativa entre elas é que uma taxa maior de tuítes da classe de notícia de segurança virtual são identificados como ASVs, o que é a provável causa da diminuição do valor de todas as estatísticas gerais.

Os resultados da fase de teste parte 2 seguem na Tabela 9:

Fase de teste parte 2	Acurácia	Cobertura (ASV)	Precisão (ASV)	Cobertura	Precisão	*F-measure*
NB-310	48.7154%	0.914	0.42	0.487	0.577	0.441
NB-1000	54.7431%	0.898	0.46	0.547	0.626	0.514
SVM-310	49.9012%	0.882	0.45	0.499	0.599	0.463
SVM-1000	55.5336%	0.859	0.497	0.555	0.623	0.53

Tabela 9 – Comparação de desempenho dos classificadores SVM e NB da segunda parte da fase de teste com tamanhos de dicionário de tamanho 310 e 1000.

A tabela 9 exibe uma comparação de desempenho da parte 2 da fase de teste entre os classificadores com melhor desempenho na fase de validação (todos com amostragem por *holdout*). A primeira coluna possui os nomes dos classificadores e tamanho do dicionário usados. Os valores de cobertura (verdadeiro positivo), precisão e da *F-measure* são o resultado das médias dos respectivos valores de cada classe ponderadas pelo número de documentos das respectivas classes.

Comparando os resultados da fase de teste parte 1 e da parte 2 temos que a acurácia diminuiu cerca de 20% para todos os classificadores e que a diferença entre a acurácia para o mesmo classificador com vocabulário de tamanhos aumentou de aproximadamente 3%. A taxa de cobertura dos ASVs, porém, aumentou de 0.03 com exceção do classificador SVM com 1000 termos cuja cobertura se manteve constante. Por outro lado, a cobertura ponderada sofreu uma diminuição de 0.2 e a precisão dos ASVs por outro lado diminuiu de 0.2 para os ASVs e a precisão ponderada caiu entre 0.11 e 0.13. As outras diferenças entre os classificadores foram mantidas entre as fases de teste parte 1 e 2.

Fase de teste parte 2	a	b	c	d
Notícia de segurança virtual = a	170	184	47	8
Alerta de segurança virtual (ASV) = b	19	269	25	0
Notícia de segurança geral = c	7	32	94	3
Spam = d	37	56	32	29

Tabela 10 – Matriz de confusão da fase de teste parte 2 do classificador SVM com $|\mathbb{V}| = 1000$ e uso de amostragem por validação cruzada com 8 *folds*

A matriz de confusão da tabela 10 é representativa da distribuição dos dados entre as classes para todos os classificadores na fase de teste parte 2. A matriz evidencia uma alta superposição da classe dos ASVs sobre a classe notícia de segurança geral de forma que há mais tuítes identificados como ASVs como pertencentes a sua real classe. Como pode-se ver, a dispersão dos ASVs entre as outras classes diminuiu, o que justifica

o aumento da taxa de cobertura, enquanto que a classe notícia de segurança geral, pelo contrário, se tornou menos separável das outras classes, principalmente dos ASVs.

	Treinamento e validação	Teste parte 1	Teste parte 2
Notícia de segurança virtual	1564 (21%)	238 (24%)	409 (40%)
Alerta de segurança virtual (ASV)	2910 (39%)	361 (36%)	313 (31%)
Notícia de segurança geral	2476 (33%)	317 (31%)	136 (14%)
Spam	472 (7%)	96 (9%)	154 (15,9%)

Tabela 11 – Distribuição dos tuítes entre as classes do problema em cada uma das fases do desenvolvimento do classificador

Na tabela 11 pode-se perceber que a distribuição dos documentos entre as classes varia pouco entre o conjunto de treinamento e validação e o conjunto de teste parte 1, enquanto que a entre o conjunto de treinamento e validação e o conjunto de teste da parte 2 a distribuição dos documentos entre as classes varia bastante. A mudança observada nas proporções de cada classe pode ser explicada pelo fato de os tuítes da fase de teste parte 2 terem sido coletados em uma época diferente dos outros conjuntos de tuítes. Esta afirmação considera que os tuítes rotulados foram escolhidos aleatoriamente e supõe-se que as proporções dos documentos entre as classes no problema se repetem no Twitter.

4.3 Discussões

Os resultados não apresentaram uma acurácia alta, mas existem alguns indicadores de que os maiores problemas para separar bem as classes está na quantidade de dados utilizados e na representação dos documentos. Isso pode ser aferido por algumas razões, dentre as quais está o fato de que o aumento no tamanho do dicionário melhorou a acurácia dos classificadores. Este fato é um bom indicativo de que os classificadores utilizados são adequados para tratar do problema, pois classificadores lineares como SVM e NB (no espaço dos logaritmos) tendem a separar melhor as classes conforme o número de dimensões do espaço aumenta, o que é um indício de que as classes são linearmente separáveis.

Além disso, a quantidade de dados utilizados para treinamento foi bastante pequena, pois foram utilizados 7422 tuítes na fase de validação e treinamento, que em média possuem 90 caracteres totalizando 2,6 Mbytes de dados. Outro argumento no sentido de que foram utilizados poucos dados para treinar o classificador é que no ajuste dos parâmetros do filtro responsável por selecionar os termos do dicionário o uso *stemming* contribuiu para separar melhor as classes, o que em geral acontece quando o espaço dos dados é esparso.

A 2ª fase de teste obteve resultados bastante piores que a 1ª fase de teste devido

a dois fatores: Aumentou a presença de tuítes pertencentes a classes mais mal definidas como spam e notícia de segurança virtual além de uma diminuição na cobertura e precisão delas; Houve uma desconcentração do conjunto de tuítes pertencentes a classe de notícia de segurança geral que possui alta precisão e cobertura nos outros conjuntos de tuítes. Isso significa que o classificador é bastante sensível a variação de tempo entre os dados usados no treinamento e no teste do classificador e que a representação dos documentos pode ter sido um pouco alterada com alguns termos se tornando mais relevantes para determinadas classes em épocas diferentes e outros se tornaram menos importantes.

Apesar de tudo isso, a taxa de cobertura dos ASV demonstrou uma tendência de aumento apesar da precisão ter diminuído bastante. isso significa que o classificador se tornou ainda melhor em identificar ASVs, mas que tuítes de outras classes se tornaram mais confundíveis com ASVs, o que deve ter sido causado pela mudança na representação da informação das outras classes, principalmente a de notícia de segurança virtual e os classificadores não puderam detectar isso.

Porém, existe um fator detectado na fase de rotulação dos tuítes que influenciou muito negativamente em todas as fases (treinamento, validação e teste) de construção do classificador e que se mostra como um dos maiores desafios na construção dos classificadores que são as representações viesadas de informação. Como os textos do tuíte são livres um tuíte pode ser um comentário sobre uma notícia e tal comentário em muitos casos representa muito mal a informação à qual ele se refere. Em outras palavras, o conjunto de termos que representa o documento faz com que ele não seja identificado pelo classificador como pertencente a sua real classe. Em muitos dos casos foi verificado que manchetes de notícias dos links presentes nestes tuítes representam muito melhor a informação que o tuíte busca transmitir.

5 Conclusão

O estudo dos dados e a construção dos classificadores contribuiu para um melhor entendimento do problema de identificar ASVs. A comparação dos classificadores mostrou que o SVM e o NB possuem desempenho similar na detecção dos ASVs e que apesar da dificuldade de separar bem as classes do problema a taxa de cobertura dos ASVs se manteve alta e constante em todos os experimentos realizados.

O trabalho também proporcionou um aprendizado rico de desafios e dificuldades que se pode encontrar em um problema de recuperação de informação envolvendo aprendizado de máquina e colaborou para que fossem feitas reflexões e a elaboração de hipóteses para explicar os problemas encontrados. Dentre os problemas identificados estão a esparsidade dos dados devido à quantidade de tuítes utilizadas para treinar o classificador e a necessidade de fazer um processamento linguístico mais apurado para obter características que separem melhor as classes.

Finalmente, ficou claro que é muito importante que os dados do problema possuam uma distribuição representativa do espaço de informações para construir um classificador com maior acurácia. Para isso é necessário utilizar dados coletados em um intervalo de tempo suficientemente longo para que os termos possuam pesos mais representativos do conjunto de dados e as classificações sejam menos sensíveis a documentos coletados em diferentes intervalos de tempo.

Ainda há muito a se fazer, pois além de existirem alguns problemas que devem ser resolvidos ou diminuídos para melhorar os classificadores também existe a necessidade de coletar dados em outras redes sociais para que o classificador tenha mais fontes de informação para tornar o espaço de dados menos esparso e assim, fazer com que os classificadores consigam separar melhor as classes.

Parte II

Parte subjetiva

Desafios e frustrações

Entre os problemas encontrados para desenvolver o trabalho os maiores foram descobrir qual área eu precisava estudar para entender os conceitos do trabalho. Por desconhecer que o problema é de recuperação de informação eu não sabia como interpretar e processar o texto dos tuítes. A falta de experiência na área de segurança computacional me deixou bastante desconfortável para rotular os tuítes mesmo depois de estudar do que se trata um ASV, entre outros conceitos da área. Por causa disso, foi necessário descartar conjuntos de tuítes rotulados mais de uma vez sendo que o maior conjunto descartado continha aproximadamente 2000 tuítes.

Considero como grave a dificuldade de me dedicar ao projeto o quanto gostaria devido a outras atividades que eu conduzia durante o trabalho. Sem dúvida alguma ser representante discente e se engajar em atividades relacionadas ao curso é algo que não combina com foco em estudos. Por causa disso, fui obrigado a revisar a bibliografia algumas vezes para lembrar o que eu estava fazendo e para escrever a monografia. De certa forma, todo o trabalho foi bastante prejudicado pelas várias quebras de fluxo do desenvolvimento das atividades.

Outro problema importante é o desvio de foco do que é realmente importante no trabalho. Um exemplo disso foi o esforço que me exigiu bastante tempo na busca de um classificador de línguas que seria usado no script de rotulação dos tuítes, pois nem todos os tuítes da base de dados estavam em língua inglesa. Este classificador não era tão importante, pois eu conseguia identificar se um tuíte estava escrito em inglês ou em outra língua ao lê-lo. Ao final o classificador desenvolvido não foi utilizado, pois por problemas com o encoding nos textos eu não consegui fazer com que ele funcionasse para línguas com letras que não pentencessem ao alfabeto latino.

Relação entre o trabalho e as disciplinas cursadas no BCC

- MAC0110 Introdução à Computação
- MAC0122 Princípios de Desenvolvimento de Algoritmos
  foram importantes para aprender lógica de programação e para aprender a trabalhar com algumas estruturas de dados como listas, vetores, tabelas de hash, também como, entender ordenação e para aprender a trabalhar com arquivos
- MAC0211 Laboratório de Programação I
  foi importante para aprender a utilizar filtros, escrever documentos em latex, adquirir boas práticas de programação como a modularização de código, utilizar scripts para realizar tarefas rápidas como compilar arquivos .tex e fazer processamento de strings
- MAT0139 Álgebra Linear para Computação
  foi importante para conhecer o espaço vetorial, entender o que é produto interno, distância euclidiana, norma, hiperplano, entre outras coisas
- MAC0315 Programação linear
  Entender minimização de função linear e variáveis residuais
- MAE0121 Introdução a Probabilidade e Estatística I e
- MAE0212 Introdução à Probabilidade e Estatística II
  foram importantes para entender o que são eventos independentes e aleatórios, além de compreender algumas distribuições de probabilidade como a uniforme e qui-quadrado
- MAC0459 Ciência e Engenharia de Dados
  foi importante para aprender a escrever melhor textos científicos e entender recuperação de informação
- MAC0460 Aprendizagem Computacional: Modelos, Algoritmos e Aplicações
  foi importante para aprender aprendizado de máquina, conhecer os classificadores, estatísticas de acerto na classificação de objetos, matriz de confusão, entre outras coisas

Trabalhos futuros

- Usar o Open Calais para identificar entidades presentes nos tuítes a fim de melhorar o classificador (feature engeneering)
- Verificar se os próprios tuítes são uma ameaça de segurança (por exemplo, phishing) como em (WHITE; MATTHEWS, 2013).
- Buscar correlação entre tipo de api utilizada para publicar os tuítes (WHITE; MATTHEWS, 2013).
- Detectar acrônimos para substituir por palavras.
- Detectar uso de sinônimos (wordnet), datas e horário para aplicar *feature engenneering*.
- Coletar mais tuítes usando pessoas para fazê-lo e submetê-las a um controle para ver as pessoas que realmente sabem o que estão fazendo das que não sabem.
- Usar mais redes sociais para obter dados para treinar os classificadores.
- Estudar o uso de outros classificadores para identificar não-ASVs.
- Estudar atribuição de pesos especiais a alguns tokens.
- Adicionar correção ortográfica.
- Capturar o conteúdo das manchetes (*headlines* dos links contidos nos tweets para utilizar como fonte de informação.

Referências

BOYD, D. M.; ELLISON, N. B. Social network sites: Definition, history, and scholarship. *J. Computer-Mediated Communication*, v. 13, n. 1, p. 210–230, 2007. Citado na página 23.

BUSSAB, W. de O.; MORETTIN, P. *Estatística básica*. [S.l.]: Saraiva, 2013. ISBN 9788502034976. Citado na página 79.

CERT.PT; CSIRT, R. N. *Taxonomia Comum para a Rede Nacional de CSIRTs*. 2012. <www.cert.pt/images/docs/Taxonomiav2.5.pdf>. Citado 2 vezes nas páginas 27 e 28.

CHANG, C.-C.; LIN, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, v. 2, p. 27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. Citado na página 50.

FUTURE, D. of H. S. *Blueprint for a Secure Cyber Future: The Cybersecurity Strategy for the Homeland Security Enterprise*. [S.l.], 2011. Citado na página 27.

GLOSSARY of IT Security Terminology Terms and definitions. [S.l.]: TeleTrusT Germany, 2009. <http://www.teletrust.de/uploads/media/ISOIEC_JTC1_SC27_IT_Security_Glossary_TeleTrusT_Documentation.pdf>. Citado 2 vezes nas páginas 27 e 28.

HALL, M. et al. The WEKA data mining software: an update. *SIGKDD Explorations*, v. 11, 2009. Issue 1. Citado na página 47.

HEADQUARTERS, U. S. D. of A. *Field Manual 3-19.30: Physical Security*. 2001. <http://www.globalsecurity.org/military/library/policy/army/fm/3-19-30/ch1.htm>. Capítulo 1. Citado na página 27.

HIDALGO, J. M. G. *Nihil Obstat*. 2013. <http://jmgomezhidalgo.blogspot.com.br/>. Citado na página 47.

HSU, C.-W.; LIN, C.-J. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks*, v. 13, n. 2, p. 415–425, 2002. Citado na página 41.

IMPERVA. *Imperva's Hacker Intelligence Summary Report: The Anatomy of an Anonymous Attack*. [S.l.], 2012. Citado na página 19.

MANNING, C. D.; RAGHAVAN, P.; SCHüTZE, H. *An Introduction to Information Retrieval*. Draft. [S.l.]: Cambridge University Press, 2009. Citado 5 vezes nas páginas 11, 36, 41, 48 e 49.

MINISTéRIO da Justiça - Órgão de Segurança. 2013. <http://portal.mj.gov.br/> em Órgãos de Segurança, Conceitos Básicos. Accessado: 26-10-13. Citado na página 26.

MITCHELL, T. M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072. Citado 3 vezes nas páginas 31, 32 e 48.

MOORE, A. A short tutorial note on computing information gain from counts. Unpublished. 1994. Citado na página 31.

PALERI, P. National security: Imperatives and challenges. In: _____. [S.l.]: Tata McGraw-Hill, 2008. p. 57. ISBN 9780070656864. Citado na página 26.

PARANAGAMA, S. *Training and test set are not compatible*. 2013. <https://thekandyancode.wordpress.com/tag/training-and-test-set-are-not-compatible/>. Citado na página 47.

PORTER, M. F. *Snowball: A language for stemming algorithms*. 2001. Disponível em: <http://snowball.tartarus.org/texts/introduction.html>. Citado na página 49.

ROGATI, M.; YANG, Y. High-performing feature selection for text classification. In: *Proceedings of the Eleventh International Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2002. (CIKM '02), p. 659–661. ISBN 1-58113-492-4. Disponível em: <http://doi.acm.org/10.1145/584792.584911>. Citado na página 49.

SANTOS, L. A. F. et al. Análise de mensagens de segurança postadas no twitter. *Anais do simpósio brasileiro de sistemas colaborativos (SBSC)*, n. 3, p. 20–28, 2012. Citado 2 vezes nas páginas 18 e 44.

SECURITY, U. D. of H. *Privacy Impact Assessment for the Initiative Three Exercise*. [S.l.], 2010. Citado na página 27.

SHIRLEY, R. *Internet Security Glossary*. 2007. <http://www.ipa.go.jp/security/rfc/RFC4949-00EN.html>. Citado na página 27.

STRATEGIC, C. for; STUDIES, I. *The Economic Impact of Cybercrime and Cyber Espionage*. [S.l.], 2013. Citado na página 18.

TWITTER Help Center - Get started: FAQs and the basics. 2013. <https://support.twitter.com/groups/50-welcome-to-twitter>. Acessado: 28-09-13. Citado na página 23.

WHITE, J. S.; MATTHEWS, J. N. It's you on photo?: Automatic detection of twitter accounts infected with the blackhole exploit kit. In: *MALWARE*. [S.l.]: IEEE, 2013. p. 51–58. ISBN 978-1-4799-2534-6. Citado na página 65.

WIKIPEDIA: Confusion Matrix. <http://en.wikipedia.org/wiki/Confusion_matrix>. Acessado: 29-01-15. Citado na página 35.

WIKIPEDIA: definição de segurança pessoal. <http://pt.wikipedia.org/wiki/Seguran%C3%A7a_pessoal>. Acessado: 29-01-15. Citado na página 27.

WIKIPEDIA: Entropy. <http://en.wikipedia.org/wiki/Entropy_(information_theory)>. Acessado: 29-01-15. Citado na página 31.

WIKIPEDIA: Statistical classification. <http://en.wikipedia.org/wiki/Statistical_classification>. Acessado: 29-01-15. Citado na página 32.

WIKIPEDIA: Supervised Learning. <http://en.wikipedia.org/wiki/Supervised_learning>. Acessado: 29-01-15. Citado na página 33.

WILSHUSEN, G. C. *Cybersecurity: Continued Attention Needed to Protect Our Nation's Critical Infrastructure*. [S.l.], 2011. Citado na página 27.

WILSHUSEN, G. C. *Cyber Threats Facilitate Ability to Commit Economic Espionage*. [S.l.], 2013. Citado 2 vezes nas páginas 27 e 28.

WOLF, F.; POGGIO, T.; SINHA, P. *Human Document Classification Using Bags of Words*. 2006. Disponível em: <http://hdl.handle.net/1721.1/33789>. Citado na página 75.

Apêndices

APÊNDICE A – Análise da pesquisa sobre detecção de Alertas de segurança virtual do Twitter

A.1 Perguntas e descrição da pesquisa

A pesquisa consiste de 13 questões. Entre elas há 10 tuítes e as outras 3 questões servem para filtrar os elementos da amostra (participantes que estudam computação ou trabalham na área), saber se o mesmo considera ter conhecimentos sobre ASVs e coletar sugestões e críticas sobre a pesquisa. Também é associado a cada participante da pesquisa o tempo, em segundos, gasto para preencher a pesquisa.

Todas as perguntas da pesquisa são obrigatórias exceto a que solicita o envio de sugestões ou críticas.

A.1.1 População alvo e amostra

A população de interesse são alunos de computação, professores e profissionais da área de computação sem necessariamente possuírem experiência em segurança virtual (*cybersecurity*). Para se obter a amostra, foram contactados os alunos e professores do IME e profissionais fora da comunidade USP via e-mail e redes sociais e, por sua vez, algumas das pessoas contactadas enviaram a pesquisa a conhecidos e colegas relacionados à área de computação.

Assim, observa-se que a amostra não é aleatória no seu sentido literal, pois ela é composta pelas pessoas com as quais eu consegui entrar em contato e algumas outras que souberam da pesquisa por meio de alguém que já tinha sido contactado por mim. Todos os respondentes participaram da pesquisa voluntariamente.

A.1.2 Descrições das questões

Os 10 tuítes utilizados na pesquisa estão todos escritos em língua inglesa. Abaixo a lista de tuítes e as respostas esperadas na classificação deles:

1. “How secure are Apple’s iPhone and iPad from malware, really? | Naked Security http://t.co/rluCYs7H” – ASV

2. “The heart attack I get when the security alarm sensor falls and I think / someones coming in...had my...” – não-ASV

3. “RT @securityaffairs: DDoS attacks in Q2, do not underestimate the cyber threat http://t.co/04ThaJ0q...” – ASV

4. “Researchers find malware targeting online stock trading software: Security research... http://t.co...” – ASV

5. “#Apple iOS Zero-day exploit sold for $500,000 http://t.co/NOFfIM4D36 #0day #iphone #security #vulner...” – ASV

6. “Possibilities for Malicious Browser Extensions Are Almost Infinite, Researcher Says: The Hacker Halt...” – ASV

7. “Lubbock, Tech officials mainstening security after Boston attack http://t.co/dIRfj9UbGC” – não-ASV

8. “Sen. Rockefeller questions cyber security of critical infrastructure after attack on gas pipelines:...” – não-ASV

9. “Homeland Security will track this article if I say electric pork cloud virus. oops. http://t.co/AcCK...” – não-ASV

10. “Avast! Avast Antivirus 8.0.1489 Virus Definitions Update Download (August 24, 2013) KEYGURU http://n...” – não-ASV

As classes às quais os tuítes podem pertencer são mutuamente exclusivas e elas são as seguintes: ASVs, Spam, notícia de segurança geral e notícia de segurança virtual. Porém, para todos os efeitos, nesta pesquisa as classes adotadas são ASV e não-ASV, que é formada pelos grupos restantes de tuítes. Os tuítes foram divididos entre 5 ASVs e 5 não-ASVs.

Os participantes classificam cada um dos tuítes utilizando a noção que cada um deles possui sobre ASVs. Ou seja, as classificações são feitas sem que lhes tenha sido apresentada uma definição de ASVs.

As respostas possíveis para cada tuíte são:

1. Alerta de segurança virtual

2. Não é alerta de segurança virtual

3. Não sei

A opção "Não sei" pode incluir um comentário opcional do respondente.

Antes de fazer a classificação de cada tuíte são apresentadas duas questões sobre o perfil do respondente, além da inclusão do tempo gasto pelo participante para preencher a pesquisa. O tempo associado a cada participante é registrado pela Qualtrics[1], plataforma que armazena as respostas da pesquisa. A primeira pergunta solicita a área de atuação do respondente, pois apenas pessoas relacionadas à área de computação (estudantes de qualquer nível educacional, profissionais ou docentes/pesquisadores) serão analisadas na amostra coletada por se considerar que tais pessoas são, em geral, mais capacitadas para fazer a classificação dos tuítes (possíveis ASVs). A segunda pergunta busca verificar se o respondente considera ter (ou não) conhecimentos sobre ASVs. O tempo de preenchimento da pesquisa é utilizado para aferir se o participante teve tempo suficiente para ler e pensar na escolha das classificações dos tuítes. O tempo passa a ser contado a partir do momento em que o participante interage com a pesquisa passando o cursor na aba em que ela foi aberta ou digitando alguma tecla. A sua contagem é finalizada quando as respostas são enviadas à plataforma da pesquisa.

A.2 Análise e resultados da pesquisa

A análise dos resultados foi feita utilizando questões de controle e as respostas da questão em que o respondente declara se possui conhecimento sobre segurança virtual considerando a hipótese de que pessoas que afirmam ter boas noções de segurança realmente são bons idenficadores de ASVs. Considerando o relatório (WOLF; POGGIO; SINHA, 2006) que mostrou que seres humanos possuem acurácia média de aproximadamente 80% para identificar tópicos de notícias de jornal sem limite de tempo vamos considerar que bons identificadores de ASVs e não-ASVs possuem taxa de acerto superior a 80%.

Foram escolhidos para servir de controle os Tuítes 2 e 7, que devem ser classificados como não-ASV. O objetivo do controle é separar os participantes que conseguem responder ambos os tuítes corretamente dos que não conseguem considerando a hipótese de que os respondentes reprovados no controle são considerados como possuidores de um baixo nível de familiaridade com ameaças de segurança virtual e suas respostas devem ser observadas com cuidado.

Ao todo 100 respondentes preencheram a pesquisa. Para compor a amostra a ser analisada foram removidas as respostas de 4 respondentes porque tratam-se de participantes que não atuam na área da computação ou relacionadas. Após a realização da análise da amostra de 96 pessoas apenas 4 (4,16%) delas, dentre as quais todas afirmaram ter noções de segurança virtual, acertaram todas as classificações de tuítes.

[1] <http://www.qualtrics.com/>

Cinco participantes preencheram a pesquisa muito rápido (menos que 60 segundos). Estes participntes provavelmente não refletiram para responder algumas questões e escolheram suas respostas de maneira aleatória ou não consciente. Três participantes levaram muito tempo (mais de 1h30) pra responder o questionário. Uma explicação possível é que eles começaram a preencher as questões, acabaram por interromper o seu preenchimento e o retomaram bastante tempo depois da interrupção. Contudo, não é possível determinar que os participantes mais lentos preencheram a pesquisa displicentemente, pois a pessoa também pode ter pesquisado sobre os assuntos relacionados ao conteúdo dos tuítes, por exemplo. Como não existe um indicador que invalide as respostas dos respondentes lentos e verificou-se que os respondentes muito rápidos possuem uma boa porcentagem de acerto na classificação dos tuítes em comparação com todos os respondentes os a participação deles foi mantida na amostra.

Ao analisar os dados resolvi comparar o desempenho dos respondentes sob dois critérios, as respostas dadas pelos respondentes para as questões de controle e a autoavaliação dos respondentes a respeito de seus conhecimentos sobre segurança virtual.

Possui noção de segurança virtual?	Aprovado no controle	Reprovado no controle
Sim	68	9
Não	16	3

Tabela 12 – Autoavaliação dos respondentes da pesquisa sobre conhecimentos de ASVs x Real conhecimento de ASVs aferido pelo controle

Para entender melhor o perfil dos participantes da pesquisa eu extraí algumas informações da tabela 12 e as coloquei nas tabelas abaixo.

Respondente considera não ter noção de segurança virtual	Aprovado no controle	Reprovado no controle
	16 (84,21%)	3 (15,79%)

Tabela 13 – Divisão dos respondentes que afirmaram não ter noções de segurança entre dois grupos: os que passaram no controle e os que foram reprovados no controle

		Reprovado no controle
Possui noção de segurança virtual?	Sim	9 (75,00%)
	Não	3 (25,00%)

Tabela 14 – Divisão dos respondentes que foram reprovados no controle entre dois grupos: os que afirmaram ter noções de segurança virtual e os que afirmaram não ter noções de segurança

A tabela 13 mostra que a maior parte (84,21%) dos respondentes que afirmaram não ter noções de segurança passaram no controle. Por outro lado, a tabela 14 mostra que 75% dos respondentes que foram reprovados no controle declararam ter noções de

segurança virtual. Isso significa que a maior parte das pessoas que foram reprovadas no controle não estão cientes de que não possuem conhecimentos suficientes para identificar ASVs. No primeiro caso, o fato de que a maior parte das pessoas passaram no controle apesar de se considerarem com pouco conhecimento sobre segurança virtual pode significar que elas adotaram um critério bastante rigoroso pra avaliarem seus conhecimentos sobre o assunto.

	Porcentagem média de acerto na classificação dos tuítes
Tem noção de segurança	71,29% dos tuítes
Não tem noção de segurança	74,57% dos tuítes

Tabela 15 – Porcentagem média de acerto dos 10 tuítes classificados pelos respondentes dos grupos de respondentes auto declarados com ou sem noção de segurança virtual

	Porcentagem média de acerto na classificação dos tuítes
Aprovadas no controle	73,80% dos tuítes
Reprovadas no controle	55,00% dos tuítes

Tabela 16 – Porcentagem média de acerto dos 10 tuítes classificados pelos respondentes dos grupos de respondentes aprovados no controle e reprovados no controle

Como pode-se perceber nas tabelas 15 e 16, os grupos de respondentes possuem uma taxa de acerto de aproximadamente 70% na classificação dos tuítes, exceto o grupo das pessoas reprovadas no controle. Isso é esperado, pois os respondentes reprovados no controle possuem pouco conhecimento para identificar ASVs e por isso obtiveram um desempenho inferior aos demais grupos.

Acerto na classificação	Tuíte 3
Respondentes com noção de segurança	59 (76,62%)
Respondentes sem noção de segurança	13 (68,42%)

Tabela 17 – Número de pessoas que acertaram a classificação do Tuíte 3 para os grupos de respondentes que declararam ter conhecimentos de segurança virtual e dos que declararam não tê-los, acompanhado de sua respectiva porcentagem em relação ao seu respectivo grupo.

Acerto na classificação	Tuíte 3
Pessoas aprovadas no controle	65 (77,38%)
Pessoas reprovadas no controle	7 (58,33%)

Tabela 18 – Número de pessoas que acertaram a classificação do Tuíte 3 para os grupos de respondentes aprovados no controle e reprovados nele, acompanhado de sua respectiva porcentagem em relação ao seu respectivo grupo.

O Tuíte 3 é facilmente classificável, mas causa um pouco de dúvida o que se observa nas porcentagens de acerto dos grupos de respondentes mostradas pelas tabelas 17 e 18,

pois se trata de um relatório (*report*) sobre ataques de negação de serviço (*DDoS*) em um determinado trimestre e não necessariamente uma ameaça ou ataque em curso a segurança virtual.

Acerto na classificação	Tuíte 4
Respondentes com noção de segurança	70 (90,90%)
Respondentes sem noção de segurança	18 (94,73%)

Tabela 19 – Número de pessoas que acertaram a classificação do Tuíte 4 para os grupos de respondentes que declararam ter conhecimentos de segurança virtual e dos que declararam não tê-los, acompanhado de sua respectiva porcentagem em relação ao seu respectivo grupo.

Acerto na classificação	Tuíte 4
Pessoas aprovadas no controle	79 (94,04%)
Pessoas reprovadas no controle	9 (75,00%)

Tabela 20 – Número de pessoas que acertaram a classificação do Tuíte 4 para os grupos de respondentes aprovados no controle e reprovados nele, acompanhado de sua respectiva porcentagem em relação ao seu respectivo grupo.

A taxa de acerto do Tuíte 4 é bastante alta em quase todos os grupos de respondentes, como mostram as tabelas 19 e 20, porque o conteúdo da mensagem trata de um tipo de ameaça bastante conhecida na área (*malwares*).

Acerto na classificação	Tuíte 1	Tuíte 5
Respondentes que afirmaram ter noções de segurança	33 (42,85%)	34 (44,15%)
Respondentes que afirmaram não ter noções de segurança	9 (48,57%)	8 (42,85%)

Tabela 21 – Número de pessoas, e sua respectiva porcentagem, que acertaram a classificação dos Tuítes 1 e 5 para o grupo dos respondentes que declararam ter conhecimentos de segurança virtual e dos que declararam não tê-los.

Acerto na classificação	Tuíte 1	Tuíte 5
Pessoas aprovadas no controle	38 (45,23%)	36 (42,85%)
Pessoas reprovadas no controle	4 (33,33%)	6 (50,00%)

Tabela 22 – Número de pessoas, e sua respectiva porcentagem, que acertaram a classificação dos Tuítes 1 e 5 para o grupo dos respondentes que foram aprovados no controle e dos que foram reprovados nele.

As tabelas 21 e 22 mostram que a taxa de acerto dos Tuítes 1 e 5 é baixa em todos os grupos de respondentes, o que é esperado. A dificuldade de identificar estes tuítes reside no fato de que alguns tipos de ASVs não são muito conhecidos, como os *zero-day exploits*, e pelo fato das notícias dos Tuítes 1 e 5 envolverem a Apple que é tida como uma empresa que produz softwares bastante seguros e por isso ASVs envolvendo produtos da

empresa seriam pouco prováveis. O fato de os respondentes reprovados no controle terem maior porcentagem de acerto no Tuíte 5 em relação aos outros grupos de respondentes pode ser interpretado como uma anomalia resultante da dificuldade de se classificar o tuíte. Tal êxito pode ter sido obtido acidentalmente, dado que os reprovados no controle possuem menos conhecimentos sobre segurança virtual que os outros grupos participantes da pesquisa.

O tuíte número 10 pode ser erroneamente considerado como ASV, mas é um Spam. A mensagem do tuíte possui uma notificação de atualização da base de dados de um antivírus (Avast). Porém, como o antivírus atualiza a base de dados automaticamente não é necessário que o usuário seja notificado da atualização via mensagem. Logo, a mensagem é interpretada como uma forma de se fazer propaganda do antivírus. O aviso só seria aceitável, e seria considerado um ASV, caso se tratasse de uma solução em particular para um problema gravíssimo e a empresa proprietária do software antivírus tivesse sido a primeira a encontrar a solução para o ASV em questão, por exemplo.

Alguns tuítes contêm expressões que possuem significado apenas em uma determinada cultura ou país. Por exemplo, a *Homeland Security* que seria algo como segurança contra terrorismo nos Estados Unidos. Portanto, é necessário levar em conta que alguns participantes podem apresentar dificuldade para preencher a pesquisa devido a uma eventual baixa familiaridade com a língua inglesa ou à falta de compreensão de algumas expressões particulares dela ou de algum povo que a tem como sua língua pátria.

Por fim, considerando que bons identificadores de ASVs possuem taxa certo na classificação dos tuítes superior a 80% será feito um teste (estatístico) de independência para verificar se há relação entre a taxa de acerto na identificação de ASVs e a aprovação ou não nos tuítes controle e um outro teste de independência entre a taxa de acerto na identificação de ASVs e a autoavaliação sobre ter ou não conhecimento sobre segurança virtual.

Para realizar os dois testes iremos utilizar a estatística qui-quadrado (BUSSAB; MORETTIN, 2013).

	AC	RC	Total
Nº de respondentes com taxa de acerto >= 80% dos tuítes	37 (33,25)	1 (4,75)	38
Nº de respondentes com taxa de acerto < 80% dos tuítes	47 (50,75)	11 (7,25)	58
Total	84	12	96

Tabela 23 – Tabela de contingência da taxa de acerto nos ASVs versus resultado do tuíte controle. Os valores em cada célula são os valores observados e os respectivos valores esperados estão entre parênteses. AC: Aprovado no Controle; RC: Reprovado no Controle.

Os níveis descritivos dos testes para as tabelas 23 e 24 são $0,018$ e $0,194$, res-

	CN	SN	Total
Nº de respondentes com taxa de acerto >= 80% dos tuítes	28 (30,48)	10 (7,52)	38
Nº de respondentes com taxa de acerto < 80% dos tuítes	49 (46,52)	9 (11,48)	58
Total	77	19	96

Tabela 24 – Tabela de contingência da taxa de acerto nos ASVs versus autoavaliação em segurança virtual. Os valores em cada célula são os valores observados e os respectivos valores esperados estão entre parênteses. CN: Com noção de segurança virtual; SN: Sem noção de segurança virtual.

pectivamente, portanto considerando um nível de significância de 5% concluímos que há relação entre a taxa de acerto nos tuítes e o resultado do tuíte controle, mas não há relação entre a taxa de acerto e a autoavaliação em segurança virtual.

Estes resultados corroboram a análise da tabela 12 segundo a qual a distinção dos respondentes entre os que se declaram conhecedores de segurança virtual e os que se consideram não conhecedores não é uma boa forma de separar os bons identificadores de ASVs. Isso também é evidenciado pelo fato do grupo de participantes que afirmaram ter um bons conhecimentos sobre segurança virtual obtiveram um desempenho similar mas inferior ao seu grupo complementar. Por outro lado, o critério de separar os respondentes em grupos aprovados e reprovados no controle correspondeu às expectativas, pois os aprovados tiveram um desempenho melhor em classificar os tuítes que os reprovados.

Desta forma, a análise dos resultados da pesquisa mostra que os respondentes não identificam bem o suficiente ASVs em língua inglesa e que a autoavaliação deles sobre o próprio conhecimento a respeito de segurança virtual é um pouco equivocada.

A.3 Considerações finais

Como minha primeira vez a trabalhar com uma pesquisa piloto eu percebi que poderia tê-la feito um pouco diferente. Eu poderia montar um perfil técnico mais completo dos participantes que o realizado nesta pesquisa com a adição de perguntas que permitissem, por exemplo, inferir melhor o nível de conhecimento que os participantes da pesquisa possuem sobre segurança virtual.

Algumas das perguntas que ajudariam a montar um perfil técnico mais detalhado dos participantes da pesquisa poderiam incluir o tempo de experiência que a pessoa possui em cada função ou emprego em que a pessoa trabalhou com segurança. Outra informação interessante é o nível educacional do participante (superior completo, mestrado, etc). Também poderia ser pedido pra que eles definissem, mesmo que de forma simples o que eles entendem por ASV.