Universidade Federal do Amazonas
Instituto de Computação
Programa de Pós-Graduação em Informática

Adeilson Souza da Silva

Detectando Comportamento Automatizado nos Tópicos de Tendência do Twitter no Brasil

Manaus
Setembro de 2015

Adeilson Souza da Silva

Detectando Comportamento Automatizado nos Tópicos de Tendência do Twitter no Brasil

Trabalho apresentado ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas como requisito parcial para obtenção do grau de Mestre em Informática.

Orientador: Prof. Dr. Eduardo Luzeiro Feitosa

Manaus
Setembro de 2015

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

da Silva, Adeilson Souza
D111d Detectando Comportamento Automatizado nos Tópicos de Tendência do Twitter no Brasil / Adeilson Souza da Silva. 2015
71 f.: il.; 31 cm.

Orientador: Eduardo Luzeiro Feitosa
Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.

1. Tweets. 2. Tópicos de Tendência. 3. Entropia. 4. Aprendizagem de Máquina. I. Feitosa, Eduardo Luzeiro II. Universidade Federal do Amazonas III. Título

FOLHA DE APROVAÇÃO

"Detectando Comportamento Automatizado nos Tópicos de Tendência do Twitter no Brasil"

ADEILSON SOUZA DA SILVA

Diessertação de Mestrado defendida e aprovada pela banca examinadora contituída pelos Professores:

Prof. Eduardo Luzeiro Feitosa - PRESIDENTE

Prof. Eduardo James Pereira Souto - MEMBRO INTERNO

Prof. Daniel Macêdo Batista - MEMBRO EXTERNO

Manaus, 25 de Setembro de 2015

Agradecimentos

Ao meu orientador, Eduardo Luzeiro Feitosa, pelos ensinamentos, conselhos e atenção dedicados no decorrer deste trabalho, por acreditar e proporcionar o prosseguimento da pesquisa.
Ao amigo Kaio Rafael pelo apoio e por incentivar e acreditar na minha capacidade desde a graduação.
Aos meus pais, Areolino Pereira e Raimunda Souza, pelo apoio, por torcerem sempre por mim e por terem me transmitido os bens mais valiosos na vida: a educação e a honestidade. A minha família, irmaos e sobrinhos pelo apoio de sempre.
Ao meu companheiro Jefferson da Cruz pelos momentos de apoio, força e compreensão no decorrer destes anos.
À FAPEAM, pelo apoio financeiro e concessão da bolsa durante o desenvolvimento desta pesquisa.
Ao Icomp/PPGI/UFAM, professores, amigos e todos que participaram contribuindo com suas amizades e troca de conhecimento.
Aos amigos de modo geral que contribuem com momentos de amizade, diversão e cumplicidade.

Gratidão à todos.

Resumo

O crescimento no número de usuários fez com que as redes sociais, especialmente o Twitter, tornassem-se suscetíveis a criação e propagação de postagens automatizadas. No Twitter, a lista de tópicos de tendência representa os assuntos mais comentados em determinada região e pode ser utilizada indevidamente por contas automatizadas. É necessário então entender e estudar a forma como esses usuários se comportam a fim de criar medidas para combatê-los e garantir que os dados publicados possuam credibilidade. Utilizando uma base de dados real coletada dos tópicos de tendência do Twitter no Brasil, no período de dezembro de 2013 a junho de 2014, com 2.853.822 contas e 11.294.861 *tweets*, uma metodologia para detectar comportamento automatizado nos tópicos de tendência do Twitter foi proposta. Para tanto, foram estudadas diversas características de texto e do comportamento dos usuários para identificar atributos capazes de distiguir usuários humanos de usuários automatizados. Também foram propostas seis (6) novas características extraídas do texto dos tweets baseadas no conceito de Entropia. Utilizando esse conjunto de atributos com algoritmos de aprendizagem de máquina supervisionada para classificação, foi possível detectar 92% das contas automatizadas na base de dados utilizada e, assim, obter uma visão do comportamento desses usuários.

Palavras-chave: Tweets, Tópicos de Tendência, Entropia, Aprendizagem de Máquina

Abstract

The growth in the number of users in social networks, especially Twitter, become themselves susceptible to creation and propagation of automated posts. On Twitter, the Trend Topics list represents the most talked subjects in a particular region and can be misused by automated accounts. Then, it is necessary to understand and study how these users behave in order to create measures to combat them and ensure that published data have credibility. Using a real database collected from the Twitter Trend Topics in Brazil, from December 2013 to June 2014, with 2.853,822 accounts and 11,294,861 *tweets*, a methodology to detect automated behavior in Trend Topics was proposed. For this, we studied several text characteristics and user behavior to identify attributes capable of distiguish human users and automated users. Also were proposed six (6) new features based on the concept of entropy. Using this set of attributes with machine learning algorithms for supervised classification, it was possible to detect 92 % of automated accounts in the database used and thus get an insight into the behavior of these users.

Keywords: Tweets, Trend Topics, Entropy, Machine Learning

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

Uma rede social é uma estrutura que inter-relaciona empresas e pessoas, permitindo que seus usuários recebam e compartilhem informações sobre os mais diversificados assuntos. Por tratar ou apresentar uma "ligação social" entre pessoas, usuários em redes sociais influenciam outras pessoas e são influenciados por elas, de acordo com as suas preferências e particularidades. Em linhas gerais, a função das redes sociais é promover o relacionamento entre seus usuários, contribuindo com novas formas de comunicação e interação social [1].

As redes sociais possuem certas particularidades como a velocidade de compartilhamento das informações, a quantidade de usuários e a quantidade de informações pessoais disponíveis. Essas características, somadas à credibilidade que os usuários possuem em relação aos seus amigos, impulsionam também o crescimento de usuários mal intencionados. Prova disso é que spam, fraudes online, phishing e disseminação de malware tornaram-se cada vez mais comuns em redes sociais.

O problema se manifesta de diversas maneiras como, por exemplo, a inclusão de vídeos contendo spam [2] e a inclusão de metadados que não descrevem adequadamente o conteúdo associado [3]. Em redes sociais como Facebook, Twitter e Pinterest, 40% dos perfis criados tem a intenção de propagar golpes por meio de links compartilhados, e 8% das mensagens postadas em suas páginas são spam, tornando esta prática muito lucrativa. Em 2012, o Facebook informou que 4% do conteúdo gerado por seus usuários apresentava algum tipo de spam, enquanto o Twitter afirmou que apenas 1,5% dos tweets continham spam [4].

Falando mais especificamente do Twitter, Chu et al [5] estimam que 50% das contas no Twitter estão ou são associadas a *bots*, mas o Twitter afirma que esse número é de apenas 5% dos seus 215 milhões de usuários ativos [6]. Muito utilizado como canal de controle para hackers manipularem suas *botnets* [7, 8], *phishing* através de URLs encurtadas [9, 10] e para propagação de spam [11] com o intuito de disseminar propagandas, pornografia e código malicioso.

1.1 Motivação

Pode-se dizer que os usuários maliciosos em redes sociais tem por o objetivo promover ataques à segurança e privacidade dos usuários legítimos, visando sempre algum tipo de benefício. Muitas dessas atividades maliciosas em redes sociais podem ser atribuídas aos *bots* sociais, programas automatizados que utilizam contas, para se passar por usuários legítimos, com o intuito de enganar ou influenciar outros usuários. Dentre os propósitos de criação dos *bots* sociais estão adulterar estatísticas, ganhar popularidade e irritar outros usuários [12, 13].

Recentemente, *bots* sociais foram usados para direcionar os resultados de um algoritmo de análise de opinião, para favorecer um candidato político [14]. Embora os *bots* sociais existam dentro de todas as redes sociais, é no Twitter que eles ganham destaque. O número de usuários e a agilidade no compartilhamento de informações fazem do Twitter um potencial e preferencial alvo para usuários mal intencionados.

Para garantir a segurança e a privacidade de seus usuários, o Twitter possui regras e práticas para criação de contas automatizadas, bem como regras e limites de uso do conteúdo na plataforma [15]. Por exemplo, postar *tweets* automatizados repetidamente para os tópicos de tendência é proibido pois pode degradar a experiência e credibilidade para os outros usuários. Também fornece métodos para que os usuários denunciem comportamento suspeito ou conteúdos ofensivos. Estas denúncias são investigadas e caso algo seja comprovado, as contas são suspensas e/ou o acesso pelo endereço IP referente à conta é temporariamente bloqueado.

Contudo, esses métodos de denúncias, especialmente para spam, acabam não sendo muito eficazes, uma vez que os *spammers* precisam apenas criar uma conta diferente para continuar enviando mensagens ou esperar até que seu endereço IP seja desbloqueado. No caso de denúncias envolvendo *tweets* nos Tópicos de Tendência (TT), o resultado é ainda pior, uma vez que o processo de suspensão é lento e os TTs são efêmeros e em muitos casos duram apenas algumas horas [16]. Vale ressaltar que a lista dos Tópicos de Tendência tem sido usada como mecanismo para gerar tráfego e lucro [2].

No que diz respeito à detecção de atividades suspeitas no Twitter, os trabalhos existentes sobre *bots* sociais [2, 17, 18, 19, 20] focam, tipicamente, na detecção de contas de usuários maliciosos e utilizam características das contas dos usuários para tal identificação.

Em comum, todos esses trabalhos visam minimizar e combater *bots* sociais através do estudo de seu comportamento, identificando características e padrões de atividades que possam ser utilizados para desenvolver contramedidas. A fim de contribuir com a detecção de *bots* em redes sociais, esta dissertação apresenta novas características que permitem caracterizar e detectar comportamento

automatizado nos Tópicos de Tendência do Twitter no Brasil utilizando entropia.

1.2 Objetivos

Esta dissertação tem por objetivo propor e avaliar novas características para caracterizar e detectar comportamento automatizado nos Tópicos de Tendência do Twitter no Brasil, aplicando conceitos de entropia, para medir padrões de escrita dos *tweets* postados pelos usuários.

Para alcançar esse objetivo, será necessário:

- Criar uma base de dados com *tweets* e dados das postagens e contas de usuários que comentam sobre os Tópicos de Tendência do Twitter no Brasil;
- Elaborar uma metodologia que englobe a coleta, o processamento, a extração de características e a avaliação de comportamento automatizado nos TTs do Brasil. A metodologia pode ser aplicada em outras redes sociais e outros cenários em que seja possível coletar dados e mensagens de texto dos usuários, desde que ocorram adequações para cada cenário em particular.

1.3 Estrutura do Documento

O restante deste trabalho está organizado da seguinte forma: o Capítulo 2 apresenta alguns conceitos utilizados neste trabalho, tais como *bot* e *bot social*, características observáveis no Twitter e Entropia. Também aborda a conceituação sobre Aprendizagem de Máquina. No Capítulo 3 são discutidos os trabalhos relacionados que focam na detecção de comportamento automático no Twitter. Os trabalhos aqui apresentados servem como base para o entendimento de técnicas aplicadas na solução proposta nesta dissertação. O Capítulo 4 apresenta e discute sobre o conjunto de atributos empregados na detecção de contas automatizadas, incluindo os novos atributos baseados em entropia. O Capítulo 5 apresenta o processo de implementação utilizado para provar a validade dos novos atributos, o que inclui a coleta de dados, rotulagem, extração de características e avaliação automática utilizando aprendizagem de máquina. No Capítulo 6 são apresentadas as análises e resultados que validam a relevância do conjunto de características propostas nesta dissertação. Por fim, no Capítulo 7 são apresentadas as conclusões, as dificuldades encontradas e os trabalhos futuros.

Capítulo 2

Conceitos Básicos

Este capítulo apresenta os principais conceitos básicos necessários para a compreensão dos temas abordados nesta dissertação. Conceitos como *bots* sociais (especialmente no Twitter), entropia e aprendizagem de máquina são descritos.

2.1 Twitter

Lançado em 2006, o Twitter é um serviço online de rede social e *microblogging* que permite aos seus usuários enviar e ler mensagens de texto chamadas de *tweets*. No Twitter, as mensagens devem conter no máximo 140 caracteres, incluindo URLs (que apontam para fotos, vídeos ou Web sites) que fazem referência ao conteúdo do *tweet*. Contudo, por padrão, os *tweets* somente podem conter URLs encurtadas, devido ao tamanho limitado.

Os relacionamentos no Twitter são direcionais, onde um usuário pode ter seguidores e seguir outros usuários. Contudo, essa relação não precisa ser recíproca. Os prática de compartilhar ou republicar *tweets* de um usuário por outros usuários é conhecida como *retweet*. Para isso, basta iniciar a mensagem com "RT@username", onde o símbolo @ faz referência a quem postou originalmente a mensagem. Os *tweets* também podem incluir respostas e menções a outros usuários utilizando o símbolo @ seguido do nome do usuário. Existem ainda as mensagens diretas para comunicação privada entre os usuários [21].

O símbolo #, chamado *hashtag*, é utilizado para marcar palavras chaves em um *tweet*. As palavras chaves marcadas ou não com *hashtags* que são mais comentadas e utilizadas nos *tweets* tornam-se a lista de Tópicos de Tendência (*Trending Topics* ou TT), uma das ferramentas mais populares do Twitter que permite capturar tendências e tópicos em discussão em determinado momento para uma localização geográfica específica [2]. Utilizando esta característica, os usuários podem encontrar rapidamente notícias sobre um assunto em particular

e saber sobre o que a maioria das pessoas comenta.

Desde sua fundação, o Twitter tem sido alvo de vários ataques, como os relatados em [16], alguns dos quais atingiram muitos usuários devido seu alcance no contexto social e cultural. Também é utilizado como canal de controle para hackers manipularem suas *botnets* [7, 8], *phishing* através de URLs encurtadas [9, 10] e para propagação de spam [11].

Devido ao seu crescimento e sua agilidade no compartilhamento de informações, o Twitter tem se tornado alvo para usuários mal intencionados ou bots.

2.2 Bots Sociais

Hoje em dia, tornou-se comum o roubo de informações e o acesso a recursos computacionais através de máquinas infectadas que compõem os sistemas computacionais. De forma geral, as máquinas infectadas utilizam um software chamado de *bot* (da palavra inglesa *robot*) que permite conectá-las a uma infraestrutura de Comando e Controle (C&C), formando assim uma rede maliciosa, conhecida como *botnet*. O conceito de *botnet* está associado ao conjunto de máquinas comprometidas que permitem ao atacante o controle remoto dos recursos computacionais para realizar atividades fraudulentas ou ilícitas [22]. Para controlar as operações de uma *botnet* é necessário uma entidade externa, conhecida como *botmaster*, que coordena as ações realizadas por cada *bot* [23].

Com o crescimento das redes sociais, o conceito de *bots* evoluiu para refletir essa tendência. **Bots sociais** são *bots* que tentam se passar por usuários humanos, interagindo em redes sociais como Facebook, Foursquare, Twitter, entre outras. Alguns contam com recursos adicionais como software de gestão, que os tornam mais reais ao coordenar contas em diferentes redes sociais. Assim, é possível que um *bot* tenha um perfil online mais duradouro e com maior número de amigos e seguidores. Pesquisadores como Boshmaf et al. [24] afirmam que essa categoria de *bots* está sendo projetada com objetivos mais grandiosos como influenciar eleições, atuar no mercado de ações, atacar governos ou até flertar com pessoas ou outros *bots* [13].

2.2.1 Bots Sociais no Twitter

De acordo com as regras do Twitter [15], *bots* sociais são programas automatizados permitidos para compartilhar conteúdo útil (notícias e eventos) e propagandas autorizadas. Comparado com outras redes sociais, os *bots* no ambiente do Twitter são diferentes devido ao tamanho limitado das mensagens. Por isso, normalmente fazem uso de URLs encurtadas para melhorar sua visibilidade e atender a restrição de espaço. Vale lembrar que o Twitter estabelece regras e

melhores práticas para criação de tarefas automatizadas como feeds em blogs ou postagens automáticas.

Contudo, no Twitter também são encontrados *bots* que disseminam informações errôneas ou inverídicas. Por exemplo, já foram identificadas tentativas de criar factoides em movimentos espontâneos e populares [25], publicação de conteúdo malicioso como spam [21] ou phishing [5], ou que simplesmente visavam aumentar o número de seguidores [26].

O Twitter [15] estabelece algumas limitações e regras quanto ao tipo de conteúdo que pode ser publicado e especifica padrões de comportamento que podem levar à suspensão e/ou cancelamento de uma conta, tais como ofensas, spam, identidade falsa, pornografia, entre outros. Dentre elas, pode-se destacar:

- Atualizações compostas, principalmente, por links e não por postagens pessoais;
- Usuário bloqueado por um grande número de usuários;
- Grande número de reclamações de spam contra determinado usuário;
- A publicação de conteúdo duplicado em várias contas ou várias atualizações duplicadas em uma conta;
- A publicação de várias atualizações não relacionadas a um assunto usando #, um assunto do momento ou popular, ou um assunto promovido;
- Criar repetidamente conteúdo falso ou enganoso na tentativa de chamar atenção para uma conta, um serviço ou link;
- Publicar links enganosos (que contenham códigos maliciosas, apropriação de cliques, entre outros);
- Seguir um grande número de usuários em um curto período de tempo;
- Vender/comprar seguidores;
- Usar ou promover sites de terceiros que reivindiquem obter mais seguidores (por exemplo, correntes de seguidores, sites que prometem "mais seguidores rapidamente" ou qualquer outro site que ofereça adicionar automaticamente seguidores na sua conta).

Devido à significância de detectar e suspender *bots* maliciosos, pesquisadores e engenheiros do Twitter tem dedicado tempo e esforço em manter a rede social livre de ataques. Como os *bots* sociais possuem diversos objetivos (*phishing*, propaganda e distribuição de malware, por exemplo), técnicas baseadas na inspeção

humana de dados de treinamento ou na contribuição da comunidade podem e vem sendo bastante utilizadas na sua detecção.

Contudo, como as mensagens no Twitter são curtas e o conteúdo das URLs é localizado em páginas externas, torna-se naturalmente mais difícil aplicar métodos tradicionais de filtragem de conteúdo baseados em mineração de texto para o Twitter [27]. Além disso, de acordo com [28], os atacantes no Twitter também vem evoluindo para fugir das técnicas de detecção existentes.

Para detectar e combater usuários maliciosos no Twitter, novas abordagens procuram identificar conjuntos de características e padrões de comportamento que sejam possíveis extrair dos *tweets* e das contas dos usuários. No caso desta dissertação é proposto um conjunto de 6 novas características baseado no conceito de entropia.

2.3 Atributos

O Twitter disponibiliza, via API *(*Application Programmimg Interface), métodos pré-definidos que permitem obter um conjunto padrão de atributos do usuário na rede e também atributos dos *tweets* [29]. A literatura em torno do assunto [2, 17, 18, 19, 20] normalmente faz distinção entre esse conjunto de atributos, identificando-os como sendo do usuário (comportamento) e dos *tweets* (conteúdo). Grande parte dos trabalhos de detecção de atividades de usuários no Twitter utiliza atributos desses conjuntos para a construção de soluções, de identificação de contas maliciosas, automatizadas, entre outros.

O Capítulo 4 apresenta todas as características utilizadas e implementadas, incluindo as propostas nesta dissertação.

2.4 Entropia

Os fundamentos da Teoria da informação, formalizados por Claude Shannon [30], têm sido estudados extensivamente devido à sua aplicabilidade em diversas áreas com problemas teóricos, bem como em áreas específicas, tais como compressão de dados, transmissão e processamento de sinais, teoria dos ruídos, correção de erros, criptografia de dados e detecção de anomalias de rede. Dentre os princípios da Teoria da Informação, a Entropia apresenta uma importante característica para quantificação da informação contida em uma mensagem.

A Entropia é uma medida que permite calcular a quantidade mínima necessária de informação para representar a fonte de uma mensagem. Portanto, a quantidade de informação contida numa mensagem é a quantidade de entropia necessária para representá-la [31]. Mais formalmente, seja X uma variável aleató-

ria discreta com alfabeto W e P a distribuição de probabilidade sobre o alfabeto W, tem-se uma função de probabilidade de massa $P(x) = Pr\{X = x\}$, para todo $x \in W$, que satisfaz $\sum_{x \in W} P(x) = 1$.

Assim, Entropia é uma medida de incerteza de uma variável aleatória para qualquer distribuição de probabilidade, definida pela equação [31]:

$$H(X) = -\sum_{x=1}^{n} p(x_i) \log_b p(x_i) \tag{2.1}$$

A Entropia de uma dada sequência de símbolos constitui um limite inferior para o número médio de bits requeridos para codificar os símbolos. No caso em que a sequência de símbolos é um texto, a Entropia pode ser calculada tendo como base esse princípio [32]. Para quantificar uma mensagem, é necessário calcular a Entropia para uma dada frequência de símbolos de uma dada mensagem, ou seja, no caso desta dissertação, dado um conjunto de *tweets* postados por um usuário, calcula-se a Entropia para esse dado conjunto de *tweets* desse usuário.

Nesta dissertação, a Entropia será utilizada com o objetivo de medir o quanto de informação os *tweets* de cada usuário representam, ou seja, calcular uma medida que represente o vocabulário de cada usuário, levando em consideração todos os *tweets* que ele postou na base de dados utilizada. Para quantificar um *tweet* é necessário obter a sequência de símbolos ou caracteres, imprimíveis ou não, utilizados no *tweet* e calcular a quantidade de vezes que cada caractere foi utilizado. No caso em que a sequência de símbolos é um texto, a Entropia pode ser calculada tendo como base o conceito que diz que a quantidade de informação contida numa mensagem é a quantidade de Entropia necessária para representá-la.

Para esta dissertação, dado um conjunto de *tweets* postados por um usuário, calcula-se a Entropia para o conjunto de *tweets* do usuário. O objetivo é medir o quanto de informação os *tweets* de cada usuário representam, ou seja, baseado no conceito de Entropia calcular uma medida que represente o conjunto de *tweets* de cada usuário, levando em consideração todos os tweets que ele postou na base de dados utilizada. Dado um conjunto de *tweets* postados por um usuário, este conjunto é convertido em uma lista com todos símbolos (palavras e caracteres) que o usuário utilizou em seus *tweets*. Então para a lista de símbolos outra lista correspondente de números decimais que representam cada símbolo de acordo com o código ASCII é calculado. Os símbolos que um usuário utilizou, podem ser entendidos como o vocabulário que representam os *tweets* do usuário. Em seguida um dicionário com a contagem da frequência com que cada símbolo ocorre no vocabulário é gerado, onde as chaves do dicionário são a representação decimal de cada símbolo e os valores são a contagem da frequência para o símbolo correspondente, ou seja, cada item do dicionário representa quantas vezes cada símbolo aparece no vocabulário de um usuário.

A Entropia que representa o vocabulário de cada usuário é definida como a soma da probabilidade de cada item multiplicado pelo log da probabilidade do mesmo item:

$$Entropia = \sum item \log item \tag{2.2}$$

O Capítulo 4 detalhará os seis (6) novos atributos propostos que fazem uso de Entropia. Para testar a relevância do conjunto de atributos propostos, os atributos são submetidos a um processo de avaliação automática que utilizada aprendizagem de máquina supervisionada com algoritmos de classificação.

2.5 Aprendizagem de Máquina

Aprendizagem de Máquina é uma área da Inteligência Artificial cujo objetivo é o desenvolvimento de técnicas computacionais para a construção de sistemas capazes de adquirir conhecimento de forma automática. De acordo com Alpaydin [33], um sistema de aprendizagem é composto por programas de computador (algoritmos) utilizados para otimizar um critério de desempenho, usando dados de exemplo ou experiência do passado.

Uma das formas de Aprendizagem de Máquina mais utilizada é a Aprendizagem Supervisionada, onde o algoritmo de aprendizagem recebe um conjunto de amostras para treinamento das quais os rótulos da classe associada são conhecidos. Cada amostra (instância) é descrita por um vetor de atributos e pelo rótulo da classe associada. O objetivo é classificar corretamente cada amostra. A tarefa de classificação consiste em associar objetos de um universo a duas ou mais classes.

Em geral, um problema de classificação pode ser caracterizado por um conjunto de treinamento, que contém objetos rotulados com uma ou mais classes, uma classe de modelos e um método de treinamento [34]. A classe de modelos é uma família parametrizada de classificadores e o método de treinamento seleciona um classificador desta família. Os métodos de treinamento podem ser vistos como algoritmos para adequação de funções, que procuram por um bom conjunto de valores dos parâmetros. Treinando o classificador, pode-se avaliar a adequação dos parâmetros utilizando-se dados de teste, isto é, dados não usados no treinamento [35].

Existem diferentes técnicas propostas para a tarefa de classificação, tais como: árvores de decisão, modelagem da entropia máxima, SVM (*Support Vector Machine*), redes neurais, entre outros. Esta seção irá se concentrar numa breve descrição dos classificadores empregados nos experimentos realizados nesta dissertação. Maiores detalhes sobre o emprego de aprendizagem de máquina em segurança de redes de computadores poderão ser obtidos em Henke et al. [36].

2.5.1 Classificadores

SVM (*Support vector Machine*)

SVM é uma técnica de classificação fundamentada nos princípios da Minimização do Risco Estrutural (*Structural Risk Minimization* - SRM) [37]. Seu objetivo é minimizar o erro no conjunto de treinamento (risco empírico) como também minimizar o erro em relação ao um conjunto de novas amostras (teste) que não foram utilizadas no processo de treinamento (risco de generalização). Em outras palavras, obter um equilíbrio entre risco empírico e risco de generalização, minimizando o excesso de ajustes com respeito às amostras de treinamento, evitando *overfiting*[1] e aumentando a capacidade de generalização.

O algoritmo original de SVM não encontra a solução desejada quando aplicado a dados não linearmente separáveis, característica presente na maioria dos problemas reais [33]. Por isso, a decisão em SVM é expressa em termos de uma função kernel $k(x, x')$ que calcula a similaridade entre dois vetores de características e coeficientes não-negativos $\{\alpha_i\}$ $i^n = 1$, que indicam exemplos de treinamentos que se encontram perto da fronteira de decisão [38]. Maiores detalhes sobre as funções de kernel podem ser obtidos em [36].

Algumas vantagens do SVM são o ótimo desempenho com grandes bases de dados, o processo de classificação rápido e a baixa probabilidade de erros de generalização. Algumas desvantagens são a escolha de uma função kernel adequada e tempo de treinamento longo, que depende da dimensionalidade dos dados.

Árvore de Decisão (J48)

Árvores de decisão são compostas por três elementos básicos: (1) nó raiz, que corresponde ao nó de decisão inicial; (2) arestas, que correspondem às diferentes características; e (3) nó folha, que corresponde a um nó de resposta contendo a classe a qual pertence a amostra a ser classificada. Em árvores de decisão, duas grandes fases devem ser asseguradas. A primeira refere-se à construção da árvore e tem como base o conjunto de dados de treinamento, sendo dependente da complexidade dos dados. Uma vez construída, regras podem ser extraídas através dos diversos caminhos providos pela árvore para que sejam geradas informações sobre o processo de aprendizagem. A segunda refere-se à classificação, pois, para classificar uma nova instância, os atributos são testados pelo nó raiz e pelos nós subsequentes, caso necessário. O resultado deste teste permite que os valores dos atributos da instância dada sejam propagados do nó raiz até um dos nós folhas, ou seja, até que uma classe seja atribuída à instância.

[1] *Overfitting* acontece quando o modelo se especializa nas amostras de treinamento, apresentando uma taxa de acerto baixa para novas amostras.

Vários algoritmos foram desenvolvidos para a construção de árvores de decisão para a tarefa de classificação. O ID3 e o C4.5, desenvolvidos por Quinlan [39], são os mais conhecidos. A principal vantagem do uso de árvores de decisão é a de obter regras que explicam claramente o processo de aprendizagem, podendo ser usadas para uma compreensão mais completa dos dados e dos atributos mais relevantes para um problema de classificação.

Decorate

(Diverse Ensemble Creation by Oppositional Relabeling of Artificial Training Examples)

O DECORATE constrói diretamente várias hipóteses utilizando exemplos de treinamento adicionais construídos artificialmente. É uma técnica de meta-aprendizado geral que usa algum classificador forte para construir um comitê diverso e efetivo de maneira simples e direta [40]. A cada iteração um novo classificador é criado e treinado com dados do conjunto de treinamento original e com dados artificialmente criados - estes são novamente criados a cada iteração, garantindo a diversidade. O novo classificador só não será inserido no comitê caso diminua o desempenho de mesmo. Novos classificadores serão iterativamente treinados e adicionados no comitê até que este alcance o tamanho desejado ou o número de iterações alcance um limite.

Como padrão, o DECORATE utiliza comitês de árvore-de-decisão. Entretanto é possível que outro classificador base seja utilizado também.

RandomForest

O algoritmo RandomForest é um tipo de *ensemble learning*, método que gera muitos classificadores e agrega o seu resultado [40]. O RandomForest gera múltiplas árvores de decisão que depois são utilizadas na classificação de novos objetos.

A técnica de RandomForest pode ser entendida como uma extensão da técnica de árvores de decisão, não no sentido de que seja mais apropriada para conjuntos de dados maiores, mas sim por se tratar de um procedimento que faz uso de métodos de reamostragem (*bootstrapping*, por exemplo) para melhorar a precisão dos modelos construídos.

De acordo com [40], tal técnica consiste, essencialmente, em:

- *Bagging*: gerar, através de reamostragem, um conjunto de subamostras provenientes da amostra original, selecionadas de maneira aleatória simples com reposição;

- *Boosting*: construir, para cada subamostra, um modelo de árvore de decisão, aumentando a ponderação das observações incorretamente classificadas

com base nos modelos criados para as outras subamostras;

- *Randomizing*: desenvolver os modelos de árvore de decisão utilizando, em cada nó, um conjunto de variáveis selecionadas aleatoriamente, diferente para cada nó.

Existem diversas métricas úteis para avaliar o desempenho de classificadores, as que foram utilizadas nesta dissertação são descritas a seguir.

2.5.2 Métricas de Avaliação

Na área de aprendizagem de máquina, adota-se a convenção de rotular os exemplos da classe de maior interesse como positivos (P) e os demais exemplos como negativos (N). Essa convenção é denominada de classificação binária. Em problemas dessa natureza, são comuns as seguintes medidas de interesse:

- VP (**Verdadeiros Positivos**): quantidade de exemplos positivos classificados corretamente;
- FN (**Falsos Negativos**): quantidade de exemplos positivos classificados erroneamente como negativos;
- FP (**Falsos Positivos**): quantidade de exemplos negativos classificados erroneamente como positivos; e
- VN (**Verdadeiros Negativos**): quantidade de exemplos negativos classificados corretamente.

No que diz respeito a medidas de desempenho, cujo objetivo é estimar a precisão do classificador e orientar a escolha de um classificador para o tipo de problema e condição testada [41], pode-se realizar avaliações usando taxas de erro e acerto e certos indicadores. As taxas de erros e acertos empregados nessa dissertação são:

- **Taxa de Verdadeiros Positivos -** VP_r: é a proporção de exemplos classificados corretamente como positivos em relação ao total real de casos positivos. Esta taxa também é conhecida como **Sensibilidade** ou *Recall*, que mede a capacidade do classificador atribuir corretamente a classe positiva. Uma taxa de verdadeiros positivos igual a 1 indica que todos os exemplos positivos foram classificados corretamente;
- **Taxa de Falsos Negativos -** FN_r: é a proporção de exemplos classificados erroneamente como negativos em relação ao total real de casos positivos. Esta taxa é complementar a taxa de verdadeiros positivos;

- **Taxa de Verdadeiros Negativos -** VN_r: é a proporção de exemplos classificados corretamente como negativos em relação ao total real de casos negativos. Esta taxa também é conhecida como **Especificidade**, que mede a capacidade do classificador atribuir corretamente a classe negativa. Uma taxa de verdadeiros negativos igual a 1 indica que todos os exemplos negativos foram classificados corretamente;

- **Taxa de Falsos Positivos -** FP_r: é a proporção de exemplos classificados erroneamente como positivos em relação ao total real de casos negativos. Essa taxa é complementar a taxa de verdadeiros negativos;

- **Taxa de erro total -** ET_r: é a proporção de classificações errôneas em relação ao total de exemplos. A **Acurácia Global** de um classificador é medida pelo complemento desta taxa (1 - ET_r);

- **Taxa de Precisão -** PP_r: é a proporção de exemplos positivos classificados corretamente em relação ao total de classificações positivas, ou seja, essa taxa corresponde à probabilidade estimada de um exemplo ser de classe positiva, dado que foi classificado como positivo. Valores altos para essa taxa (próximos de 1), não necessariamente garantem que o algoritmo está classificando bem os exemplos da classe minoritária, pois pode ter classificado vários exemplos positivos como negativos.

Indicadores

Uma vez que a análise isolada das taxas apresentadas não é suficiente, especialmente na presença de dados desbalanceados, algumas medidas combinam mais de uma das taxas previamente apresentadas em um mesmo indicador.

Medida F (ou *F-score* ou *F-measure*) é um indicador que combina a taxa de precisão e a taxa de verdadeiros positivos [42]. O resultado desse indicador está no intervalo [0,1], sendo o melhor resultado Medida F = 1 e o pior resultado é Medida F = 0.
Curva ROC (*Receiver Operating Characteristic Curve*) é uma ferramenta gráfica bastante útil na avaliação e seleção de classificadores, por permitir estudar a relação entre as medidas de sensibilidade e especificidade.

AUC (*Area Under Curve*) ou área abaixo da Curva ROC é uma ferramenta de avaliação de desempenho de algoritmos. A área situada abaixo dessa curva (AUC) é um indicador usualmente adotado como medida de qualidade do classificador. Sokolova [42] apresentam a seguinte simplificação no cálculo desse indicador:

$$AUC = \frac{(Sensibilidade + Especificidade)}{2} \tag{2.3}$$

Capítulo 3

Trabalhos Relacionados

Trabalhos cujo objetivo é estudar atividades suspeitas de usuários no Twitter são normalmente organizados em duas categorias principais: detecção de contas maliciosas e detecção de mensagens maliciosas. A primeira visa relacionar padrões que permitam definir se o usuário da conta é ou não um *spammer*. A segunda categoria tenta relacionar o conteúdo disseminado a atividades maliciosas como, por exemplo, spam ou *phishing*.

A fim de analisar e adquirir conhecimento de outros trabalhos que também analisam e discutem atividades maliciosas e comportamento automatizado no Twitter, este capítulo descreve alguns trabalhos e, posteriormente, discute suas contribuições e pontos positivos e negativos.

Vale ressaltar que embora existam poucos trabalhos relacionados efetivamente a detecção de comportamento automatizado no Twitter, de acordo com Zhang [43], *tweets* relacionados a spam estão, geralmente, associados com contas que possuem um alto grau de automação. Por isso, muitos dos trabalhos discutidos neste Capítulo tratam de spam ou *spammers*.

3.1 Detecção de Contas Maliciosas

De acordo com Amleshwaram e Reddy [44], os trabalhos existentes que envolvem a detecção de atividade maliciosa e/ou automatizada em contas de usuários do Twitter podem ser classificados em duas categorias: (1) os que analisam as propriedades da conta, e (2) os que analisam o domínio.

Na primeira categoria, o foco é verificar as propriedades das contas dos usuários. Número de seguidores, número de URLs nos *tweets* e a distância entre vítimas e *bots* em um grafo social [19] são exemplos de características avaliadas. De forma geral, tais trabalhos utilizam técnicas de aprendizagem de máquina para classificar contas como legítimas ou de *spammers*, de acordo com uma série

de características. A segunda categoria foca na análise dos domínios ou URLs. A ideia é detectar URLs maliciosas através de análise de contas em *honeypots* e no exame das URLs ou domínios postados em *tweets* em comparação com *blacklists* públicas. As seções seguintes apresentam diversos trabalhos nessas categorias.

3.1.1 Propriedade das Contas

O trabalho de Yardi et al. [45] foca na identificação de contas de usuários. Os autores estudaram o comportamento de um pequeno grupo de usuários maliciosos, através do acompanhamento da *hashtag #robotpickuplines* ao longo de todo seu ciclo de vida dentro do Twitter. Eles descobriram que contas de spam não são significativamente mais novas em comparação com contas legítimas. Em média, contas legitimas tem 258 dias enquanto contas de spam 269 dias. Contudo, perceberam que o número total de seguidores e amigos para *spammers* é três vezes maior do que usuários legítimos.

Como resultado, os autores argumentam que os *spammers* respondem com um pouco mais de frequência do que os usuários legítimos. Isso pode ser explicado pelas estratégias para fugir de sistemas de detecção, limitando a quantidade de conteúdo que é enviado a partir de uma determinada conta. Entretanto, o uso de *hashtag* foi significativamente maior entre *spammers*. Isso porque quanto mais *hashtags* em *tweets* de spam, mais esses *tweets* vão aparecer em pesquisas e, portanto, mais as pessoas poderão potencialmente acessá-las.

O trabalho feito por Benevenuto et al. [2] procura resolver a questão de detectar *spammers* no Twitter. Os autores coletaram uma base de dados do Twitter com mais de 54 milhões de usuários, 1.9 bilhões de URLs e quase 1.8 bilhões de *tweets*. Em seguida, criaram uma coleção rotulada com usuários classificados manualmente em *spammers* e não-*spammers*. Depois, estudaram as características de conteúdo dos *tweets* e o comportamento do usuário no Twitter, e aplicaram um método de aprendizagem supervisionado para identificar *spammers*.

Para o processo de classificação, os autores testaram um conjunto de classificadores utilizando os atributos de conteúdo dos *spammers* e contas de usuários para detectar *spammers*. Cada usuário é representado por um vetor de valores, um para cada atributo. O classificador SVM obteve os melhores resultados frente aos outros classificadores. O conhecimento adquirido foi aplicado para classificar novos usuários em duas classes: *spammers* e não-*spammers*. Aproximadamente, 70% de *spammers* e 96% de não-*spammers* foram classificados corretamente. Somente uma pequena fração de não-*spammers* foram classificados erradamente.

Lee et al. [17] propuseram e avaliaram uma abordagem baseada em *honeypot* para detectar comportamento de spam. O trabalho emprega duas abordagem principais: (1) a implantação de contas *honeypots* para coletar perfis de spam fraudulentos; e (2) análise estatística das propriedades destes perfis de spam

para criar classificadores que permitam filtrar *spammers* antigos e novos. A ideia é que ao detectar a atividade suspeita de um usuário (um pedido de amizade não solicitado, por exemplo), o *honeypot* colete evidências do candidato a *spammer*. Então, estatísticas são extraídas das características dos perfis coletados (número de amigos, texto no perfil, idade, entre outros).

A avaliação foi realizada com um conjunto inicial de treinamento contendo perfis legítimos e perfis de *spammers*. Na prática, foram avaliados mais de 60 classificadores diferentes no Weka(Waikato Environment for Knowlwdge Analysis)[1], com valores padrão para todos os parâmetros, utilizando métricas padrões como precisão, *recall*, acurácia, medida F, FP, VP e curva ROC[2]. O classificador com melhores resultados foi o Decorate, que obteve uma acurácia de 88,98%, medida F de 0,888 e uma taxa de FP de 5,7%, em uma coleção de 215.345 perfis, 4.040.415 *tweets*, 51.650.754 de contas sendo seguidas e 65.904.253 de seguidores. Como resultado final, os autores observaram que a similaridade de conteúdo nos *tweets* de cada *spammer* é grande comparada às outras classes porque alguns deles postam quase o mesmo conteúdo ou *tweets* duplicados.

A abordagem realizada por Kruegel et al. [18] também utiliza contas *honeypots* em três grandes redes sociais (facebook, durante um ano, twitter e myspace, durante 11 meses). Assim, criaram 300 perfis *honeypots* em cada rede social para registrar todas as atividades, como solicitações de amizades, mensagens, convites entre outras, de outros usuários. A ideia é investigar como os *spammers* estão usando as redes sociais e identificar características que permitem detectá-los.

Uma vez que a primeira ação de um *spammer* é enviar solicitação de amizade às vítimas, os autores coletaram 4.250 solicitações como conjunto inicial. Em cima delas, classificaram os *bots* de spam que foram detectados, de acordo com diferentes níveis de atividade e estratégias, em quatro categorias:

- *Displayer*, aqueles que não publicam mensagens de spam, apenas mostram conteúdo de spam em sua página de perfil;
- *Bragger*, aqueles que postam mensagens de spam em sua própria página como status pessoal, para que suas vítimas visualizem em seus perfis;
- *Poster*, aqueles *bots* que enviam uma mensagem direta para cada vítima em seu perfil, mas que podem ser visualizadas por outros usuários;
- *Whisperer*, aqueles que enviam mensagens particulares para as suas vítimas endereçadas a um usuário específico.

[1] É uma coleção de algoritmos de aprendizagem de máquinas para tarefas de mineração de dados

[2] Todas essas métricas são explicadas na Seção 2.5.2

Para avaliação, usaram o Weka com o algoritmo Random Forest como classificador. O foco foi dado à detecção de "bragger" e "poster", porque não requerem perfis reais e são detectáveis observando seus perfis. Para o Twitter, escolheram 500 perfis de *spammers* que contataram as contas *honeypots* ou que foram selecionados manualmente, obtendo uma taxa de FP de 2.5% e FN (Falso Negativo) de 3% para o treinamento. Durante um período de três meses coletaram 135.834 perfis e detectaram 15.932 *spammers*.

Já o trabalho de Wang [19] modela o Twitter como um grafo $G = (V,A)$ que consiste de um conjunto V de nós (vértices), representando contas de usuários, e um conjunto A de arcos (arestas direcionadas), que conectam os nós. Cada arco é um par ordenado de nós distintos. Um arco $a = (i, j)$ é direcionado de Vi a Vj, que mostra que o usuário i está seguindo o usuário j. Assim, a ideia do trabalho é, baseado na política de spam do Twitter, utilizar características baseadas em conteúdo e o grafo de relacionamento entre os usuários para facilitar a detecção de *spammers*.

O resultado mostra que não existem muitas contas de spam seguindo grande quantidade de usuários, ao contrário do que se esperava. Além disso, alguns *spammers* têm muitos seguidores. Para características baseadas em conteúdo, a maioria das contas de spam teve uma grande quantidade de *tweets* duplicados. Contudo, o autor percebeu que essa característica (quantidade de *tweets* duplicados), embora importante para detecção de spam, nem sempre apresenta valores contundentes e, também, que a maioria dos *tweets* de spam também contém URLs e *reply* @.

Em outro trabalho, Wang [46] aplica métodos de aprendizagem de máquina para detectar *bots* de spam no Twitter. As características extraídas para detecção de spam incluem três (3) baseadas em grafo (número de amigos, de seguidores e a razão de seguidores por amigos) e três (3) baseadas em conteúdo (número de *tweets* duplicados, número de links HTTP e o número de *replies/mentions*). O trabalho aplica diferentes métodos de classificação em um conjunto de dados que contém 25.847 contas, cerca de 500K de *tweets* e cerca de 49M de relacionamentos. Para avaliar, foram rotuladas 500 contas em spam e não-spam. O resultado mostra que existe cerca de 1% de contas de spam no conjunto de dados.

Wang et al. [47] desenvolveram um framework de detecção de spam social que pode ser utilizado para qualquer rede social. O framework pode ser separado em três componentes principais: (1) *Mapping* e *Assembly* - técnicas de mapeamento são usadas para converter um objeto específico de rede social em um modelo padrão definido pelo framework para objeto (modelo de perfil, modelo de mensagem ou modelo de página Web); (2) *Pre-filtering* - técnicas *fast-path*[3] são utilizadas

[3]De acordo com os autores, foram usadas *blacklists*, *hashing* e correspondência de similaridade

para verificar objetos recebidos contra objetos de spam conhecidos; (3) Classificação - técnicas de aprendizagem de máquina supervisionada são utilizadas para classificar o objeto de entrada e objetos associados.

Para avaliação, usaram dados do Twitter, MySpace e WebbSpamCorpus e definiram três modelos que representam os objetos mais importantes em redes sociais. Foram usados mais de 40 tipos de classificadores de aprendizagem de máquina supervisionada, incluindo Naive Bayes, SVM e LogitBoost. A base de dados foi um conjunto com mais de 900.000 contas de usuários, mais de 2.4 milhões de *tweets* e todas as URLs. Os usuários são representados pelo modelo de perfil, os *tweets* pelo modelo de mensagem e páginas Web associadas com as URLs pelo modelo de página Web. Naive Bayes obteve o melhor desempenho global com medida F e acurácia alcançada de 0,894 e 95,9%, respectivamente.

Yang et al. [28] apresentam a primeira análise empírica profunda de táticas de evasão utilizadas por *spammers* baseada em um conjunto de dados contendo cerca de 500.000 contas de usuários e mais de 14 milhões de *tweets*. Este é o primeiro trabalho a propor características de detecção baseadas em nós vizinhos no Twitter. A ideia é focar em usuários que postam URLs para sites de *phishing* e *malware*. As principais táticas de evasão que foram identificadas e são utilizadas por *spammers* podem ser categorizadas em dois tipos: (1) táticas de evasão de características baseadas em perfil, como número de seguidores e *tweets*; e (2) táticas de evasão de características baseadas em conteúdo, como similaridade entre *tweets* e *tweets* duplicados. Além disso, em outro trabalho [48], os autores apresentam o primeiro estudo de caso aprofundado de análise de relacionamentos sociais internos de contas criminosas. Descobriram que contas criminosas tendem a estar conectadas socialmente formando uma pequena rede, e que nós centrais no grafo social são mais propícios a seguir contas criminosas.

Song et al. [49] propõem um método de filtragem de spam no Twitter. Em vez de características de contas, considera as características de relações entre o remetente e o receptor da mensagem, que são difíceis para os *spammers* manipularem. A ideia é construir grafos dirigidos baseados nas relações seguidores e seguidos no Twitter. Nos grafos, duas características das relações são medidas: distância e conectividade entre os usuários. A distância é o comprimento do caminho mais curto e a conectividade é medida utilizando o corte mínimo e caminho aleatório.

Os autores coletaram 148.371 perfis do Twitter, 267.551 *tweets*, 4.317.161 seguidos e 963.181 seguidores. Foram selecionados aleatoriamente usuários legítimos, utilizando seus IDs, enquanto que as contas de spam foram selecionadas dentre as contas denunciadas ao Twitter. Verificaram manualmente se cada conta é um *spammer* ou não. No total, foram identificadas 308 contas de spam e 10.000 mensagens de spam. Selecionaram aleatoriamente 5.000 mensagens, onde remetentes e receptores são usuários legítimos do conjunto de dados, e então

construíram grafos para cada par de usuário. Em média, os grafos tem 5.000 nós.

Para avaliação, foram selecionados 1.000 usuários legítimos e 300 *spammers* do conjunto de dados e extraídos os 50 *tweets* mais recentes. Os usuários foram classificados utilizando vários classificadores no Weka, com uma precisão de cerca de 99,7% e FP de apenas 0,6%, os classificadores BayesNet e LogitBoost obtiveram o melhor desempenho dentre os classificadores testados. Os autores foram capazes de descobrir que a maioria dos spams vem de usuários a uma distância de mais de três saltos dos receptores. A partir dos resultados, verificaram que a conectividade entre usuários normais é diferente da conectividade entre *spammers* e usuários normais.

No entanto, segundo os autores, este método tem dois problemas. Primeiro, se um usuário normal cria uma nova conta e envia uma mensagem para um amigo antes da nova conta ter um seguidor, a mensagem será filtrada. Isso ocorre porque as características da nova conta são as mesmas que um *spammer*, já que quando a nova conta é criada ela não estabeleceu qualquer relação ainda. Isso, no entanto, é um problema temporal, porque a nova conta vai ter seguidores em breve. O segundo problema é que o sistema irá identificar as mensagens como normal, mesmo que as mensagens venham de amigos que possuem a conta roubada. Às vezes, os atacantes enviam spam através de contas de usuários normais usando *Cross-Site Request Forgery* (CSRF) [4] ou roubo de senha. Além disso, muitos aplicativos maliciosos usam truques para conseguir uma permissão de escrita dos usuários normais.

Chu et al. [5] realizam uma série de medições para caracterizar as diferenças entre humanos (usuário legítimo), *bot* (programa automático) e *cyborg* (*bot* auxiliado por humano ou humano auxiliado por *bot*), em termos de comportamento de postagem, conteúdo do *tweet* e propriedade da conta. Assim, propuseram um sistema automatizado de classificação que consiste em quatro (4) componentes principais:

- **Componente de entropia**, que usa o intervalo de postagens como uma medida de complexidade de comportamento e detecta a temporização periódica e regular, que é um indicador de automação;
- **Componente de aprendizagem de máquina**, que utiliza conteúdo do *tweet* para verificar se os padrões de texto contêm spam ou não;
- **Componente de propriedades das contas**, que emprega propriedades úteis de conta como dispositivo de postagem utilizado e o número de URLs compartilhadas para detectar desvios da normalidade;

[4] CSRF é uma classe de ataques que explora a relação de confiança entre um aplicativo Web e um usuário legítimo.

- **Tomador de decisão**, baseado no *Linear Discriminant Analysis* (LDA), que usa a combinação linear das características geradas pelos três componentes para categorizar um usuário desconhecido como humano, *bot* ou *cyborg*.

Para avaliar o sistema, os autores coletaram mais de 500.000 usuários do Twitter, incluindo mais de 40 milhões de *tweets* postados por eles. Para desenvolver o sistema de classificação automática, treinaram um conjunto de dados que contém amostras de humanos, *bot* e *cyborg* rotuladas manualmente através da verificação de *logs* dos usuários e página de perfil. O conjunto de treino inclui no total 3.000 amostras, sendo 1.000 de cada classe. O conjunto de testes é criado de forma similar com 3.000 amostras. Os dois conjuntos contém 8.350.095 *tweets*, a partir dos quais é possível extrair características úteis para a classificação como comportamento de postagem e padrões do texto.

Como resultado, os autores perceberam que: (1) para a categoria humano, 5,1% de usuários humanos foram classificados como *cyborg* por engano; (2) para a categoria *bot*, 6,3% dos *bots* são categorizados erroneamente como *cyborg*; e (3) para a categoria *cyborg*, 9,8% de *cyborgs* são classificados como humanos e 7,4% como *bot*. No geral o sistema pode diferenciar com precisão humanos de *bot*. No entanto, é muito mais difícil distinguir *cyborg* de humano ou *bot*.

3.1.2 Propriedades do Domínio e URLs

Thomas et al. [50] desenvolveram Monarch, um sistema em tempo real para identificar URLs de spam em mensagens no Twitter. A ideia é rastrear URLs, como elas são submetidas, e determinar se elas redirecionam para spam. A arquitetura do sistema consiste de três elementos chave: (1) um *front-end*, que aceita URLs fornecidas pelos serviços Web em busca de uma decisão de classificação; (2) um conjunto de navegadores hospedados em infraestrutura de nuvem, que visita URLs para extrair características marcantes; e (3) um mecanismo de classificação distribuído projetado para escalar características que retornam rapidamente uma decisão para saber se uma URL leva a conteúdo de spam.

A classificação baseia-se em características de spam, incluindo características extraídas das propriedades léxicas de URLs, infraestrutura de hospedagem e conteúdo de páginas (HTML e URL). Novas características também foram coletadas, como conteúdo de cabeçalho HTTP, frames de páginas, conteúdo carregado dinamicamente, comportamento de páginas como eventos Javascript, uso de plugin e comportamento de redirecionamento de uma página. A coleção de características e classificação de URLs ocorre no momento em que a URL é submetida ao serviço Web. Também demonstram que uma implementação do Monarch pode alcançar o rendimento de 638.000 URLs classificadas por dia, com uma acurácia global de

91% com 0,87% de FP.

Amleshwaram e Reddy [44] utilizaram uma abordagem baseada em aprendizagem supervisionada e propuseram novas características extraídas do comportamento de postagem do usuário, das URLs, do conteúdo e da página de perfil. A base de dados empregada contou com dois conjuntos de dados coletados do Twitter: Conjunto A, com aproximadamente 500K de usuários, 14M de *tweets* e 6M de URLs (como utilizado em [28]), e Conjunto B, com 110.789 contas de usuários, 2.27M de *tweets* e 263K de URLs.

Para a avaliação, os autores utilizaram 2467 contas de spam e 4854 contas de usuários legítimos, treinando quatro algoritmos de aprendizagem de máquina (Árvore de Decisão, *Random Forest*, *Bayes Network* e Decorate) usando a ferramenta Weka. Utilizando apenas as novas características propostas, a taxa de TP foi cerca de 15% para todos os classificadores testados o melhor desempenho foi apresentado pelo Decorate. As outras características alcançaram 93,6% de *spammers* com uma taxa de FP de 1,8%. Para identificar novos *spammers*, os autores testaram um conjunto de dados de 31.308 usuários compostos de contas benignas e contas de spam suspensas pelo Twitter. Utilizando o modelo aprendido, identificaram 2.378 (7%) contas como *spammers* comparando com o Twitter, que havia suspendido apenas 21,4% das contas (2.3K).

Lee e Kim [51] propõem um sistema de detecção de URLs suspeitas para o Twitter chamado WARNING-BIRD. Em vez de investigar as páginas de destino de URLs individuais em cada *tweet*, consideraram cadeias correlatas de URLs redirecionadas e incluídas em um número de *tweets*. Devido ao fato dos recursos dos atacantes serem limitados e precisarem ser reutilizados, uma parte de suas cadeias devem ser compartilhadas. Os autores encontraram um número de características de URLs suspeitas derivadas de cadeias de URL redirecionadas e informação de contexto dos *tweets* relacionados.

Para avaliar e comparar as características, utilizaram medida F para representar o grau de discriminação de uma característica. Baseado nesta métrica, foi identificado que o comprimento das cadeias redirecionadas é a característica mais importante, seguida pelo número de fontes diferentes e o desvio padrão da data de criação das contas. Já a característica menos importante é o número de contas que postaram uma URL. Os resultados demonstram que atacantes usam um grande número de contas diferentes para distribuir suas URLs maliciosas. Os autores realizaram uma análise diária no conjunto de treinamento e em média, 3.756,38 pontos de entrada para URLs aparecem mais que uma vez em cada "janela de *tweet*" (10.000 *tweets*) durante um dado dia. Em média, 282,93 URLs suspeitas foram detectadas, onde 19,53 URLs são falso positivo e 30,15 URLs são recém descobertas.

3.2 Detecção de Mensagens Maliciosas

Como mencionado na seção anterior, os trabalhos que focam na detecção de contas maliciosas no Twitter dependem de características das contas, incluindo o número de seguidores e amigos, taxa de URLs, data de criação da conta, entre outras. Contudo, a maioria destas características podem ser manipuladas por *spammers*. Assim, a detecção de *tweets* de spam isoladamente e sem informação prévia do usuário tem surgido como uma nova forma de identificar spam, baseado na aplicação de PLN(Processamento de Linguagem Natural), para extrair características [5]. Embora, existam poucos trabalhos, alguns podem ser aplicados ao contexto do Twitter.

O trabalho de Mishne et al. [52] utiliza uma abordagem baseada em modelo de linguagem para detectar URLs de spam em blogs e páginas similares. A ideia é examinar o uso de modelos de linguagem no post do blog, em comentários relacionados e na página dos comentários, e assim explorar as divergências, utilizando KLD (*Kulback-Leibler Divergence*, uma medida de diferença entre duas distribuições de probabilidade), nos modelos de linguagem para classificar efetivamente comentários como spam e não-spam. O método pode ser implantado de dois modos: **de forma retrospectiva**, ao inspecionar uma página de blog que já conta com comentários; ou de **maneira online**, para ser usado por software de blog para bloquear *spammers* em tempo real. Vale ressaltar que no caso de comentário de spam, os modelos de linguagem são susceptíveis a serem substancialmente diferentes: o *spammer* está normalmente tentando criar links entre sites que não tem nenhuma relação semântica (um blog pessoal e um site pornográfico, por exemplo).

Para os experimentos, coletaram aleatoriamente 50 postagens em blogs juntamente com 1.024 comentários (spams e não-spams). Classificaram manualmente 332 (32%) como comentários legítimos alguns contendo links para páginas relacionadas e outros que não possuem links. Outros 692 comentários (68%) foram de spam. Os comentários classificados como spam são de diversos tipos, enquanto alguns possuem simples palavras de spam acompanhadas de links para sites de spam, outros empregam linguagens mais sofisticadas. Os resultados com Naive-Bayes, utilizando a probabilidades máxima, classificaram 68% dos comentários como spam. Enquanto este artigo discute comentários em blogs, o problema e a solução são relevantes para outros tipos de comentários de spam, como em wiki.

Benczúr et al. [53] propuseram um método para detectar links nepotistas (aqueles que tentam se favorecer ou favorecer terceiros) para filtragem de spam na Web, utilizando uma abordagem de modelos de linguagem. Este método utiliza uma combinação de link, conteúdo da página e texto âncora de spam. Assim, um link é baixado (coletado) se os modelos de linguagem das fontes e página marcada tem uma grande divergência.

Os autores utilizaram 31 milhões de páginas coletadas do domínio *.de*, com uma amostra aleatória de 1.000 páginas classificadas manualmente. Capturaram links de spam, penalizando *hyperlinks* e calculando valores de *PageRank*, e ao mesmo tempo identificaram conteúdo de spam se ele não possuir uma fonte confiável do mesmo tema. Finalmente, penalizam diretamente âncoras falsas que dão um valor muito elevado em sistemas de recuperação da informação na Web. Mediram a eficiência do método atribuindo a cada página um valor de *NRank* (*Nepotism Rank*). Em média, observaram que páginas legítimas tem significativamente maior rebaixamento em *NRank* comparado a *PageRank*.

Martinez-Romo e Araujo [54, 55] aplicaram uma abordagem de modelo de linguagem, para diferentes fontes de informação extraídas de páginas Web, a fim de proporcionar indicadores de alta qualidade para a detecção de spam Web. A hipótese é que se duas páginas são ligadas (link), então devem ser relacionadas topicamente (que tenham tópicos em comum), mesmo que seja uma relação contextual fraca. Os autores analisaram diferentes fontes de informação de páginas Web que pertencem ao contexto de um link e aplicaram KLD sobre eles para caracterizar a relação entre duas páginas ligadas.

Utilizaram duas coleções públicas de Web spam disponíveis coletadas do domínio web *.uk* entre maio de 2006 e maio de 2007, respectivamente. WEBSPAM-UK2006 inclui 77.9 milhões de páginas e mais de 3 bilhões de links sobre 11.400 domínios. WEBSPAM-UK2007 inclui 105.9 milhões de páginas e mais de 3.7 bilhões de links sobre 114.529 domínios. Estas coleções de referência foram rotuladas como normal, spam ou fronteira. Para a tarefa de classificação, utilizaram árvore de decisão. Quando aplicados aos conjuntos de dados de WEBSPAM-UK2006 e WEBSPAM-UK2007, 89,4% e 54,2% domínios de spam, com medida F de 0,86 e 0,40, respectivamente, foram detectados.

Kantchelian et al. [56] desenvolveram um método para detecção de spam em comentários de blogs utilizando uma métrica denominada complexidade de conteúdo, que descreve a quantidade de redundância associada com uma string. A base para a métrica de complexidade de conteúdo é taxa de compressão de strings. O intuito é responder, de forma normalizada, a seguinte pergunta informal: quanto de informação determinado texto contém? Assim, é criado um conjunto de características utilizando a taxa de compressão.

Primeiramente, normalizaram as mensagens removendo, por exemplo, caracteres repetidos. Depois, agruparam as mensagens em grupos que compartilham o mesmo autor, o mesmo IP remetente, as mesmas URLs incorporadas, entre outros. Finalmente, calculam a métrica de complexidade de conteúdo para todos os grupos de duas ou mais mensagens. O conjunto de dados utilizado são comentários de uma variedade de plataformas de mídias sociais como blogs pessoais, políticos ou empresariais, sites de notícias e de entretenimento, composto

por outros idiomas, além do inglês, como espanhol, português, chinês, japonês ou francês, em um nível significativo. Avaliam a abordagem utilizando o modelo de variável latente, com as características de complexidade de conteúdo associadas com os comentários do conjunto de dados, alcançando uma precisão de 97% utilizando Regressão Logística padrão.

O trabalho realizado por Martinez-Romo e Araujo [16] apresenta uma nova metodologia para detectar *tweet* de spam nos *trending topics* do Twitter. O estudo é baseado na linguagem usada em cada *tweet* para identificar aqueles cujo objetivo é desviar o tráfego de usuários legítimos para sites de spam. Utiliza uma extensão básica da abordagem de modelos de linguagem para analisar a divergência entre os modelos de linguagem de um tópico e cada *tweet* marcado como suspeito com esse tópico. Utiliza também o conteúdo e o título da página direcionada.

Como resultado, o trabalho obteve uma taxa de FP semelhante a outros trabalhos, mas tem a vantagem de poder ser utilizado em um sistema de tempo real, uma vez que necessita apenas da informação dos *tweets* e, portanto, seu custo computacional é baixo. Os autores concluíram que quanto maior a coleção, menor é a medida F e a taxa de FP, sugerindo que o tamanho da coleção pode influenciar os resultados. A classificação correta de não-spam e spam alcançou valores de 89,3% e 93,7%, respectivamente. Apenas 6,3% dos *tweets* de não-spam foram classificados erroneamente como spam. Os resultados obtidos utilizando características baseadas em modelos de linguagem obtiveram desempenho melhor que características baseadas em conteúdo. A união dos dois conjuntos obteve resultados melhores superando os resultados de cada conjunto separadamente.

3.3 Comportamento Automatizado

A categoria de trabalhos que buscam identificar apenas atividade automatizada, preocupam-se em detectar esse comportamento a fim de dar maior credibilidade aos dados que são utilizados para a construção de novas aplicações e serviços baseados no Twitter.

Para determinar se uma conta no Twitter possui comportamento automatizado, Zhang [43] analisa as atualizações de contas de usuários, utilizando apenas a informação do horário de publicação (disponível publicamente) e a fonte de onde o *tweet* foi publicado. Avaliou 106.573 contas distintas, coletadas durante 3 semanas em abril de 2010. Assim, o autor aplicou um teste chi quadrado de Pearson [5], e descobriu, principalmente, que contas automatizadas exibem padrões de intervalos de atualizações distintos que não podem ser observados visualmente.

[5] É um teste não paramétrico de hipóteses que se destina a encontrar um valor da dispersão para duas variáveis nominais, avaliando a associação existente entre variáveis qualitativas.

Também percebeu que uma parcela dessas contas publicam seus *tweets* da própria Web e que palavras relacionadas a spam estão associadas com contas que possuem um alto grau de automação.

Benevenuto [57] aborda o problema de detectar *bots* no Twitter, focando na identificação de comportamento que foge às estratégias de identificação de atividade automática. Utilizando como base o teste de atividade automática aplicado por Zhang [43] em um conjunto de contas suspensas do Twitter que foram identificadas por [58] e também em um conjunto de contas não-suspensas, rotulou uma base com 110.233 usuários. Identificando atributos linguísticos nas postagens dos usuários e padrões de comportamento capazes de diferenciar usuários *bots* e humanos, foi capaz de detectar cerca de 92% dos *bots* na base de dados.

Diferente de trabalhos anteriores, John et al. [59] propõe uma nova abordagem utilizando análise de sentimento nos *tweets*, propondo um conjunto de atributos identificados por meio da ferramenta SentiMetrix's, juntamente com características já utilizadas em outras abordagens como conteúdo do *tweet* e comportamento do usuário para identificar *bots* no Twitter. Demonstram que o número de fatores relacionados a sentimento são a chave para a identificação de *bots*.

3.4 Discussão

Embora o foco da maioria dos trabalhos apresentados neste capítulo seja a detecção da disseminação de spam no Twitter, praticamente todos eles fazem uso de características extraídas a partir: (1) da contas dos usuários, (2) do conteúdo das mensagens (*tweets*) e (3) das propriedades de URLs e domínios.

Ao avaliar os trabalhos que utilizam contas de usuários na detecção, percebe-se que o baixo custo na extração das características e a facilidade na identificação das mesmas são as principais vantagens. Prova disso é o grande número de autores que empregam tal ideia na detecção de spam no Twitter. A principal desvantagem dessa abordagem, e notadamente reconhecida por grande parte dos autores, reside na facilidade que os *spammers* ou gerentes de contas automatizadas tem em manipular algumas características e assim passar ileso nos processos de detecção. Tal fato tem feito com que pesquisadores elaborem novas características, como as extraídas do grafo de relacionamento dos usuários, que são mais difíceis de serem manipuladas. Entretanto, a extração dessas novas características envolve um grande consumo de tempo e recursos computacionais.

Já os trabalhos que analisam o texto das mensagens tem como vantagem o uso de características dificilmente manipuladas por *spammers*. Até mesmo o uso de PLN (Processamento de Linguagem Natural) já foi proposto e implementado para auxiliar a detectar mensagens de spam no Twitter. A principal contribuição é medir o comportamento dos usuários e procurar identificar padrões de escrita,

intervalos de postagens regulares e uso extensivo de termos comuns em todas suas mensagens. O grande e principal desafio, neste caso, é a restrição de espaço (140 caracteres), o que obriga os usuários a empregar um vocabulário limitado em contexto e informação. Assim, ao avaliar os trabalhos nessa categoria, tornou-se claro que quando combinado a outros conjuntos de características (conta do usuário normalmente), o processo de detecção de spam torna-se mais acurado. Isso faz que com essa combinação seja ideal na detecção de comportamentos nos tópicos de tendência em tempo real.

Os trabalhos que procuram identificar URLs maliciosas no Twitter, analisam basicamente características extraídas de propriedades léxicas, infraestrutura de hospedagem e conteúdo de páginas. O objetivo maior é verificar se uma URL leva a algum conteúdo de spam, por exemplo. Dentro deste cenário, o desafio é aprimorar técnicas que sejam capazes de identificar URLs suspeitas em tempo real, uma vez que como as URLs no Twitter são encurtadas, os usuários nem sempre preocupam-se com a origem das mesmas e são facilmente levadas a acessá-las. A combinação de características do conteúdo dos *tweets* também ajuda a melhorar a detecção de URLs maliciosas.

No que tange o uso de aprendizagem de máquina, uma variada gama de algoritmos de classificação supervisionada é empregada nos trabalhos avaliados para detectar spam no Twitter. SVM, Árvore de Decisão, Naive Bayes, Decorate e Random Forest estão entre as opções mais usadas. A detecção é feita utilizando conjuntos de características do usuário, do conteúdo e, ainda, da análise de URLs, ou a combinação entre dois ou mais conjuntos de características. Essa combinação de diferentes conjuntos de características para detecção de spam tem mostrado um melhor desempenho em termos de acurácia, precisão, *recall*, entre outros, quando comparada com a utilização de apenas características do usuário, do conteúdo ou de URLs separadamente [2, 19]

Apesar dos métodos de detecção de atividades maliciosas apresentados mostrarem-se eficientes, não é objetivo destes trabalhos detectar atividade automatizada que não esteja envolvidas em atividades maliciosas, como por exemplo, contas criadas com a intenção de produzir notícias ou opiniões falsas e/ou enviesadas para um tópico ou usuário específico, com o objetivo de alterar os tópicos de tendência. Os trabalhos citados que procuram detectar atividades automatizadas isoladamente analisam o Twitter de forma global, na intenção de contribuir com a detecção de *bots* que adulteram as informações no Twitter.

No caso desta dissertação, a intenção restringe-se a analisar somente o cenário dos tópicos de tendência do Twitter no Brasil, com o objetivo de detectar atividade automatizada e não atividades maliciosas. Mas por quê analisar os tópicos de tendência? Porque já que os tópicos são os assuntos mais comentados, é muito provável que muitos *bots* participem ativamente dos tópicos. Porquê restrito ao

Brasil? A intenção é demonstrar uma análise dos tópicos de tendência no cenário brasileiro e demonstrar como *bots* se comportam nesse cenário, já que não foram encontrados trabalhos específicos nesse sentido.

Para detectar atividade automatizada nesta dissertação são utilizados atributos das contas e do conteúdo dos tweets, alguns já foram utilizados em outros trabalhos. Como contribuição, são propostos novos atributos baseados no conceito de Entropia para medir padrões de escrita dos *tweets* e seu poder discriminativo para a tarefa de identificar atividade automatizada nos tópicos de tendência do Twitter no Brasil.

Capítulo 4

Atributos Extraídos do Twitter

Este Capítulo descreve o conjunto de atributos empregados na detecção de contas automatizadas. Primeiro, são descritos os atributos de conteúdo, aqueles relativos a propriedades do texto dos *tweets* postados pelos usuários, que capturam propriedades específicas relacionadas à forma como os usuários escrevem seus *tweets*. Em seguida, os atributos de comportamento, aqueles que capturam especificamente propriedades do comportamento do usuário em termos de frequência de postagens, interações sociais e influência no Twitter, são apresentados. Por fim, os seis novos atributos baseados em entropia são detalhados e uma discussão sobre sua capacidade de detecção de contas automatizadas em conjunto aos outros atributos é feita.

4.1 Atributos de Conteúdo

Atributos de conteúdo são baseados em propriedades do texto dos *tweets* postados pelos usuários e procuram identificar características que tenham relação com à forma como os usuários escrevem seus *tweets*. Como os usuários postam diversos *tweets*, uma maneira de medir essas características é analisar o número médio da ocorrência de diversos atributos no texto dos *tweets*. Outros atributos levam em consideração apenas o número do total de ocorrências ou ainda a existência ou não do mesmo. A forma como os usuários escrevem seus *tweets* pode ajudar a diferenciar o comportamento de usuários automatizados e usuários humanos. A Tabela 4.1 apresenta os atributos de conteúdo extraídos dos *tweets* empregados nesta dissertação.

O conjunto de atributos de conteúdo dos *tweets* utilizados nesta dissertação foi identificado levando em consideração outros trabalhos [2, 17, 18, 19, 20]. Em linhas gerais, esses trabalhos utilizaram e provaram, com resultados, o poder descritivo desses atributos em medir e diferenciar a intenção de usuários na rede,

Tabela 4.1: Atributos de Conteúdo Extraídos dos *Tweets*

Atributo	**Descrição**
favorite_ count_ tweet	Número de vezes que um *tweet* foi marcado como favorito por outro usuário
favorited_ tweet	Indica se um *tweet* foi marcado como favorito por outro usuário
source_ tweet	Aplicativo ou ferramenta de onde o *tweet* foi publicado
retweeted	Indica se um *tweet* foi "retweetado"
retweet_ count	Número de vezes que o *tweet* foi "retweetado" (contado pela presença de *RT @username* no texto)
diversidade_ lexica	Número de tokens únicos dos *tweets* de um usuário dividido pelo numero total de tokens dos mesmos *tweets*
media_ tweets_ dia	Número total de *tweets* únicos de um usuário dividido pelo número de dias que ele esteve ativo
media_ tweets_ hora	Número total de *tweets* únicos de um usuário dividido pelo número de horas que ele esteve ativo
num_ fontes	Número de fontes diferentes que um usuário publica
num_ palavras	Número total de palavras diferentes usadas por um usuário
media_ palavras_ tweet	Número total de palavras diferentes usadas por um usuário dividido pelo número total de *tweets* de um usuário
media_ urls_ tweet	Número total de URLs compartilhadas por um usuário dividido pelo número total de *tweets* de um usuário
media_ urls_ topico	Número total de URLs compartilhadas por um usuário dividido pelo número total de tópicos em que ele comentou
media_ tweets_ topico	Número total de *tweets* postados por um usuário dividido pelo número total de tópicos em que ele comentou.
media_ palavras_ topico	Número total de palavras diferentes usadas por um usuário dividido pelo número total de tópicos em que ele comentou
media_ hashtags_ topico	Número total de palavras chaves usadas por um usuário dividido pelo número total de tópicos em que ele comentou
media_ hashtags_ tweet	Número total de palavras chaves usadas por um usuário dividido pelo número total de *tweets* de um usuário
media_ mencao_ tweet	Número total de "@username" usadas por um usuário dividido pelo número total de *tweets* de um usuário
media_ mencao_ topico	Número total de "@username" usadas por um usuário dividido pelo número total de tópicos em que ele comentou

bem como na caracterização de mensagem suspeitas ou envolvidas em atividades automatizadas com objetivos maliciosos.

Nesta dissertação, além dos atributos já utilizados em outros trabalho, foram empregados atributos pouco usados (e até mesmo não empregados) na literatura com o propósito de identificar atividade automatizada no Twitter. São eles: $favorited\_tweet$ e $favorite\_count\_tweet$, $retweeted$. A importância destes atributos pode ser justificada pelo fato de que eles medem a forma como um *tweet* é visto por outros usuários na rede. Por exemplo, os atributos $favorited\_tweet$ e $favorite\_count\_tweet$ procuram medir a preferência de outros usuários por determinado *tweet*, pressupondo que mensagens muito favoritadas provavelmente alcançam um maior número de usuários. É importante ressaltar que estes atributos estão disponíveis no próprio Twitter, via API.

4.2 Atributos de Comportamento

Atributos de comportamento são baseados em propriedades do comportamento do usuário em termos de frequência de postagens, interações sociais e influência, que procuram identificar características que tenham relação com à forma como os usuários interagem e sua popularidade no Twitter. Analisar esses atributos pode revelar diferenças em termos de número de usuários que uma conta segue ou o número de usuários que seguem uma conta, ou ainda a preferência dos tweets de determinado usuário. A relevância desses atributos será analisada no processo de avaliação automática.

A Tabela 4.2 apresenta os atributos de comportamento do usuário empregados nesta dissertação.

Os atributos de comportamento do usuário utilizados nesta dissertação foram extraídos e identificados conforme foi visto em outros trabalhos [2, 17, 18, 19, 20], onde foram utilizados, por exemplo, para caracterizar comportamento de *spammers*. Grande parte desses trabalhos procuram detectar comportamentos suspeitos utilizando características que podem revelar a forma como spammers interagem com outros usuários e sua popularidade no Twitter.

Além destes atributos do usuário, alguns novos foram propostos. São eles: $favorites\_count\_user$, $listed\_count$, $default\_profile$ e dif_tweets. Novamente, assim como nos atributos de conteúdo, estes novos atributos estão disponíveis no próprio Twitter, via API.

Espera-se que usuários humanos e usuários com comportamento automatizado possuam características diferentes em relação a participação e interação na rede. Atributos como quantidade de amigos, a quantidade de seguidores e a quantidade de mensagens postadas, juntamente com outros conjuntos de atributos, já demonstraram ser boas medidas para diferenciar o comportamento de usuários

Tabela 4.2: Atributos de Comportamento do Usuário

Atributos	**Descrição**
id_user	Identificador único de um usuário
verified	Indica se uma conta é verificada pelo Twitter, normalmente indivíduos famosos e marcas importantes.
favourites_count_user	Número total de *tweets* marcados como favoritos em uma conta.
listed_count	Número de listas públicas que um usuário é membro.
protected	Indica se um usuário tem os *tweets* protegidos. Em caso afirmativo é impossível ter acesso aos seus *tweets*.
default_profile	Indica se um usuário alterou a imagem padrão do plano de fundo.
num_followers	Total de contas que seguem um usuário.
num_followed	Total de contas que um usuário segue.
rate_followers_followed	Número total de seguidores dividido pelo número total de seguidos.
num_tweets	Total de *tweets* de uma conta.
count_age	Tempo de existência da conta de um usuário até a última postagem no conjunto de dados.
dif_tweets	Diferença entre o número de *tweets* e os *tweets* publicados nos tópicos de tendência.

maliciosos no Twitter.

4.3 Atributos baseados em Entropia

Para quantificar um *tweet* é necessário obter a sequência de símbolos ou caracteres, imprimíveis ou não, e calcular a quantidade de vezes que cada caractere foi utilizado. Em outras palavras, é necessário calcular a Entropia de uma dada sequência de símbolos.

Para esta dissertação, dado um conjunto de *tweets* postados por um usuário, a Entropia é calculada para o conjunto de *tweets* ou para cada *tweet* do usuário. O objetivo é medir o quanto de informação os *tweets* de cada usuário representam, levando em consideração todos os *tweets* que ele postou na base de dados utilizada. Para tanto, seis (6) novos atributos, extraídos das informações coletadas sobre usuários e suas mensagens foram propostos e são empregados nesta dissertação. A Tabela 4.3 apresenta esses atributos.

O atributo *entropia_total* representa a medida geral de entropia para o vocabulário de cada usuário, onde o vocabulário é entendido como todos os símbolos que ele utilizou nos *tweets* coletados e símbolos são todas as palavras e caracteres obtidos de um dado conjunto de *tweets* postados pelo usuário. É importante res-

Tabela 4.3: Atributos Propostos baseados em Entropia

Atributos	**Descrição**
entropia_total	Entropia total que representa o vocabulário de um usuário
media_entropia_tweets	Entropia média por *tweet* publicado
media_entropia_topico	Entropia média por tópico
entropia_usuario_topicos_diferentes	Entropia dos *tweets* de um usuário em tópicos diferentes, levando em consideração apenas as *hashtags* utilizadas
entropia_usuarios_mesmo_topico	Entropia dos *tweets* de usuários diferentes nos 10 tópicos com maior número de *tweets*
entropia_usuarios_topicos_diferentes	Entropia apenas para um *tweet* por tópico

saltar que, no que tange a implementação dessa característica, para cada *tweet* ou conjunto de *tweets*, uma lista com todos símbolos é gerada, bem como uma lista correspondente de números decimais que representam cada símbolo de acordo com o código ASCII.

Já os atributos $media\_entropia\_tweets$ e $media\_entropia\_topico$ representam, respectivamente, o valor do atributo $entropia\_total$ dividido pelo total de *tweets* que o usuário possui no conjunto de dados, e o valor do atributo $entropia\_total$ dividido pelo número de tópicos que o usuário participou no conjunto de dados.

O atributo $entropia\_usuario\_topicos\_diferentes$ calcula, para cada usuário, a entropia do conjunto de *tweets* por tópico, levando em consideração apenas as *hashtags* utilizadas. Assim, é formado o conjunto de palavras chaves para cada usuário. Devido ao tamanho limitado dos *tweets*, muitos *bots* constroem mensagens utilizando apenas *hashtags* em seus *tweets* [57]. Essa prática mostra uma estratégia para alcançar diversos tópicos de uma só vez.

O atributo $entropia\_usuarios\_mesmo\_topico$ calcula a entropia dos *tweets* de cada usuário para os 10 tópicos com maior número de *tweets*. Essa medida representa o quanto cada usuário pode variar o vocabulário participando do mesmo tópico. A intenção é medir o conteúdo dos *tweets* quando os usuários comentam sobre os mesmos assuntos nos tópicos com o maior número de *tweets*, já que são os mais representativos em número de mensagens.

O atributo $entropia\_usuarios\_topicos\_diferentes$ calcula a entropia apenas para um *tweet* por tópico. Essa medida mostra se os usuários escrevem *tweets* únicos ou com textos muito parecidos, independentemente do tópico. Ao escolher aleatoriamente um *tweet* por tópico, a intenção é medir se o usuário na base de dados possui um vocabulário realmente diversificado. A ideia é que para comentar sobre um mesmo tópico, normalmente um usuário vai postar mensagens não tão

diversificadas, já que fazem referência a um mesmo assunto. Enquanto que um *tweet* para cada tópico deveria ser mais diversificado em termos de vocabulário, já que faz referência a assuntos diversos.

Todas as características baseadas em Entropia apresentadas visam quantificar o quanto cada usuário diversificou as palavras e caracteres utilizadas em seus *tweets*. Assim, se o alfabeto de um usuário é bastante diversificado, então a Entropia será elevada. Caso contrário (alfabeto pouco diversificado ou repetitivo), a Entropia será baixa. Espera-se que usuários humanos possuam uma Entropia maior, já que postam mensagens com conteúdo variado enquanto *bots* postam mensagens automáticas com conteúdo menos diversificado.

Os atributos de conteúdo e os atributos de comportamento já demonstraram em diversos trabalhos serem eficientes para a detecção de usuários maliciosos no Twitter, então a fim de comprovar também sua eficiência na detecção de comportamento automatizado são utilizados nesta dissertação com um conjunto de 6 novos atributos baseados em entropia. A medida de entropia que representa o vocabulário dos usuários é relevante na medida em que, juntamente com os outros conjuntos de atributos revela padrões de escrita e frequência de postagens dos usuários. Os novos atributos baseados em entropia juntamente com os atributos de conteúdo fornecem uma medida para diferenciar o conteúdo postado pelos usuários enquanto os atributos de comportamento analisam a forma como os usuários interagem e se relacionam em termos de popularidade e preferência na rede. Os atributos propostos nesta dissertação quando utilizados em conjunto devem ser capazes de revelar características revevantes para diferenciar usuários humanos de usuários automatizados. O desafio é avaliar o quanto eles impactam na detecção de comportamento auotmatizado, ou ainda quais são mais relevantes isoladamente ou em conjunto.

Capítulo 5

Implementação

Este Capítulo descreve o processo de implementação da metodologia utilizada na construção da base de dados que será utilizada no processo de extração do conjunto de características propostas para a detecção de comportamento automatizado. As características propostas também serão necessárias no processo de avaliação automática utilizando aprendizagem de máquina para determinar a relevância de cada grupo de características. A implementação mostra passo a passo as etapas para alcançar os objetivos propostos.

5.1 Implementação

Como forma de melhor ilustrar o processo de implementação, foi utilizada uma simples metodologia, composta pelas seguintes atividades principais: Coleta de dados, Rotulagem, Extração de Características e Avaliação.

A **Coleta de Dados** consiste na obtenção (coleta) da lista dos Tópicos de Tendência do Twitter no Brasil. Obviamente, também são coletadas as informações dos *tweets* que fazem referência a esses Tópicos (data e hora da postagem, por exemplo) e informações referentes aos usuários (contas), como ID e nome dos usuários, que postaram os *tweets*. Esta atividade é realizada através de scripts que utilizam a API do Twitter para coletar os dados nos Tópicos de Tendência do Brasil. Os dados coletados são automaticamente armazenados em um base local.

Após a atividade de coleta, o **Processo de Rotulagem** se inicia com a função de identificar os usuários de acordo com o padrão de comportamento de postagens automatizado ou humano. Para tanto, a rotulagem é realizada através do uso de testes automatizados. O resultado final da rotulagem é utilizado na atividade seguinte de extração, mas, de forma mais prática, também desempenha papel crucial no treino do classificador. Uma rotulagem mal executada

pode comprometer o processo de aprendizagem e tornar a detecção de usuários automatizados falha.

A **Extração de Características** consiste no uso de ferramentas (scripts) para possibilitar a extração de atributos (características) que representam cada usuário rotulado e seus *tweets*. É nesta atividade que, por exemplo, são extraídos, recolhidos ou elaborados atributos (ou conjunto de atributos) para medir padrões de escrita e postagem dos *tweets*. No final da atividade, cada usuário rotulado é representado por um vetor de características que serão utilizadas para o processo de classificação automática.

A última atividade é a de **Avaliação**. Nela, o conjunto de dados final, formado pelo conjunto de usuários rotulados e seus vetores de características, é utilizado pelo classificador para treinar e adquirir conhecimento necessário na detecção de usuário com comportamento automatizado ou humano. Como resultado final, um modelo de classificação permitirá ter usuários como entrada e notificar, na saída, se eles tem comportamento automatizado ou humano.

A Figure 5.1 ilustra a metodologia utilizada.

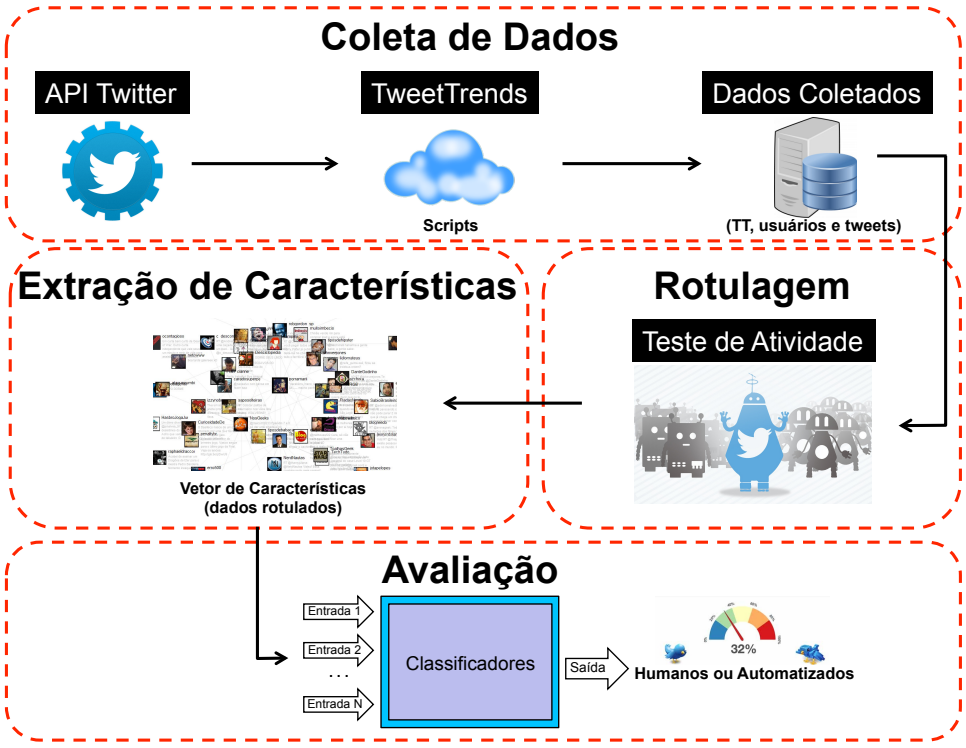

Figura 5.1: Visão geral da Metodologia Utilizada

5.1.1 Coleta de dados

A coleta de dados é realizada via API do Twitter. Para tanto, um Crawler (chamado de "Tweetrends.py") foi desenvolvido na linguagem Python[1].

O Twitter disponibiliza APIs, que utilizam o protocolo HTTP, para fornecer aos seus clientes determinadas funcionalidades descritas por conjuntos de recursos, que podem ser acessados remotamente através de requisições HTTP comuns. Como resposta às requisições, são recebidas coleções de dados estruturados em formato JSON (*JavaScript Object Notation*), que tem o conteúdo em formato de texto codificado com algum esquema de caracteres. A API do Twitter, em sua versão 1.1, fornece acesso a vários métodos e recursos, dentre os quais vale destacar os que foram utilizados nesta dissertação: *Search API*, *Trends*, *Tweets* e *Users*.

A lista completa dos recursos e métodos está disponível em `dev.twitter.com/docs/api/1.1`. No entanto, muitos recursos disponíveis pela API necessitam de autenticação para que sejam acessados por uma aplicação.

5.1.2 Autenticação

A API do Twitter permite a autenticação através de dois mecanismos baseados no padrão OAuth[2]:

1. **Autenticação Application-Only** - A autenticação não fica vinculada a um usuário específico, mas sim a uma aplicação previamente registrada. Quando autenticado com esse mecanismo, a aplicação não poderá realizar algumas operações típicas de um usuário, como postar *tweets*, por exemplo. Esse tipo de autenticação é mais indicado para aplicações que não terão um usuário interagindo com a rede social.

2. **Autenticação de Usuário** - A autenticação se dá diretamente por um usuário, de forma que a aplicação possa realizar operações comuns a usuários. Este tipo de mecanismo é mais indicado para o caso de aplicativos que vão interagir com a rede social pelo usuário.

Nesta dissertação é utilizado somente o tipo 1 de autenticação (*application only*), já que o objetivo é apenas extrair dados.

[1] A escolha de Python se deve pela presença de bibliotecas nativas que realizam chamadas por meio de requisições HTTP, permitindo a interação com a API do Twitter

[2] É um protocolo aberto que permite autenticação segura através de um método simples e padrão para aplicações Web, móveis e desktops [60]

5.1.3 Acessando os recursos de busca da API

Como mencionado anteriormente, os recursos utilizados nesta dissertação foram *Search API*, *Trends*, *Users* e *Tweets*. *Search API* é parte da API do Twitter que permite consultar a lista de tópicos juntamente com seus *tweets*, os usuários que os postaram e dados como hora da postagem, nome do usuário, entre outros.

A lista de tópicos de tendência para determinada região é obtida através do recurso *Trends* (https://dev.twitter.com/rest/reference/get/trends/place). Para utilizar o recurso, basta acessar a URL https://api.twitter.com/1.1/trends/place.json?id=identificador, onde $id = identificador$ é código para o local de interesse. O identificador dever ser do tipo WOEID (*Where On Earth IDentifier*) [3]. O Brasil é representado pelo identificador "23424768".

Após obter a lista de tópicos, é preciso coletar os *tweets* para cada tópico utilizando o recurso *Tweets*. A busca pelos *tweets* para cada tópico é feita passando um valor ao parâmetro *q* na URL https://api.twitter.com/1.1/search/tweets.json?q=tópico1, onde o parâmetro *q* refere-se ao tópico para o qual os *tweets* serão coletados. O conteúdo retornado é uma string no formato JSON, que é decodificado utilizando a função *json.loads* e retornando um dicionário.

A Figura 5.2 representa um pseudo-código em Python para consulta ao recurso *Tweets*, já incluindo o processo de autenticação (linhas com os caracteres »> na frente) e o respectivo dicionário retornado (última linha).

```
>>> from requests_oauthlib import OAuth1Session          # Importação das bibliotecas
>>> import json                                          # Importação das bibliotecas
.....
>>> session = OAuth1Session(API_KEY, API_SECRET, ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
                                                        # Abrindo a seção de autenticação

>>> url = "https://api.twitter.com/1.1/search/tweets.json?q=\%s"   # Acesso ao recurso twetts
>>> url = url \% (requests.utils.quote("topico1"))      # Pesquisa (requisição) sobre o tópico 1

>>> response = session.get(url)                         # Retorno da requisição
>>> tweets = json.loads(response.content)               # Decodificação da resposta em formato JSON
>>> print tweets.keys()                                 # Impressão do Dicionário retornado

[u'search_metadata', u'statuses']                       # Dicionário retornado (resposta)
```

Figura 5.2: Exemplo de Consulta ao Recurso (em Python)

Ainda na Figura 5.2, os *tweets* coletados são apresentados em uma lista dentro do dicionário, na posição de chave *statuses*. Cada *tweet* é um dicionário dentro

[3] https://developer.yahoo.com/geo/geoplanet/guide/concepts.html

dessa lista e para cada dicionário que representa um *tweet* tem-se atributos como:

[u'text', u'user']

O atributo *u'text'*, por exemplo, é o *tweet* que foi publicado. O atributo *u'user'* contém todas as informações do usuário que postou o *tweet*.

5.1.4 Limites no acesso

A fim de evitar excessos, o Twitter impõe limites na quantidade de requisições que uma aplicação pode fazer dentro de uma janela de tempo [61]. Para autenticação *application-only*, que é utilizada nesta dissertação, a janela é de 15 minutos, podendo realizar 15 requisições por janela. Além desse limite, as requisições de busca possuem limite na quantidade de valores que retornam por vez.

Para evitar respostas muito grandes, o serviço do Twitter não retorna mais do que 100 *tweets* como resposta a uma única requisição. Esse valor pode ser alterado através do parâmetro *count* na URL de busca.

Nesta dissertação, o valor máximo por requisição (100 *tweets*) foi escolhido. Isso leva a um total de 15 requisições por janela de 15 minutos. Cada requisição busca a lista com 10 tópicos. Para cada tópico, 100 *tweets* são recuperados. Assim, um total de 15 mil *tweets* são coletados a cada 15 minutos, 60 mil por hora.

5.1.5 Rotulagem

Para o processo de rotulagem, as contas de usuários coletadas são submetidas a um teste de atividade automática, similar ao feito em [43], utilizando o teste Chi Quadrado de Pearson, para verificar se um conjunto de horários de postagens de um usuário é consistente com a distribuição uniforme de minutos por hora esperada para usuários humanos. A regra comum para o teste é que as frequências observadas e esperadas devem ser pelo menos 5, caso contrário o teste é inválido, a exemplo do que já foi proposto anteriormente [43, 57], as contas com menos de 30 *tweets* foram consideradas insuficientes por falharem no teste.

De forma mais direta, a implementação do teste Chi Quadrado utilizada nesta dissertação está disponível na biblioteca Scipy (`http://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.mstats.chisquare.html`), cujo valor retornado é a probabilidade da distribuição observada de horários. Se a probabilidade é muito baixa, a conta avaliada exibe um comportamento não uniforme nos horários em que publica atualizações. Neste caso, possui um comportamento mais parecido com um humano. Por outro lado, se a probabilidade é muito alta, a conta avaliada está usando um mecanismo que faz com que publique os *tweets*

com um nível de uniformidade improvável de ser observada por usuários humanos, o que indica um comportamento automatizado. Para este teste utiliza-se um nível de significância de dois lados de 0.001, ou 0,1%, como limite para falhar no teste [43].

5.1.6 Extração de Características

A extração de características é realizada utilizando scripts python que processam os dados e identificam os atributos coletados, via API, e os atributos calculados como as características baseadas em entropia, média de *tweets* publicados por hora e por tópico, média de palavras e URLs por *tweet* e por tópico. Assim, cada usuário rotulado é representado por um vetor de características que serão utilizadas no processo de classificação.

A extração de características é realizada de forma individual para cada usuário, levando em consideração o ID de cada um no Twitter. Seu funcionamento é o seguinte:

- Com o auxilio de scripts, são coletados os IDs dos usuários do conjunto de dados e, para cada ID, todos os *tweets* postados;
- Os dados então são organizados por ID de usuário, juntamente com o conjunto de *tweets* de cada usuário;
- Todos os atributos que ficam disponíveis via recursos da API pelos métodos *Search API*, *Trends*, *Users* e *Tweets* são extraídos;
- Na sequência, atributos como as médias de *tweets* e de URLs e os atributos baseados em entropia são calculados também de forma individual e única para cada usuário.

No total são extraídos 36 atributos entre os relacionados a comportamento do usuário, conteúdo e baseados em entropia (listados na Tabela 6.13). A saída final é um arquivo que será utilizado no processo de avaliação, onde cada linha do arquivo representa um usuário com todos os seus atributos e a classe a que pertence no final, como é o padrão para a ferramenta Weka.

5.1.7 Avaliação

A fim de avaliar o poder discriminativo dos atributos sugeridos nesta dissertação, os usuários rotulados são submetidos ao processo de avaliação na ferramenta Weka utilizando algoritmos de classificação para medir a capacidade de detectar novos usuários. Estes usuários representam o conjunto de dados final que

será utilizado para o treinamento do algoritmo de classificação. O treinamento é necessário para que o algoritmo obtenha conhecimento suficiente em detecção de comportamento automatizado. Após o algoritmo de classificação adquirir conhecimento suficiente, novos usuários serão recebidos como entrada para que o classificador possa detectá-los e, então, classificá-los em duas classes: automatizado ou humano.

5.2 Protocolo Experimental

5.2.1 Ambiente

Todo o processo de coleta dos dados, rotulagem e extração de características, bem como os experimentos e testes foram executados em duas máquinas. A primeira é uma estação de trabalho Intel Core 7, 3,4 Ghz, com 8 GB de memória RAM, 500 GB de disco, com sistema operacional Linux, distribuição Ubuntu 14.04. A segunda é um servidor composto por dois (2) processadores Intel(R) Xeon(R) CPU E5649 2,53GHz, com 92 GB de memória RAM e 1,8 GB de disco, com sistema operacional Debian.

5.2.2 Conjunto de Dados

A **base de dados inicial** empregada no desenvolvimento dessa dissertação foi formada por um total de 2.853.822 usuários e 11.294.861 *tweets* únicos, coletados no período entre dezembro de 2013 e junho de 2014. Durante esse período de coleta, foram encontrados 2.712 tópicos de tendência distintos (únicos) com cerca de 2.000.000 de URLs.

Dado o grande tamanho da base de dados, foi necessário utilizar algum artifício para reduzir a quantidade de usuários. Ao analisar, manualmente, a base de dados coletados, percebeu-se que, embora muitos usuários ativos possuam grande número de mensagens publicadas no Twitter, esses mesmos usuários publicam poucos *tweets* nos tópicos de tendência. Assim, após análises na literatura, optou-se por empregar o mecanismo implementado em [43] para reduzir a quantidade de usuários. A ideia é que contas com menos de 30 *tweets* publicados durante o período de coleta foram consideradas insuficientes e foram descartadas por falharem no teste de atividade automática.

Por exemplo, o usuário X possui um total de 500 *tweets* (obtido através do atributo *statuses_count*) publicados no Twitter até o último dia de coleta, mas desse total apenas 28 *tweets* foram publicados nos tópicos de tendência. Este usuário é considerado inválido por conter dados insuficientes para investigação de comportamento automatizado nos tópicos de tendência.

Após verificar o número de *tweets* de cada usuário no conjunto de dados, uma nova base de dados foi montada, contendo 50.012 usuários com mais de 30 *tweets* nos tópicos de tendência do Brasil. Em seguida, aplicando o teste de atividade automática aos horários e frequência de postagens por minuto, hora e dia, proposto por [43], foram obtidas 37.919 contas (aprovadas no teste de atividade automática) como pertencentes a usuários humanos e 12.093 contas foram reprovadas (pertencentes a contas de *bots*). Para confirmar e assegurar a validade do teste, os perfis das contas classificadas como automatizadas e seus *tweets* foram consultados utilizando scripts que acessam os perfis e conjunto de tweets dos usuários. No final, a **base de dados final**, para treino e teste, é formada por 50.012 usuários (37.919 humanos e 12.093 automatizados), com 4.352.107 *tweets* e 829.082 URLs, sendo 322.918 URLs únicas.

De posse da base final, foi montada uma base de treinamento com 25.026 amostras rotuladas como Humano e 7.981 amostras rotuladas como Automatizado e uma outra base para teste contendo 12.893 amostras rotuladas como Humano e 4.112 amostras rotuladas como Automatizado. Essa distribuição de valores se refere a 64% das contas para treinamento e 36% para teste, como descrito na literatura de aprendizagem de máquina [33].

Capítulo 6

Avaliação e Resultados

Este capítulo descreve, em três seções, os aspectos essenciais para alcançar o objetivo de identificar contas no Twitter como humanas ou automatizadas nos tópicos de tendência no Brasil. A primeira seção apresenta uma análise passiva de algumas características selecionadas para extração e os mecanismos e ferramentas implementadas para obtenção de seus valores. A segunda seção descreve o treinamento envolvendo aprendizagem de máquina. Em outras palavras, apresenta o processo de ajuste nos classificadores avaliados. Por fim, o teste de validação das novas características é apresentado.

6.1 Análise Empírica

Antes de apresentar os resultados relacionados ao treinamento e testes envolvendo os classificadores e os atributos, incluindo os baseados em Entropia, algumas particularidades e observações a respeito do conjunto de dados obtido são mostradas. Esta seção descreve a análise passiva dos elementos observados na base, tais como a estrutura dos *tweets*, URLs, tópicos e usuários.

6.1.1 Tópicos

Como mencionado nos capítulos iniciais desta dissertação, os usuários do Twitter comentam sobre os mais variados assuntos, que variam de atividades diárias, passam por opiniões diversas sobre eventos e preferências pessoais e chegam até notícias sobre acontecimentos no país. Mas o que será que foi comentado durante os sete (7) meses que a base de dados inicial foi coletada? A Tabela 6.1 destaca os 10 tópicos mais comentados.

Na Tabela 6.1, o tópico *Lovatics* destaca a paixão dos fãs pela cantora Demy Lovato em sua Turnê no Brasil em 2014. Os outros tópicos também destacam a

Tabela 6.1: Os 10 Tópicos mais Comentados

Quant. *tweets*	**Tópicos**	**Ranking**
454.746	Lovatics	1
144.331	Me&MyGirls	2
88.458	KatyCats	3
54.555	#musicfans	4
40.822	#BelieberSegueBelieber	5
39.818	Fly	6
39.595	#AssistaJustTheTwoOfUsP9	7
32.538	#FãSegueFãcomDirectionlizado	8
30.913	#BrazilWantsMoonshineJungleTour	9
30.218	Clara	10

paixão de fãs por artistas que estavam em destaque em 2014, como Katy Perry, Justin Bieber, entre outros. Também fica claro que muitos usuários do Twitter utilizam esse tipo de conteúdo nos tópicos para promover um site, blog, rádio ou programa de TV, ou ainda algum evento particular, que de alguma forma tem relação com música ou promoções de eventos.

Como a base inicial possui mais de 2.700 tópicos de tendência, a Tabela 6.2 apresenta mais outros 10 tópicos, integrantes dos 100 mais comentados, que merecem ser destacados por tratarem de assuntos relevantes ou interessantes.

Tabela 6.2: 10 Tópicos Relevantes entre os 100 mais Comentados.

Quant. *tweets*	**Tópicos**	**Ranking**
29.483	#FatosAssustadores	11
26.934	#PessoasMaravilhosasQueConheciGracasAoTwitter	16
13.833	Brazil	54
13.244	#Cite10Medos	58
13.143	#TerOpiniaoNaoéPreconceitoBial	59
12.756	#AntesDoTwitterEu	61
12.131	#GoogleNãoDeixeOFacebookComprarOTwitter	65
10.270	Carnaval	85
10.187	ENEM	87
8.852	Copa	93

Os tópicos ENEM, Carnaval e Copa tratam sobre eventos únicos que aconteceram durante o período de coleta (final de 2013 até a metade de 2014) no Brasil ou ainda sobre a proximidade desses eventos. O tópico *Brazil* fala sobre fatos relacionados ao Brasil no cenário internacional.

Pode-se perceber facilmente a presença de muitos usuários humanos postando nos tópicos de tendência. A intenção desses *tweets* era criticar ou simplesmente

opinar sobre fatos. Os exemplos são os tópicos $\#TerOpiniaoNaoPreconceitoBial$ e $\#GoogleNoDeixeOFacebookComprarOTwitter$, que expressam críticas dos usuários sobre questões de preconceito nas suas diversas formas e opiniões ou sugestões a respeito da compra do Twitter pela rede social Facebook. Nos demais tópicos ($\#FatosAssustadores$, $\#Cite10Medos$, $\#AntesdoTwitterEu$, $\#PessoasMaravilhosasQueConheciGracasAoTwitter$), os usuários relatam situações particulares falando sobre medos e alegrias em diversas situações.

6.1.2 Hashtags e Tweets

Embora não exista uma regra, é natural pensar que usuários normalmente compartilham uma *hashtag* (tópico) por vez em cada *tweet*, ou seja, comentam sobre o tópico em questão. Em alguns casos, utilizam uma URL para algum conteúdo externo como, por exemplo, fotos e vídeos que tem relação com o assunto comentado.

A Figura 6.1 ilustra alguns exemplos de *tweets* observados na base coletada, cujo padrão de escrita parece ser comum à maioria dos usuários.

#CiteFrasesTipicasDeMãe "'quando chegar em casa a gente conversa"
http://t.co/oTRjM3ZPNr

#Em2013 eles passaram dos limites :@ | http://t.co/AgUiGmGDvy

#10CoisasQueMeEstressa meninas riquinhas que se sentem melhor que eu só por ter mais dinheiro, mas nao sabe nem lavar uma louça

#ArtistaDoSéculo Demi Lovato http://t.co/GA75maCMUo

#Cite11CoisasQueVoceNaoViveSem celular

#GoogleNãoDeixeOFacebookComprarOTwitter apenas orando

#GoogleNãoDeixeOFacebookComprarOTwitter se o google perder, o twitter vai ter agr caixa de notificação, pode crer, scrr mãe

#OdeioCopaDoMundoPorque pq? é só pra ganhar dinheiro, esses fdp fica investindo dinheiro em estádio, áo invés de investir na educação

Figura 6.1: Exemplo do uso de *hashtag* em *tweets* humanos

Pode-se observar no exemplo que os usuários comentam sobre os tópicos de uma maneira natural e lógica, procurando destacar apenas o tópico ao qual estão comentando.

Por outro lado, *tweets* com suspeita de comportamento automatizado são compartilhados em períodos regulares de tempo e usam várias *hashtags* em cada *tweet*. A Figura 6.2 ilustra alguns exemplos de *tweets* que indicam que usuários com comportamento automatizado constroem seus *tweets* seguindo um padrão de escrita repetitivo.

Para melhor ilustrar a relação entre *hashtags* e *tweets*, a Tabela 6.3 demonstra a distribuição de *hashtags* por *tweets*. Vale ressaltar que os valores apresentados

01:40 #TrendsSP #BixaMemoria #PessoasMaravilhosasQueConheciGraçasAoTwitter #FatosAssustadores #AskColton #NemOCéuÉLimitePraFly

02:00 #TrendsSP #NemOCéuÉLimitePraFly #PessoasMaravilhosasQueConheciGraçasAoTwitter #bixamemoria #FatosAssustadores #AskColton

02:20 #TrendsSP #PessoasMaravilhosasQueConheciGraçasAoTwitter #NemOCéuÉLimitePraFly #bixamemoria #FatosAssustadores #AskColton

02:40 #TrendsSP #ParamoreOnJingleBall #PessoasMaravilhosasQueConheciGraçasAoTwitter #NemOCéuÉLimitePraFly #FatosAssustadores #bixamemoria

03:00 #TrendsSP #ParamoreOnJingleBall #PessoasMaravilhosasQueConheciGraçasAoTwitter #NemOCéuÉLimitePraFly #FatosAssustadores #bixamemoria

Figura 6.2: Exemplo do uso de *hashtag* em *tweets* automatizados

foram extraídos, aleatoriamente, de 1.000 contas das classes Humano e Automatizado da base de dados final.

Tabela 6.3: Número de *hashtags* por *tweets*

Hashtags	Humanos	Automatizados
1 - 2	796	87
2 - 3	170	162
3 - 4	23	234
4 - 5	7	319
5 - 6	4	198
Total de Contas	**1000**	**1000**

É possível observar na Tabela 6.3 que cerca de 79% das contas da classe Humano e 10% da classe Automatizado compartilham uma *hashtag* por *tweet*. Para a faixa de duas (2) *hashtags* por *tweet*, as classes Humano e Automatizado tem, respectivamente, cerca de 17% e 16%, o que demonstra um comportamento similar em termos de postagens. Para as faixas de 3 a 6 *hashtags* por *tweet* fica evidente que as contas da classe Automatizado possuem maior ocorrência. Cerca de 73% das contas automatizadas compartilham, em média, de 3 a 6 *hashtags* por *tweet*. Já nas contas humanas, cerca de 30% compartilham entre 3 a 6 *hashtags* por *tweet*. Isso revela que contas automatizadas tendem a postar mais *hashtags* que contas humanas, provavelmente com o intuito de serem mais visualizadas em uma busca por tópicos.

6.1.3 URLs e Tweets

Como também mencionado na Introdução, o uso extensivo de URLs encurtadas pode ser considerado comportamento suspeito no Twitter. A Tabela 6.4 demonstra a relação entre URLs e *tweets*. Assim como na seção anterior, os valores apresentados foram extraídos, aleatoriamente, de 1.000 contas das classes Humano e Automatizado da base de dados final.

Tabela 6.4: Número de *URLs* por *tweets*

URLs	Humanos	Automatizados
0 - 1	531	10
1 - 2	266	206
2 - 3	183	252
3 - 4	20	258
4 - 5	0	166
5 - 6	0	108
Total de Contas	**1000**	**1000**

A Tabela 6.4 mostra que a maioria dos usuários humanos, cerca de 53% fica na faixa com menos de uma URL por *tweet* enquanto que para as faixas de 1 - 2, de 2 - 3 e 3 - 4 URLs apresentam 26%, 18% e 2%. Isso demonstra que os usuários humanos publicam poucas URLs em seus *tweets*. Já as contas automatizadas quase nunca publicam menos do que uma URL (apenas 1% nessa faixa). O foco, cerca de 75%, se concentra nas outras faixas, com valores acima de 20% nas faixas entre 1 e 2, 2 e 3, 3 e 4. Tais valores comprovam que contas automatizadas postam mais *tweets* com URLs e que normalmente apontam para conteúdo extra, como vídeos, imagens e links para outros locais. Contudo, não é possível afirmar que conteúdo malicioso é o objetivo dessas URLs.

6.2 Treinamento

O processo de treinamento é necessário para ajustes de parâmetros individuais ou em conjunto para cada classificador a fim de obter o melhor desempenho de cada classificador sobre o conjunto de dados utilizado nesta fase. O resultado esperado é um modelo capaz de identificar comportamento automatizado utilizando os atributos propostos e, posteriormente, testar em uma nova base de dados para avaliar o desempenho dos classificadores.

Para demonstrar o melhor desempenho de cada classificador no processo de treinamento, observando as métricas de avaliação citadas na seção anterior, utiliza-se uma matriz de confusão. Cada uma das posições na matriz representa

o número percentual de amostras em cada classe original e como elas foram previstas pelo classificador.

6.2.1 Avaliação dos Classificadores

No processo de treinamento de classificadores para esta dissertação foram utilizados 8 classificadores. Contudo, são apresentados somente os que obtiveram os melhores resultados que foram: Decorate, SVM, RandomForest e J48. Todo os classificadores foram testados usando a ferramenta Weka e utilizando o processo de validação cruzada de 10 partições para o treinamento.

6.2.2 SVM

O classificador SVM, além de diferentes funções Kernel, tem como parâmetro mais importante o C, que determina a rigidez do modelo em relação à tolerância a erros. Quanto maior o valor de C mais rígido e preciso será o modelo e mais custoso também. No entanto, quanto menor esse valor mais tolerante a erros e menos rígido.

Para o classificador SVM os resultados obtidos no treinamento foram os menos satisfatórios, conforme pode ser observado na matriz de confusão (Tabela 6.5), onde apenas 43,80% dos usuários automatizados e 59,60% dos humanos foram classificados corretamente no melhor desempenho do classificador.

Tabela 6.5: Matriz de confusão do SVM

		Previsto	
		Humano	Automatizado
Correto	Humano	59,60%	40,40%
	Automatizado	56,20%	43,80%

Mesmo após diversos ajustes e alterações nos parâmetros, o desempenho mostrou-se muito abaixo do que é considerado bom para a tarefa de detectar usuários humanos e automatizados no conjunto de dados utilizado, em conjunto com os atributos propostos. O melhor resultado para o SVM é mostrado na Tabela 6.6, utilizando Kernel Polinomial grau 1.0. Os testes para outros kernel como, por exemplo, RBF (*Radial Basis Function*), não obtiveram melhores resultados e por isso não são mostrados.

O melhor resultado para o SVM tem resultados muito abaixa do do esperado quando comparado a problemas semelhantes em outros trabalhos [57], fazendo então com que ele seja incapaz de distinguir usuários humanos e automatizados

Tabela 6.6: Resultado do SVM

Parâmetro	Taxa de TP	Taxa de FP	Precisão	Revocação	Medida F	Área ROC	Classe
C10	0.596	0.562	0.769	0.596	0.672	0.517	Humano
C10	0.438	0.404	0.257	0.438	0.324	0.517	Automatizado

na base de dados utilizada. Por este motivo, seus resultados servem para fins de demonstração.

6.2.3 J48

Para classificadores baseados em árvore de decisão, neste caso o algoritmo J48 (baseado no algoritmo C4.5), o parâmetro de maior relevância é o **Fator de Confiança** (FC), que determina a poda de nós descendentes até o nó de decisão, de forma a estabelecer a classe das amostras. Funciona de forma similar ao parâmetro C do classificador SVM, ou seja, quanto maior seu valor, mais acertivo e quanto menor, maior as chances de erros de classificação.

O treinamento realizado com o J48 foi melhor quando comparado ao SVM. Houve aumento no número total de contas classificadas corretamente em cada classe. Cerca de 67% de contas automatizadas e 90% de contas humanos foram classificados corretamente, conforme a Tabela 6.7.

Tabela 6.7: Matriz de confusão do J48

J48		Previsto	
		Humano	Automatizado
Correto	Humano	90,30%	9,70%
	Automatizado	32,30%	67,70%

A Tabela 6.8 apresenta os resultados obtidos com o classificador J48.

Tabela 6.8: Resultado do J48

Parâmetro	Taxa de TP	Taxa de FP	Precisão	Revocação	Medida F	Área ROC	Classe
FC 0.50	0.903	0.323	0.898	0.903	0.9	0.782	Humano
FC 0.50	0.677	0.097	0.69	0.677	0.684	0.782	Automatizado

Mesmo havendo um aumento significativo na quantidade de acertos para a classe Humano, os acertos na classe Automatizado representam apenas um pouco mais da metade do total de amostras da classe, o que não é suficiente para o problema de detectar usuários automatizados e humanos. A taxa de TP fica em

torno de 90% para a classe Humano e demonstra uma maior capacidade de acertar corretamente as amostras para esta classe. Para a classe Automatizado, a taxa de TP fica em torno de 67%. Isso demonstra que o classificador aprendeu melhor a classe Humano, com uma taxa de FP 9,7% das amostras classificadas erroneamente como Automatizado, enquanto que a taxa de FN, 32,3% de amostras da classe Automatizado foram classificadas erroneamente como Humano.

6.2.4 Decorate

Observando a matriz de confusão gerada para o Decorate (Tabela 6.9), o melhor caso mostra que para a classe Automatizado, os acertos na classificação ficam em torno de 66% das amostras e os erros ficam em cerca de 33%. Já para a classe Humano, a maioria das amostras foram classificadas corretamente, cerca de 92% de acertos. Contudo, a capacidade do classificador em distinguir usuários humanos e automatizados é baixa, visto que muitos *bots* sociais foram classificados erroneamente como usuários humanos.

Tabela 6.9: Matriz de confusão do Decorate

Decorate		Previsto	
		Humano	Automatizado
Correto	Humano	92,80%	7,20%
	Automatizado	33,20%	66,80%

A Tabela 6.10 apresenta os resultados obtidos com o classificador Decorate. As métricas para avaliar o desempenho do modelo gerado, como a taxa de TP e a taxa de FN, demonstram uma grande probabilidade do classificador acertar somente a classe Humano, que nesta dissertação é a de maior quantidade de amostras.

Tabela 6.10: Resultado do Decorate

Parâmetro	Taxa de TP	Taxa de FP	Precisão	Revocação	Medida F	Área ROC	Classe
R3.0I90	0.928	0.332	0.897	0.928	0.913	0.906	Humano
R3.0I90	0.668	0.072	0.748	0.668	0.705	0.906	Automatizado

6.2.5 RandomForest

Para os classificadores treinados nesta dissertação, o RandomForest foi o que obteve o melhor desempenho, a exemplo do trabalho de [57]. A matriz de confusão mostra o melhor desempenho do RandomForest no processo de treinamento

(Tabela 6.11). É possível observar que a capacidade de identificar corretamente as amostras em cada classe melhorou em relação aos demais classificadores já apresentados. Para a classe Automatizado, o modelo gerado acertou em torno de 92% das amostras no teste realizado para o melhor conjunto de parâmetros escolhidos no processo de treinamento. De igual modo, para a classe Humano, cerca de 94% das amostras foram classificadas corretamente, elevando assim o poder discriminativo do classificador testado.

Tabela 6.11: Matriz de confusão do RandomForest

RandomForest		Previsto	
		Humano	Automatizado
Correto	Humano	94,73%	5,27%
	Automatizado	7,60%	92,40%

A Tabela 6.12 apresenta os resultados obtidos com o classificador RandomForest.

Tabela 6.12: Resultado do RandomForest

Parâmetro	Taxa de TP	Taxa de FP	Precisão	Revocação	Medida F	Área ROC	Classe
I220K40	0.947	0.076	0.975	0.947	0.960	0.937	Humano
I220K40	0.924	0.053	0.848	0.924	0.884	0.937	Automatizado

As métricas de avaliação para o desempenho do RandomForest demonstram que as taxas de TP (0.947) e FP(0.053) para Humano e as taxas de TP (0.924) e FP (0.076) para Automatizado estão mais próximas do que seria ideal para um classificador ótimo. Analisar uma métrica isoladamente, tende a privilegiar uma só classe. A melhor análise deve ser feita de forma conjunta, levando em consideração o melhor desempenho individual para cada classe, ou seja, a capacidade do modelo gerado em distinguir amostras nas classes existentes, diminuindo as taxas de erro.

Assim, após o treinamento realizado com os classificadores e avaliando as métricas de desempenho para comparar o melhor resultado para cada classificador, o RandomForest demonstrou ser o mais adequado para o problema de detectar e distinguir usuários humanos e automatizados na base de dados utilizada nesta dissertação, juntamente com o conjunto de atributos sugeridos.

6.3 Testes

A fim de avaliar o poder discriminativo para detectar comportamento automatizado, o classificador RandomForest que obteve os melhores resultados, foi testado em uma nova base de dados rotulada com 12.893 usuários humanos e 4.112 usuários automatizados.

6.3.1 Relevância dos atributos

A fim de medir a relevância dos atributos utilizados foi calculado o ganho de informação, isto é, o quanto cada atributo é representativo para o problema de distinguir usuários humanos e usuários automatizados.

A Tabela 6.13 apresenta o ranking do ganho de informação dos 36 atributos empregados nesta dissertação. Vale ressaltar que a descrição desses atributos foi apresentada na Seção **??**.

Tabela 6.13: Ranking Ganho de Informação

Num	Atributo	Num	Atributo
1	media_tweets_hora	2	media_tweets_dia
3	media_tweets_topico	4	media_entropia_tweets
5	media_hash_topico	6	entropia_usuario_topicos_diferentes
7	diversidade_lexica	8	media_palavras_tweet
9	media_palavras_topico	10	media_mencao_topico
11	num_palavras	12	media_hash_tweet
13	entropia_usuarios_mesmo_topico	14	media_entropia_topico
15	media_urls_tweet	16	entropia_usuarios_topicos_diferentes
17	media_urls_topico	18	media_menção_tweet
19	dif_tweets	20	statuses_count
21	idade_conta	22	entropia_total
23	listed_count	24	razao_foll_fri
25	followers_count	26	friends_count
27	favourites_count_user	28	retweet_count
29	source_tweet	30	num_fontes
31	favorite_count_tweet	32	verified
33	default_profile	34	protected
35	favorited_tweet	36	retweeted

Os três (3) atributos de maior relevância estão relacionados a média de *tweets* postados por hora (*media_tweets_hora*), por dia (*media_tweets_dia*) e por tópicos (*media_tweets_topico*). Os dois primeiros ajudam a provar que usuários humanos compartilham mensagens de maneira não uniforme tanto por hora quanto no decorrer de um único dia, uma vez que as pessoas comentam sobre um

tópico específico. Por outro lado, as contas automatizadas comentam em diversos tópicos de forma aleatória e mantém uma quantidade uniforme de mensagens postadas por faixa horária. No caso de postagens por tópico (*media_tweets_topico*), os *bots* normalmente participam ativamente em inúmeros tópicos enquanto os usuários humanos comentam de maneira esporádica, as vezes concentrando-se em poucos tópicos.

Os atributos relacionados à forma como os usuários escrevem seus *tweets* também demonstram ter uma grande relevância. Atributos como a média de *hashtags* por tópico (*media_hash_topico*) e por *tweet* (*media_hash_tweet*) demonstram que os *bots* abusam do uso de palavras chaves para marcar seus *tweets* e assim têm maior chances de serem encontrados, por exemplo, em uma busca por assuntos. A ideia é fazer com que um mesmo *tweet* esteja ligado a vários tópicos de uma só vez, alcançando assim um maior número de usuários, ganhando maior visibilidade e, por consequência, aumentando a possibilidade de alcançar mais seguidores. Do mesmo modo, usuários *bots* compartilham muitas URLs em suas mensagens e também fazem menção a outros usuários da rede. O uso extensivo de URLs pode ser uma forma de tentar atrair os usuários mencionados e os seguidores desses usuários a clicar nos links, que em muitos casos direcionam para conteúdo externo como vídeos, imagem ou outros sites.

Os atributos baseados no cálculo da entropia das mensagens demonstram ser eficazes e importantes para distinguir os usuários. Observando que os atributos relativos a média de entropia por *tweet* e por tópico estão entre os TOP 20 do conjunto de atributos, é possível assumir que existe uma alto grau de similaridade entre mensagens de usuários *bots* para tópicos diferentes. Isso fica muito evidente em um busca na base de dados, onde o padrão de escrita destas mensagens normalmente abusa do uso de múltiplas hashtags muitas vezes só alternando a ordem destas na mensagem fazendo com que a mesma mensagem esteja presente em diversos tópicos.

Consequentemente a medida da entropia destas mensagens é praticamente a mesma já que o texto é praticamente o mesmo. No Twitter não é permitido publicar a mesma mensagem mais de uma vez em um curto intervalo de tempo, por isso os *bots* utilizam diversas *hashtags* ou invertem a ordem em que elas aparecem na mensagem para dar a ilusão de que são mensagens diferentes. Também utilizam a lista de tópicos para inserir alguns na mensagem e alternar entre esses tópicos, criando diversas mensagens similares. Uma análise na base de dados permite observar que usuários humanos compartilham mensagens com conteúdo variado.

6.3.2 Análise da Relevância dos Atributos baseados em Entropia

Como forma de provar a importância dos atributos baseados em Entropia, levando em consideração o ganho de informação para os 20 melhores atributos no ranking, quatro (4) atributos foram avaliados usando as 1.000 contas das classes Humano e Automatizado, as mesmas usadas na Seção 6.1.

Entropia por *Tweet*

A Figura 6.3 mostra que usuários humanos apresentam uma Entropia por *tweet* na faixa entre média até alta, enquanto usuários automatizados ficam na faixa entre muito baixa até média. Embora ambos os valores para as duas classes atinjam cerca de 80%, dos *tweets* avaliados, os dados reafirmam que humanos possuem um vocabulário mais diversificado do que os *bots*. Além disso, os *bots* tendem a ser mais repetitivos e menos espôntaneos na escrita dos *tweets*.

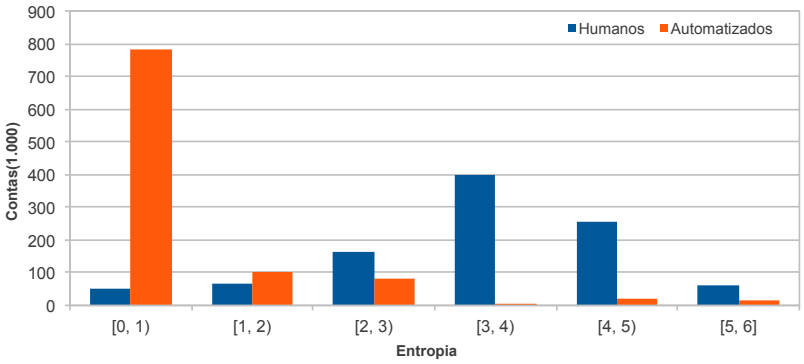

Figura 6.3: Entropia dos *Tweets*

Entropia dos Tópicos

De igual modo, a Figura 6.4 mostra que à Entropia para o total de tópicos comentados é semelhante a Entropia por *tweet*, uma vez que os *tweets* fazem referência aos tópicos.

Entropia nos mesmos Tópicos

Na Figura 6.5 é possível notar dos usuários humanos que comentam em um mesmo tópico, cerca de 87% possuem Entropia considerada entre baixa até média/alta, enquanto usuários automatizados, cerca de 82%, possuem Entropia na faixa entre muito baixa e baixa. Normalmente, usuários humanos que comentam várias vezes em um mesmo tópico não chegam a diversificar tanto o vocabulário,

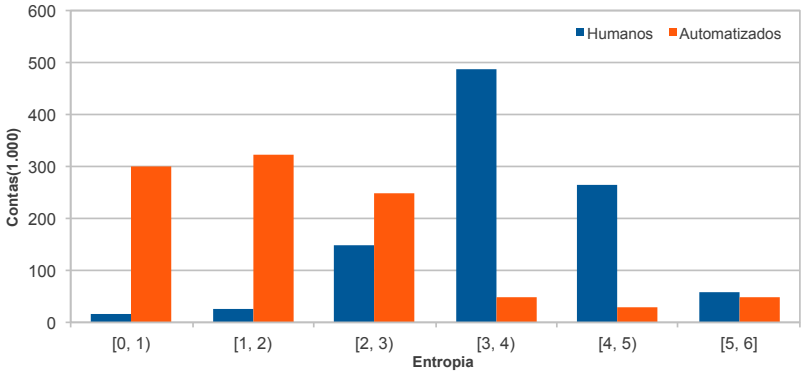

Figura 6.4: Entropia dos Tópicos

já que o assunto de todos seus *tweets* será específico para o tópico em questão. Por outro lado, os usuários automatizados postam *tweets* massivamente para elevar a popularidade de um ou mais tópicos e acabam por repetir *hashtags* e URLs de maneira frequente nos *tweets*.

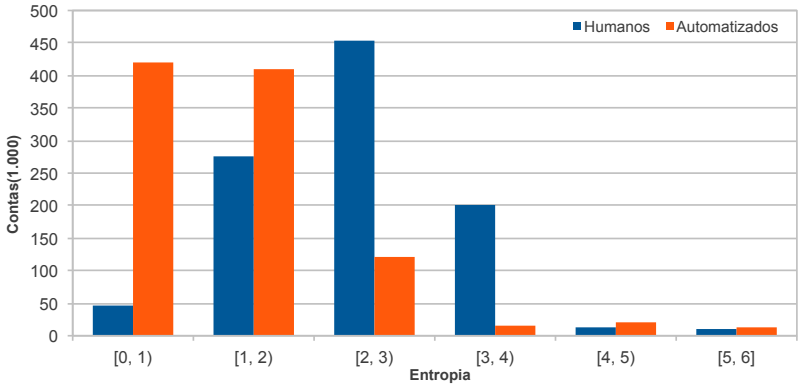

Figura 6.5: Entropia no mesmo Tópico

Entropia em Tópicos Diferentes

Na Figura 6.6, para *tweets* publicados em tópicos diferentes, nota-se que dos usuários humanos, cerca de 94%, possuem Entropia considerada alta ou muito alta, enquanto dos usuários automatizados, cerca de 95%, possuem Entropia considerada muito baixa a média. Isso reflete a maior divergência em termos de vocabulário, uma vez que, mesmo para tópicos diferentes, usuários automatizados publicam *tweets* com pouca diversidade de palavras, enquanto humanos escrevem de maneira natural e espontânea.

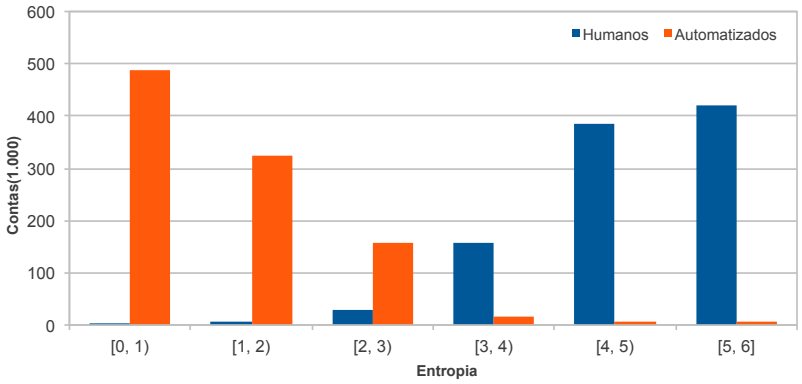

Figura 6.6: Entropia em Tópicos diferentes

6.3.3 Redução do Conjunto de Atributos

A fim de avaliar o poder discriminativo dos três (2) conjuntos de atributos separadamente, foram testados os atributos do conteúdo(C), do usuário (U) e de entropia (E) no classificador RandomForest. A Tabela 6.14 apresenta o desempenho obtido para cada um destes conjuntos ou ainda para mais de um conjunto.

Tabela 6.14: Resultado do RandomForest

Atributos	Humanos	Automatizados
C	89,80%	86,95%
U	79,90%	82,90%
E	86,20%	83,00%
C+U	91,80%	91,70%
C+E	92,80%	92,10%
U+E	91,80%	90,50%
C+U+E	94,73%	92,40%

Os resultados para o conjunto de atributos de conteúdo são os melhores, conseguindo identificar a maior parte dos usuários *bots* em relação aos outros conjuntos de atributos. Em seguida está o conjunto de atributos baseados na entropia do texto das mensagens e por último os atributos do usuário.

Os atributos de conteúdo juntamente com os atributos de entropia representam o melhor resultado, ou seja, revelam que a forma como os usuários escrevem seus *tweets* é a principal característica para distinguir humanos de *bots*. O motivo é simples, esses dois conjuntos de atributos analisam justamente a forma como os *tweets* são escritos e quanto de similaridade possuem entre si e também o quanto a entropia do texto dos *tweets* é representativa para cada usuário.

Os atributos de usuários mostram-se nesta escala com menor poder discriminativo tanto isoladamente quanto em conjunto com outra categoria de atributos. Isso se deve ao fato principalmente de que muitos atributos do usuário não fazem tanta distinção assim entre as classes humanos e *bots*, justamente porquê em nossa base existem muitos *bots* ativos que participam normalmente dos tópicos de tendências postando notícias e atualizações de blogs e sites externos, o que diferencia estes *bots* de usuários comuns é principalmente suas mensagens e a frequência de participação nos tópicos.

Muitos *bots* são criados especificamente para participarem da lista dos tópicos, postando tweets sobre os mais diversos assuntos com o objetivo principal de divulgar um blog, site, evento ou alcançar possíveis seguidores no Twitter.

Capítulo 7

Conclusão

Esta Dissertação apresentou uma metodologia para detecção de comportamento automatizado nos tópicos de tendência do Twitter, através de determinadas características extraídas dos *tweets* e das contas dos usuários, empregando técnicas de aprendizagem de máquina. A importância deste trabalho se deve ao fato de que embora o próprio Twitter proíba a postagem de *tweets* automatizados nos tópicos de tendência, tal prática tem se tornado cada vez mais comum, podendo simplesmente degradar a experiência dos usuários como também expô-los a ataques e atividades maliciosas.

Além da metodologia proposta, outra contribuição deste trabalho é a proposição de seis (6) novas características, baseadas no conceito de Entropia, para a tarefa de detectar comportamento automatizado no Twitter. A utilização do conceito de Entropia para medir e diferenciar o padrão de escrita dos usuários no Twitter é importante porque demonstra a possibilidade de mensurar o vocabulário dos usuários. Análises de usuários com comportamento automatizado através da medida de Entropia calculada para seus *tweets* é baixa, o que demonstra que o vocabulário é pouco diversificado ou repetitivo. Por outro lado, os usuários que possuem comportamento humano apresentam Entropia alta para seus *tweets*, demonstrado que possuem um vocabulário bastante diversificado.

Os resultados mostraram que a junção de atributos extraídos dos usuários e do conteúdo dos *tweets* mais os baseados em Entropia, a fim de avaliar a eficácia da metodologia, permitem detectar comportamento automatizado nos tópicos de tendência do Twitter no Brasil. Essa detecção é possível, principalmente, devido ao padrão de escrita observado nos *tweets* dos usuários que possuem comportamento automatizado.

No processo de avaliação utilizando aprendizagem de máquina, o classificador RandomForest foi o que obteve o maior número de amostras classificadas corretamente, acertando 92,40% dos usuários automatizados e 94,73% dos usuários humanos. Já a fração de usuários automatizados classificados erroneamente como

humanos foi de cerca de 7,60%. Ao inspecionar a base de dados gerada, percebe-se que estes usuários postam menos *hashtags* e postam *tweets* em intervalos de tempo menos regulares, fazendo com que o classificador não consiga acertar sua classe de rótulo.

Usuários que possuem comportamento automatizado abusam da lista de tópicos de tendência e incorporam nos seus *tweets* repetidamente as palavras marcadas com # ou somente as palavras chaves. Acredita-se que para a construção de seus *tweets*, os usuários automatizados criam programas que fazem uso da lista de tópicos e a incorporam automaticamente na escrita dos *tweets*, postando em intervalos de tempo regulares e não interagindo com outros usuários.

É necessário ressaltar que os resultados apresentados nesta dissertação são exclusivos para a base de dados utilizada neste trabalho. Mesmo diante de outras soluções já propostas, o resultado deste trabalho é satisfatório no que diz respeito às taxas de acerto e relevância dos atributos para diferenciar usuários humanos e usuários automatizados. Fica evidente a diferença no intervalo de postagens dos usuários identificados na base de dados. Esta talvez seja a principal diferença juntamente com o padrão de escrita para cada classe de usuário.

7.1 Dificuldades Encontradas

A maior dificuldade para a realização deste trabalho foi encontrar a maneira correta e mais adequada para organizar e estruturar os dados coletados, uma vez que outras soluções já propostas utilizam base de dados próprias e não é permitido compartilhar dados de usuários porque isso viola a privacidade e integridade dos usuários no Twitter.

Além disso, os dados coletados online dependem de conexão à Internet ativa e de estabilidade e disponibilidade da API do Twitter. Em alguns casos, quando a conexão ou a API estão indisponíveis ocorrem erros nas requisições que precisam ser tratadas e corrigidas no momento em que ocorrem ou posteriormente, de forma a não comprometer a estrutura da base de dados.

7.2 Trabalhos Futuros

Atividades automatizadas no Twitter não são necessariamente maliciosas, uma vez que o Twitter permite a criação de ferramentas para atualizações automáticas. A detecção de atividades maliciosas como spam, *phishing*, entre outros, é um novo passo para a tarefa de detecção de atividades automatizadas nos tópicos de tendência.

Já existe na literatura trabalhos que buscam detectar atividades com spam

e *phishing*, no entanto estes trabalhos identificam de forma geral e não especificamente nos tópicos de tendência. Detectar atividades maliciosas nos tópicos de tendência ajuda a combater ataques de forma mais eficaz, uma vez que os tópicos são efêmeros e são atualizados constantemente.

Como trabalho futuro é necessário criar grupos de atributos cada vez mais robustos e que representem com maior relevância comportamento automatizado para que seja possível detectar alguma atividade maliciosa que ocorra nos tópicos de tendência do Twitter no Brasil. Outro ponto a ser considerado como trabalho futuro é criar mecanismos de detecção online de comportamento automatizado e/ou malicioso, isso é um grande desafio uma vez que demanda processamento em tempo real dos dados necessários para investigar possíveis ameaças nos tópicos de tendência do Twitter.

Referências Bibliográficas

[1] M. Naaman, J. Boase, and C.-H. Lai, "Is it really about me?: Message content in social awareness streams," in *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, CSCW '10, (New York, NY, USA), pp. 189–192, ACM, 2010.

[2] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on twitter," in *Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*, July 2010.

[3] F. Benevenuto, T. Rodrigues, V. A. F. Almeida, J. M. Almeida, M. A. Gonçalves, and K. W. Ross, "Video pollution on the web.," *First Monday*, vol. 15, no. 4, 2010.

[4] Geoffrey A. Fowler, "Facebook: One Billion and Counting," 2012. http://www.wsj.com/articles/SB10000872396390443635404578036164027386112.

[5] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Detecting automation of twitter accounts: Are you a human, bot, or cyborg?," *Dependable and Secure Computing, IEEE Transactions on*, vol. 9, pp. 811–824, Nov 2012.

[6] Tom Gara, "One Big Doubt Hanging Over Twitter's IPO: Fake Accounts," 2013. http://www.wsj.com/article_email/SB10001424052702303492504579113754194762812-lMyQjAxMTAzMDAwMzEwNDMyWj.html.

[7] J. Nazario, "Twitter-based Botnet Command Channel," 2009. https://asert.arbornetworks.com/twitter-based-botnet-command-channel/.

[8] E. Kartaltepe, J. Morales, S. Xu, and R. Sandhu, "Social network-based botnet command-and-control: Emerging threats and countermeasures," in *Applied Cryptography and Network Security* (J. Zhou and M. Yung, eds.), vol. 6123 of *Lecture Notes in Computer Science*, pp. 511–528, Springer Berlin Heidelberg, 2010.

[9] Malware Bytes, "Twitter Phishing Spamrun: "Strange Rumors About You'," 2014. https://blog.malwarebytes.org/fraud-scam/2014/09/twitter-phishing-spamrun-strange-rumors-about-you/.

[10] Symantec, "Phishing: The Easy Way to Compromise Twitter Accounts," 2013. http://www.symantec.com/connect/blogs/phishing-easy-way-compromise-twitter-accounts.

[11] Neal Ungerleider, "Almost 10% of Twitter is Spam," 2015. http://www.fastcompany.com/3044485/almost-10-of-twitter-is-spam.

[12] D. Harris, "Can evil data scientists fool us all with the world?s best spam?," 2013. http://gigaom.com/2013/02/28/can-evil-data-scientists-fool-us-all-with-the-worlds-best-spam.

[13] J. Messias, L. Schmidt, R. Oliveira, and F. Benevenuto, "You followed my bot! transforming robots into influential users in twitter," *First Monday*, vol. 18, no. 7, 2013.

[14] M. Orcutt, "Twitter mischief plagues mexico?s election," 2012. http://www.technologyreview.com/news/428286/twitter-mischief-plagues-mexicos-election/.

[15] Twitter, "Denunciar spam no Twitter," 2014. https://support.twitter.com/articles/263349-como-denunciar-por-spam-no-twitter.

[16] J. Martinez-Romo and L. Araujo, "Detecting malicious tweets in trending topics using a statistical analysis of language," *Expert Systems with Applications*, vol. 40, no. 8, pp. 2992 – 3000, 2013.

[17] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: Social honeypots + machine learning," in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, (New York, NY, USA), pp. 435–442, ACM, 2010.

[18] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, (New York, NY, USA), pp. 1–9, ACM, 2010.

[19] A. H. Wang, "Don't follow me: Spam detection in twitter," in *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pp. 1–10, July 2010.

[20] A. Wang, "Machine learning for the detection of spam in twitter networks," in *e-Business and Telecommunications* (M. Obaidat, G. Tsihrintzis, and J. Filipe, eds.), vol. 222 of *Communications in Computer and Information Science*, pp. 319–333, Springer Berlin Heidelberg, 2012.

[21] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?," in *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, (New York, NY, USA), pp. 591–600, ACM, 2010.

[22] B. McCarty, "Botnets: big and bigger," *Security Privacy, IEEE*, vol. 1, pp. 87–90, July 2003.

[23] F. C. Freiling, T. Holz, and G. Wicherski, "Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks," in *Proceedings of the 10th European Conference on Research in Computer Security*, ESORICS'05, (Berlin, Heidelberg), pp. 319–335, Springer-Verlag, 2005.

[24] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu, "The socialbot network: When bots socialize for fame and money," in *Proceedings of the 27th Annual Computer Security Applications Conference*, ACSAC '11, (New York, NY, USA), pp. 93–102, ACM, 2011.

[25] J. Ratkiewicz, M. Conover, M. Meiss, B. Gonçalves, S. Patil, A. Flammini, and F. Menczer, "Truthy: Mapping the spread of astroturf in microblog streams," in *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, (New York, NY, USA), pp. 249–252, ACM, 2011.

[26] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi, "Measuring user influence in twitter: The million follower fallacy," in *4th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010.

[27] G. Mishne, D. Carmel, and R. Lempel, "Blocking blog spam with language model disagreement.," in *AIRWeb*, pp. 1–6, 2005.

[28] C. Yang, R. C. Harkreader, and G. Gu, "Die free or live hard: Empirical evaluation and new design for fighting evolving twitter spammers," in *Proceedings of the 14th International Conference on Recent Advances in Intrusion Detection*, RAID' 11, (Berlin, Heidelberg), pp. 318–337, Springer-Verlag, 2011.

[29] Twitter, "API Overview," 2015. https://dev.twitter.com/overview/api.

[30] C. E. Shannon, "A mathematical theory of communication," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, pp. 3–55, Jan. 2001.

[31] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley-Interscience, 2nd ed., 2006.

[32] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st ed., 2009.

[33] E. Alpaydin, *Introduction to Machine Learning*. The MIT Press, 2nd ed., 2010.

[34] C. Zhai, *Statistical Language Models for Information Retrieval*. Morgan and Claypool Publishers, 2008.

[35] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 ed., 2008.

[36] M. Henke, E. Nunan, C. Santos, E. Souto, E. M. Dos Santos, and E. Feitosa, "Aprendizagem de maquina para seguranca em redes de computadores: Metodos e aplicacoes," in *Livro dos Minicursos do XI Simposio Brasileiro em Seguranca da Informacao e de Sistemas Computacionais* (SBC, ed.), pp. 53–103, SBC, Novembro 2011.

[37] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[38] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious urls," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, (New York, NY, USA), pp. 1245–1254, ACM, 2009.

[39] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[40] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.

[41] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, (San Francisco, CA, USA), pp. 1137–1143, Morgan Kaufmann Publishers Inc., 1995.

[42] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond accuracy, f-score and roc: A family of discriminant measures for performance evaluation," in *Proceedings of the 19th Australian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence*, AI'06, (Berlin, Heidelberg), pp. 1015–1021, Springer-Verlag, 2006.

[43] C. M. Zhang and V. Paxson, "Detecting and analyzing automated activity on twitter," in *Proceedings of the 12th International Conference on Passive and Active Measurement*, PAM'11, (Berlin, Heidelberg), pp. 102–111, Springer-Verlag, 2011.

[44] A. A. Amleshwaram, A. L. N. Reddy, S. Yadav, G. Gu, and C. Yang, "Cats: Characterizing automation of twitter spammers.," in *COMSNETS*, pp. 1–10, IEEE, 2013.

[45] S. Yardi, D. Romero, G. Schoenebeck, and danah boyd, "Detecting spam in a twitter network," *First Monday*, vol. 15, January 2010.

[46] A. H. Wang, "Detecting spam bots in online social networking sites: A machine learning approach," in *Proceedings of the 24th Annual IFIP WG 11.3 Working Conference on Data and Applications Security and Privacy*, DBSec'10, (Berlin, Heidelberg), pp. 335–342, Springer-Verlag, 2010.

[47] D. Wang, D. Irani, and C. Pu, "A social-spam detection framework," in *Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference*, CEAS '11, (New York, NY, USA), pp. 46–54, ACM, 2011.

[48] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu, "Analyzing spammers' social networks for fun and profit: A case study of cyber criminal ecosystem on twitter," in *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, (New York, NY, USA), pp. 71–80, ACM, 2012.

[49] J. Song, S. Lee, and J. Kim, "Spam filtering in twitter using sender-receiver relationship," in *Proceedings of the 14th International Conference on Recent Advances in Intrusion Detection*, RAID'11, (Berlin, Heidelberg), pp. 301–317, Springer-Verlag, 2011.

[50] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time url spam filtering service," in *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, (Washington, DC, USA), pp. 447–462, IEEE Computer Society, 2011.

[51] S. Lee and J. Kim, "Warningbird: A near real-time detection system for suspicious urls in twitter stream," *IEEE Trans. Dependable Secur. Comput.*, vol. 10, pp. 183–195, May 2013.

[52] G. Mishne, "Blocking blog spam with language model disagreement," in *In Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb*, 2005.

[53] A. A. Benczúr, I. Bíró, K. Csalogány, and M. Uher, "Detecting nepotistic links by language model disagreement," in *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, (New York, NY, USA), pp. 939–940, ACM, 2006.

[54] J. Martinez-Romo and L. Araujo, "Web spam identification through language model analysis," in *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb '09, (New York, NY, USA), pp. 21–28, ACM, 2009.

[55] L. Araujo and J. Martinez-Romo, "Web spam detection: New classification features based on qualified link analysis and language models," *Information Forensics and Security, IEEE Transactions on*, vol. 5, pp. 581–590, Sept 2010.

[56] A. Kantchelian, J. Ma, L. Huang, S. Afroz, A. D. Joseph, and J. D. Tygar, "Robust detection of comment spam using entropy rate.," in *AISec* (T. Yu, V. N. Venkatakrishan, and A. Kapadia, eds.), pp. 59–70, ACM, 2012.

[57] C. Freitas, F. Benevenuto, and A. Veloso, "Socialbots: Implicações na segurança e na credibilidade de serviços baseados no twitter," in *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, May 2014.

[58] S. Ghosh, B. Viswanath, F. Kooti, N. K. Sharma, G. Korlam, F. Benevenuto, N. Ganguly, and K. P. Gummadi, "Understanding and combating link farming in the twitter social network," in *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, (New York, NY, USA), pp. 61–70, ACM, 2012.

[59] J. P. Dickerson, V. Kagan, and V. S. Subrahmanian, "Using sentiment to detect bots on twitter: Are humans more opinionated than bots?," in *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2014, Beijing, China, August 17-20, 2014*, pp. 620–627, 2014.

[60] OAuth, "OAuth," 2015. http://oauth.net.

[61] Twitter, "API Rate Limits," 2015. https://dev.twitter.com/rest/public/rate-limiting.