Redes Neurais Convolucionais na Detecção de Bots Sociais: Um Método Baseado na Clusterização de Mensagens Textuais

Paulo A. Braz[1], **Ronaldo R. Goldschmidt**[1]

[1]Seção de Engenharia da Computação (SE/8) – Instituto Militar de Engenharia (IME)
22.290-270 – Rio de Janeiro – RJ – Brazil

{paulo.braz,ronaldo.rgold}@ime.eb.br

Abstract. *Social bots are responsible for malicious activities in social networks. State-of-the-art on social bot detection combines account data with textual content statistics extracted from messages posted in those environments. Nevertheless, statistical consolidation may lead to information loss. In such scenario, this work searches for experimental evidence that supports the hypothesis that the usage of original textual content from messages may improve detection accuracy. To help our search, we developed a method that applies convolutional neural networks to identify suspicious messages based on their original textual content. Those network models are trained on samples selected from clustered messages. Experiments with our method on Twitter data confirm our hypothesis.*

Resumo. *Cada vez mais bots sociais executam atividades maliciosas em redes sociais. O estado da arte na detecção desse tipo de malware considera, entre outras informações, medidas estatísticas calculadas a partir do conteúdo das mensagens postadas nas redes. Como esses cálculos podem ocasionar perda de informação, o presente artigo busca evidências experimentais que apoiem a hipótese de que o uso do conteúdo textual original das mensagens pode aprimorar a precisão de detecção. Para esse fim, foi proposto um método que utiliza redes neurais convolucionais para identificar mensagens e contas suspeitas. Tais redes são treinadas com amostras obtidas pela clusterização dos textos originais das mensagens. Experimentos com o Twitter confirmam a hipótese levantada.*

1. Introdução

Na atualidade, redes sociais como Facebook e Twitter figuram como veículos amplamente utilizados para a troca de informações de cunho diversificado e democrático entre milhões de usuários em todo o mundo [Badri Satya et al. 2016]. Em função de sua natureza e popularidade, muitas vezes apresentam-se como ambientes onde é possível observar traços de percepção pública e tendências de opinião de grupos sobre diversos assuntos [Ratkiewicz et al. 2011].

Em geral, as redes sociais estão vulneráveis à ação dos chamados *bots* sociais. Um *bot* social é uma conta gerenciada por *software*, capaz de interagir com humanos e produzir conteúdo de forma automática [Ferrara et al. 2016]. Contas desse tipo podem ser utilizadas de forma maliciosa para tentar manipular opiniões de usuários das redes sociais por meio da propagação de notícias falsas com o objetivo de adulterar a percepção pública acerca da realidade [Freitas et al. 2014]. Como exemplo mais notável, tem-se a última eleição dos Estados Unidos sob investigação no que tange à atuação dos

bots sociais, sobretudo no expediente de disseminação de notícias falsas e difamação. [Allcott and Gentzkow 2017].

Neste contexto, a detecção de *bots* sociais é a tarefa de identificar contas automatizadas, a fim de mitigar os efeitos nocivos causados por essas entidades e suas ações. A maior parte dos trabalhos em detecção de *bots* sociais segue a abordagem baseada em atributos ou características comportamentais das contas [Ferrara et al. 2016]. Os trabalhos desta abordagem utilizam informações que descrevem o comportamento das diversas contas. Por exemplo: o número de amigos ou seguidores de uma dada conta, a quantidade de contas que cada conta segue, o número de mensagens propagadas em uma certa janela de tempo, a quantidade de caracteres na descrição do perfil da conta, a quantidade de fotos postadas no perfil, dentre outros. Diversos desses trabalhos aplicam algoritmos de aprendizado de máquina para, diante de dados comportamentais, construir modelos de classificação binária que sejam capazes de identificar quais contas são, de fato, *bots* sociais [Ferrara et al. 2016].

Além de informações comportamentais sobre as contas das redes sociais, alguns dos trabalhos da abordagem de detecção de *bots* sociais baseada em atributos também consideram dados estatísticos sobre os textos das mensagens propagadas pelas contas, tais como: quantidade de *links* nas mensagens, número de menções a terceiros via "@nomeusuário", etc [Ferrara et al. 2016]. Para tanto, o conteúdo textual das mensagens é submetido a uma etapa de engenharia de atributos, responsável pela extração desses dados. Contudo, como em qualquer consolidação de dados, este processo pode vir a incorrer na perda de informação [Beaudry and Renner 2012]. Assim, o presente trabalho levanta a hipótese de que a utilização de estatísticas para representar mensagens propagadas pode eliminar indícios relevantes para uma detecção de *bots* sociais mais precisa.

Isto posto, o objetivo deste artigo é apresentar evidências experimentais de que a utilização dos textos originais das mensagens postadas pode contribuir para produção de modelos para detecção de *bots* sociais mais precisos do que aqueles que se baseiam exclusivamente nos atributos comportamentais dessas contas. Para tanto, o artigo propõe um método de detecção de *bots* sociais que utiliza *clusterização* de dados a fim de selecionar amostras representativas de mensagens postadas na rede social para, em seguida, treinar uma rede neural convolucional[1] [LeCun et al. 2004] adaptada para analisar o conteúdo original dessas mensagens a fim de identificar quais são suspeitas de terem sido produzidas por *bots* sociais, uma evolução do que foi apresentado em [Braz and Goldschmidt 2017]. Em seguida, aplica-se um modelo que busca classificar cada conta como *bot* ou não *bot* com base nos atributos comportamentais e no percentual de mensagens da conta em questão que tenham sido consideradas suspeitas pela etapa anterior. Resultados obtidos a partir de experimentos envolvendo o Twitter fornecem indícios que confirmam a validade da hipótese levantada.

O restante do artigo segue organizado da seguinte forma: a seção 2 apresenta os principais trabalhos relacionados; a seção 3 exibe a formalização do método proposto, descrevendo cada uma das etapas envolvidas; a seção 4 descreve os experimentos realizados e dos resultados obtidos; e, por fim, a seção 5 destaca as principais contribuições do trabalho e aponta para possíveis alternativas de trabalhos futuros.

[1]Este tipo de rede pertence a uma classe de algoritmos de aprendizado de máquina capazes de identificar características complexas a partir de características mais simples [Bezerra 2016]

2. Trabalhos Relacionados

De acordo com [Ferrara et al. 2016], as abordagens utilizadas para empreender a detecção de *bots* sociais podem ser divididas em quatro grupos: topológicas, *crowdsourcing*, comportamentais e híbridas.

Também denominadas abordagens baseadas em grafos (do inglês, *graph-based*), as abordagens topológicas interpretam a rede social como um grafo onde os vértices representam as contas e as arestas denotam as ligações entre os usuários. A fim de detectar *bots* sociais, os trabalhos deste grupo consideram atributos topológicos (i.e. informações sobre a estrutura da rede) tais como medidas de centralidade e distribuição de graus dos vértices, dentre outras. Por exemplo, o trabalho descrito em [Beutel et al. 2013] apresenta uma solução voltada para o ambiente do Facebook que utiliza um algoritmo de detecção de comunidades para identificar *bots* sociais, partindo da premissa de que essas contas estão mais fortemente conectadas a outros *bots* do que ligados a contas legítimas. A tabela 1 apresenta exemplos de trabalhos que seguem esta abordagem.

Nas abordagens baseadas em *crowdsourcing* estão os trabalhos onde a rotulação das contas em *bot* e não *bot* é realizada de forma manual por pessoas com experiência na identificação de contas suspeitas. A partir do resultado do processo de rotulação, é feito um procedimento de consolidação de opiniões a fim de indicar como contas suspeitas aquelas que apresentem maior quantidade de rotulações do tipo *bot*. Como exemplos de trabalhos que pertencem às abordagens de *crowdsourcing* podem ser citados: [Stein et al. 2011] e [Wang et al. 2012].

Mais populares entre as abordagens de detecção de *bots* sociais, abordagens comportamentais compreendem os trabalhos cuja detecção se baseia em atributos sobre o perfil e o padrão comportamental das contas tais como: idade da conta, localização do usuário, tempo entre a geração de mensagens, taxa de envio de solicitação de amizade, quantidade de seguidores, etc. Por exemplo, o trabalho descrito em [Xiao et al. 2015] apresenta uma solução para o Linkedin que busca identificar *bots* sociais no momento do cadastro das contas, procurando evitar que contas suspeitas comecem a acessar a rede. Para tanto, os autores utilizam informações cadastrais da conta, o endereço IP utilizado no cadastro, entre outros. Outros exemplos de trabalhos que utilizam atributos comportamentais estão listados na tabela 1.

Nas abordagens híbridas (do inglês, hybrid-based) estão os trabalhos que utilizam tanto atributos topológicos quanto atributos comportamentais para detecção automática de *bots* sociais. [Yang et al. 2014] é um exemplo típico de trabalho que pertence a este grupo de abordagens. Nele, os autores apresentam uma solução para detecção de contas automatizadas no contexto da rede RenRen[2] (segunda maior rede social na China) utilizando dados da topologia da rede e padrões de click (do inglês, *clickstream*) de cada conta de usuário. A tabela 1 apresenta outros exemplos de trabalhos que utilizam abordagens híbridas na detecção de contas suspeitas.

Um outro aspecto importante que pode ser observado na tabela 1 é que seguindo a linha do uso de informações comportamentais sobre as contas, alguns dos trabalhos de detecção de *bots* sociais também consideram o conteúdo dos textos das mensagens propagadas pelas contas. Na maior parte deles, são extraídos dados estatísticos das mensagens

[2]http://www.renren-inc.com/en/

	Atributos Topológicos	Conteúdo Textual	Atributos Comportamentais	Estatísticas Textuais
[Alvisi et al. 2013]	X			
[Badri Satya et al. 2016]			X	
[Barbon et al. 2017]				X
[Beutel et al. 2013]	X		X	
[Boshmaf et al. 2015]	X		X	
[Braz and Goldschmidt 2017]		X	X	
[Cao et al. 2012]	X			
[Chu et al. 2012]			X	
[Davis et al. 2016]	X		X	X
[Gilani et al. 2017]	X		X	X
[Igawa et al. 2016]		X		
[Hwang et al. 2012]			X	
[Keretna et al. 2013]				X
[Kudugunta and Ferrara 2018]		X		X
[Xiao et al. 2015]			X	
[Yang et al. 2014]		X	X	X
[Yang et al. 2015]			X	
[Wang et al. 2013]			X	
[Wang et al. 2016]			X	

Tabela 1. Trabalhos de detecção de *bots* sociais publicados a partir de 2012 - Panorama sumarizado

tais como: quantidade de URL's no texto, quantidade menções únicas a contas de terceiros, número de palavras em um *tweet*, número de *POS tags* em um *tweet*, entropia de palavras utilizadas no *tweet*, dentre outros. Embora enriqueçam o conjunto de dados a ser utilizado na construção dos modelos de detecção, tais trabalhos assumem o risco de eliminar informações relevantes diante da consolidação estatística.

Cabe ressaltar que são poucos os trabalhos que consideram o conteúdo original dos textos das mensagens na construção dos modelos de classificação. O trabalho de [Igawa et al. 2016] figura como uma das exceções, ao apresentar uma técnica que utiliza o conteúdo textual original dos *tweets*, transformando os textos em sinais, fazendo uso de *wavelets*. Essas atuam como descritores da distribuição dos termos-chave encontrados nos *tweets* gerados pelas contas relacionadas. Deste modo, cada conjunto contendo todos os textos propagados de cada usuário representa um documento relacionado àquele usuário. Baseado em *Random Forests* [Ho 1995], o modelo concebido pelo referido trabalho apresentou uma acurácia competitiva de 94% na classificação das contas automatizadas, apenas buscando traços e assinaturas da escrita identificadas nos textos. Não obstante o bom desempenho, este trabalho ficou limitado a um *dataset* pequeno, contendo apenas 100 contas, sendo difícil não considerar uma eventual ocorrência de *overfitting*[3] do modelo gerado.

[Braz and Goldschmidt 2017] e [Kudugunta and Ferrara 2018] figuram como outros exemplos na utilização do texto original das mensagens, sendo os primeiros trabalhos a apresentar técnicas baseadas em *deep learning* para a detecção de contas automatizadas. [Kudugunta and Ferrara 2018] apresenta uma técnica que utiliza uma rede neural recorrente (LSTM - *Long Short-Term Memory*) em conjunto com redes convolucionais sobre os conteúdos textuais das mensagens e sobre as estatísticas extraídas a partir deles. Deste modo, os autores advogam que o modelo proposto atinge por volta de 99% de acurácia (AUC), figurando assim como o estado da arte. Já [Braz and Goldschmidt 2017] repre-

[3]Sobreajuste - quando o modelo ajusta demasiadamente ao conjunto de dados utlizado.

senta o primeiro trabalho a utilizar aprendizado profundo na detecção de *bots* sociais, até onde foi possível verificar. Os autores propuseram um método de classificação de contas chamado DetBot Alfa que utiliza um rotulador de mensagens baseado em uma rede neural convolucional [LeCun et al. 2004]. Para tal, o conteúdo original das mensagens é utilizado como entrada da rede, aplicando a formatação preconizada por uma técnica chamada *Char Quantization* [Zhang et al. 2015]. A partir disso, cada conta selecionada aleatoriamente recebe um dado relativo ao grau de suspeição depreendido pela rede acerca das mensagens. Assim sendo, quanto mais mensagens rotuladas como suspeitas, maior o nível de suspeição desta conta. Atributos comportamentais destas contas em conjunto com o grau de suspeição são, então, submetidos a diversos modelos de classificação para empreender a rotulação destas contas. O referido trabalho reportou evidências experimentais preliminares indicando o aprimoramento na detecção de *bots*, com o emprego do conteúdo textual original agregado às demais fontes de informação das contas. Não obstante, o método DetBot Alfa não se aplica a bases de dados volumosas, mas sim apenas a pequenas amostras selecionadas aleatoriamente dessas bases. Por conta disso, o presente trabalho descreve em detalhes na próxima seção o método DetBot Beta, uma evolução do método DetBot Alfa para lidar com um número maior e mais representativo de contas e mensagens.

3. Método Proposto

Denominado DetBot Beta, o método proposto é uma evolução do método DetBot Alfa apresentado em [Braz and Goldschmidt 2017] e tem como objetivo apoiar a busca por evidências experimentais de que a utilização dos textos originais das mensagens postadas pode contribuir na construção de modelos para detecção de *bots* sociais mais precisos do que aqueles que se baseiam exclusivamente nos atributos comportamentais dessas contas. Diferentemente do DetBot Alfa que seleciona amostras de contas e mensagens de forma aleatória, o DetBot Beta direciona o processo de seleção de forma a obter amostras representativas da rede completa, ou seja, mensagens de todo o *dataset*. Para tanto, agrupa as mensagens por similaridade de conteúdo a fim de obter amostras que representem cada *cluster* identificado. Em seguida, o DetBot Beta treina uma rede neural convolucional adaptada para analisar o conteúdo original dessas mensagens a fim de identificar quais são suspeitas de terem sido produzidas por *bots* sociais. Por fim, aplica um modelo que busca classificar cada conta v como *bot* ou não *bot* com base nos atributos comportamentais e no percentual de mensagens de v que tenham sido consideradas suspeitas pela etapa anterior.

Em termos formais, seja *N* uma rede social representada por um grafo dirigido $G(V, E)$, onde cada vértice $v \in V$ representa uma conta (ou usuário) de N e cada aresta $e \in E$ corresponde a um par ordenado entre duas contas u e v, representado por $e = (u, v)$, que indica que u segue v. Suponha que cada conta u possui um conjunto de mensagens $M_u = \{m_{u,1}, m_{u,2}, ..., m_{u,u_k}\}$ postadas por ela em N. O conjunto de todas as mensagens de N é representado por $M_N = \bigcup_{i=1}^{|V|} M_{u_i}$, onde $|V|$ corresponde à cardinalidade de V. Considere ainda que cada conta u possui uma lista ordenada de atributos $(u.a_1, u.a_2, ..., u.a_r, u.c)$, sendo $u.a_i$ comportamentais e $u.c$ um atributo binário que informa se u é *bot* ou não. O método proposto possui quatro etapas conforme ilustrado na Figura 1. A descrição de cada uma delas encontra-se detalhada a seguir.

A Etapa 1, Clusterização e Seleção de Mensagens, é responsável por construir

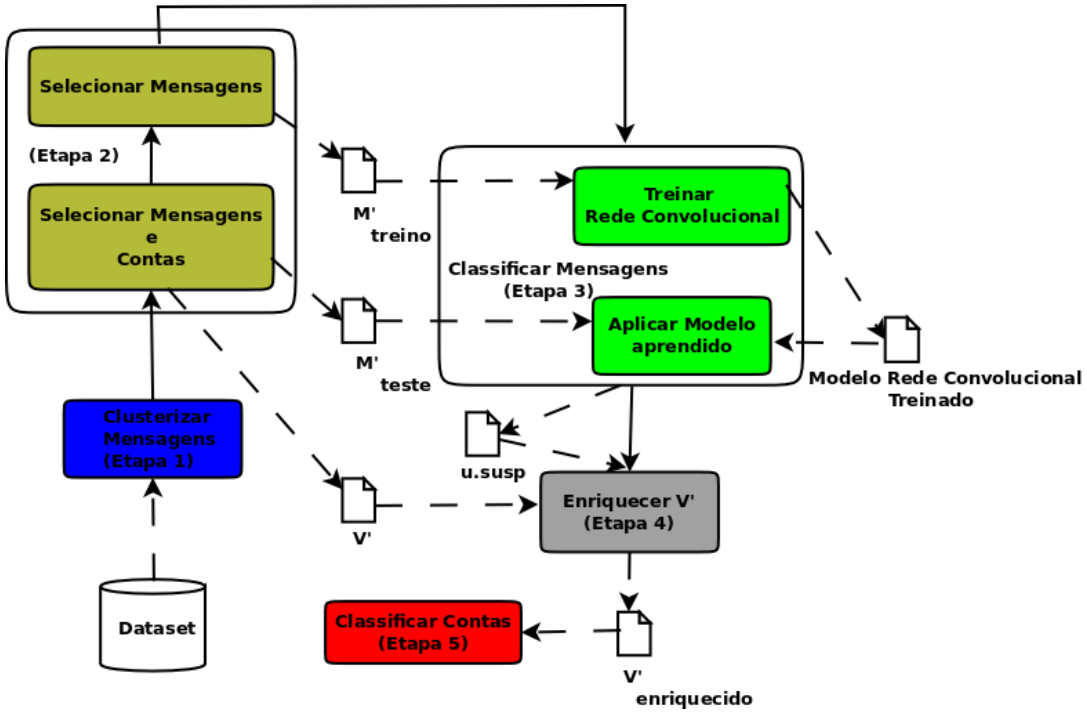

Figura 1. Visão Geral das Etapas do Método DetBot Beta.

subconjuntos de mensagens que serão utilizados pelas demais etapas. Inicialmente, considera-se $V' = \emptyset$, $V'_{Temp} = \emptyset$, $M'_{Treino} = \emptyset$ e $M'_{Teste} = \emptyset$. A construção dos subconjuntos é realizada da seguinte forma: executa-se uma *clusterização* do conjunto completo de mensagens M_N, o que contempla todo o conteúdo textual da rede N. Como saída, as mensagens acabam particionadas em k *clusters* disjuntos, representados por $C_1, C_2, ..., C_k$, onde k é a quantidade de conjuntos gerados. Em seguida, para cada *cluster* C_i são selecionadas aleatoriamente p mensagens de C_i e armazenadas em M''_i. Após a seleção das $p \times k$ mensagens, constroi-se M'_{Treino}, da seguinte maneira: $M'_{Treino} = \bigcup_{i=1}^{k} M''_k$. Para cada mensagem $m \in M'_{Treino}$, recupera-se a conta u tal que $m \in M_u$ e atualiza-se $V'_{Temp} = V'_{Temp} \cup \{u\}$. Após a construção de V'_{Temp} e M'_{Treino}, V' é formado por $|V'_{Temp}|$ contas u_i selecionadas aleatoriamente, de tal forma que $u_i \notin V'_{Temp}$. Por fim, para cada $u_i \in V'$, são selecionadas aleatoriamente p amostras de M_{u_i}. Em seguida tais amostras são incluídas em M'_{Teste}. Assim sendo, ao final do procedimento, tem-se que $M'_{Treino} \cap M'_{Teste} = \emptyset$. Em seguida, toda mensagem $m \in M'_{Treino} \cup M'_{Teste}$ é enriquecida com a classificação da conta u responsável por sua propagação, i. e., $u.c$. Tal informação é utilizada na etapa 2.

A Classificação de Mensagens, segunda etapa do processo, subdivide-se em dois passos. No primeiro, treina-se uma rede neural convolucional com o conjunto M'_{Treino} composto por amostras de mensgagens de cada *cluster* identificado. Para tanto, a rede recebe como entrada o texto original de cada mensagem m, sendo a classificação da mensagem a saída desejada $u.c$ a ser aprendida pela rede. Uma vez concluído o treinamento da rede neural, o segundo passo consiste em aplicar o modelo aprendido pela rede a cada uma das mensagens contidas em M'_{Teste}, composta por mensagens de contas de usuários selecionados aleatoriamente, contabilizando ao final, para cada conta u, o percentual de mensagens classificadas como sendo suspeitas de terem sido propagadas por *bots* sociais.

Denominada Enriquecimento de V', a etapa de número 3 do processo é responsável por acrecentar uma nova informação em cada uma das contas $u \in V'$. Assim, a lista ordenada de atributos que descreve u passa a ser $(u.a_1, u.a_2, ..., u.a_r, u.c, u.susp)$, onde $u.susp$ contém o percentual de mensagens de u classificadas como suspeitas na etapa anterior.

Dado um algoritmo de classificação S, a quarta e última etapa, Classificação de Contas, realiza um processo de validação cruzada com q conjuntos sobre V'. Este processo consiste em dividir V' em q conjuntos de contas para, em seguida, realizar q iterações. Em cada iteração, um dos q conjuntos é utilizado como conjunto de teste e os $q - 1$ restantes utilizados para treinamento de S. O desempenho do modelo de classificação construído a cada iteração é armazenado. O processo se repete até que todos os q conjuntos tenham sido utilizados uma vez como conjunto de teste. Ao final do processo, é obtido o desempenho médio dos q modelos gerados. Em cada iteração deste processo, os atributos $a_1, a_2, ..., a_r, susp$ são fornecidos como entradas e o atributo c como saída de S.

4. Experimento e Resultados

A fim de avaliar o método proposto em busca de indícios experimentais que apoiem a hipotése levantada neste artigo, foram realizados testes com o mesmo *dataset* adotado em [Braz and Goldschmidt 2017] e [Lee et al. 2011]. O referido dataset é composto por dados de contas (e suas respectivas mensagens) do *Twitter* relativos ao período de 30 de dezemdo de 2009 a 2 de agosto de 2010. O mesmo possui as seguintes características principais:

1. 22.2K contas *bots* e 19.2K contas de humanos legítimos e seus respectivos atributos comportamentais;
2. 2.3M de tweets gerados pelas contas *bot* e 3.2M de tweets gerados por contas dos usuários legítimos relacionados;
3. Diversidade de idiomas utilizados na escrita dos tweets: observam-se postagens escritas em 7 idiomas distintos: inglês, francês, alemão, mandarim, japonês, italiano e espanhol;

Primeiramente, foi necessário criar um processo para identificação de idiomas[4], de modo a manter no dataset para os processos de aprendizado apenas mensagens em um mesmo idioma: inglês. Para tal, utilizou-se um *toolkit* em Python contendo um dicionário[5] de palavras para o inglês contendo tanto termos tradicionais como termos voltados para a Web. Desta forma, após a aplicação desta filtragem o dataset passou a conter 3.71M de mensagens: 1.93M geradas por contas *bot* e 1.78M de mensagens criadas por usuários legítimos.

A base contém os atributos comportamentais indicados na Tabela 2, além dos textos de todas as mensagens publicadas no período. Ao todo, o *dataset* contém cerca de 40 mil contas e 5,5 milhões postagens relacionadas à estas contas, sendo 87 postagens por conta, em média, aproximadamente. A escolha deste *dataset* deveu-se fundamentalmente à disponibilidade dos atributos comportamentais em conjunto com os textos publicados pelas respectivas contas.

[4]Para tal, criou-se um script em Python para filtrar mensagens, mantendo apenas as escritas em inglês

[5]NLTK - https://www.nltk.org/

Tabela 2. Atributos comportamentais disponíveis no *dataset* do experimento.

Atributos	Descrição
# usuários seguindo	número de usuários que a conta segue
#tweets	número de tweets postados pela conta
razão de #seguindo por #seguidores	razão de #usuários seguidos por #seguidores
#usuários seguidores	número de seguidores

Como exposto na seção 2, em [Braz and Goldschmidt 2017] foi apresentado o DetBot Alfa, método capaz de empreender a classificação de contas considerando amostras de mensagens e contas selecionadas aleatoriamente. No presente trabalho, além de experimentos adicionais com o DetBot Alfa, foram realizados experimentos utilizando o método DetBot Beta, descrito na seção 3. Tal método busca classificar contas utilizando amostras de mensagens selecionadas a partir de grupos que representem o universo de mensagens da rede.

Desta forma, foram avaliados 7 cenários, ao todo. Nos 4 primeiros[6], foi utilizado o método DetBot Alfa. O método DetBot Beta foi aplicado nos 3 últimos. A tabela 3 apresenta detalhes estatísticos sobre a composição dos 7 cenários[7]. Já para a seleção de mensagens e contas dos cenários 1 ao 4 seguiu o procedimento resumido na seção 2 e detalhado em [Braz and Goldschmidt 2017].

A seleção de mensagens e contas dos cenários 5 a 7 seguiu o procedimento descrito na etapa 1 do método DetBot Beta. No entanto, antes de iniciar o processo de clusterização das mensagens disponíveis no *dataset*, foi necessário realizar algumas operações de pré-processamento dos textos.

Como no referido *dataset* existem *tweets* em diversos idiomas e uma mesma conta pode ter postado mensagens em mais de um idioma, a primeira operação realizada foi uma filtragem das contas cujas mensagens estivessem todas escritas na língua inglesa. Com isso, das cerca de 5.5 milhões de mensagens inicialmente existentes, restaram apenas 3.6 milhões. Para seleção e processamento destas mensagens em inglês, utilizou-se a biblioteca NLTK [Bird et al. 2009].

Logo, no tocante à clusterização das mensagens, do mesmo modo foi necessário representar cada um dos *tweets* como um vetor de palavras. A representação utilizada foi a *Bag of Words* [Liddy 2001].

O algoritmo de clusterização utilizado foi o k-Means [Jain and Dubes 1988] com a distância euclidiana. Para escolha do número de *clusters* k, foram realizados testes onde o valor de k variou de 4 a 10. Para cada valor, foi calculado o coeficiente de *Jaccard* [Salton and McGill 1986] entre os conjuntos de palavras associados aos *clusters* a fim de se verificar o grau de interseção entre eles. O valor selecionado ($k = 5$) foi o que apresentou menor índice de *Jaccard* entre todos os analisados.

Após a clusterização, variou-se a quantidade de mensagens selecionadas de cada

[6]Cabe destacar que os cenários 1 e 2 são os mesmos apresentados em [Braz and Goldschmidt 2017].

[7]É importante enfatizar que foi adotada a mesma distribuição de classes encontradas no *dataset* original, em relação ao número de contas: 50% de contas do tipo *bot* e 50% de contas legítimas, em todos os cenários.

cluster, levando aos três cenários indicados. Assim, no quinto cenário foram utilizadas 2.000 mensagens publicadas, no sexto, 6.000 mensagens e no sétimo, 20.000 mensagens para o treinamento e teste do classificador dos *tweets*.

Tabela 3. Dados estatísticos sobre os cenários do experimento

	x_{treino}	x_{teste}	$x_{treino} + x_{teste}$	$\|V'\|$	Média Msg/Conta	Contas *bot*	DetBot
1	1000	1000	2000	53	37.7	50%	Alfa
2	3000	3000	6000	160	37.5	50%	Alfa
3	10000	10000	20000	533	37.5	50%	Alfa
4	50000	50000	100000	2659	37.6	50%	Alfa
5	1000	1000	2000	30	33.3	50%	Beta
6	3000	3000	6000	90	33.3	50%	Beta
7	10000	10000	20000	100	100	50%	Beta

A etapa de classificação de mensagens foi implementada para todos os cenários com uma rede neural convolucional cuja configuração e arquitetura estão descritas na Tabela 4. A implementação deste modelo foi feita utilizando *Python*, *Keras* [Chollet et al. 2015] e Tensorflow [Abadi et al. 2016].

Tabela 4. Arquitetura da rede convolucional

	Filtros	Kernel	Stride
2 Camadas	150	7X7	1
4 Camadas	150	3X3	1
Fully Connected	31 neurônios	–	–
Última camada	2 neurônios	–	–

As mensagens precisaram ser formatadas a fim de serem submetidas à rede convolucional. Para esta formatação foi utilizada a técnica *Char Quantization* proposta por [Zhang et al. 2015][8]. Desta forma, cada texto de mensagem publicada foi transformado em uma matriz binária de dimensão 150x64, uma vez que 150 é o limite máximo de caracteres para cada postagem no Twitter e 64 é a quantidade de caracteres considerados para mapeamento. A Figura 2 apresenta o conjunto de caracteres considerados para fins de mapeamento. Assim, as matrizes geradas são esparsas, de forma que cada linha de uma matriz gerada corresponde a um e, somente um, caracter da mensagem associada e, portanto, teve o *bit* setado para 1 exatamente na coluna relativa ao caracter identificado e 0 para as demais colunas da linha em questão.

```
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v',
 'w', 'x', 'y', 'z', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '-', ',', ';', '.', '!', '?', ':', '/',
 '\',',_', '@', '#', '$', '%', '^', '&', '*', '+', '-', '=', '<', '>', '(', ')', '[', ']', '{', '}']
```

Figura 2. Conjunto de caracteres usado para mapeamento das mensagens.

[8]A escolha por este tipo de formatação deveu-se ao bom desempenho proporcionado pela técnica nos experimentos de classificação de texto relatados pelos autores.

Ainda na etapa de Classificação de Mensagens foi executado o procedimento de *holdout* (com 2/3 de M'_{Treino} para treino e 1/3 de M'_{Treino} para validação), para cada um dos cenários. Deste modo, com o classificador treinado utilizando M'_{Treino}, fez-se a apresentação de M'_{Teste} para o modelo efetuar a classificação de cada mensagem m e, por conseguinte, viabilizar o cálculo do percentual de mensagens classificadas como sendo geradas por *bots* para todas as contas da amostra $(u.susp)$. Deste modo, com as taxas geradas, pode-se enriquecer V', finalizando assim a execução da Etapa 3.

Assim como em [Braz and Goldschmidt 2017], para classificação das contas foram utilizados dois algoritmos de classificação: *Random Forest* [Ho 1995] e MLP (*Multilayer Perceptron*) [Rosenblatt 1961]. Em cada cenário, cada algoritmo foi submetido a um processo de validação cruzada utilizando 10 conjuntos. Os algoritmos executados foram instanciados com suas configurações *default*, utilizando Weka [Witten et al. 2016]. Importante notar que foi adotada a mesma configuração dos algoritmos durante todo o processo. A Tabela 5 apresenta os parâmetros adotados por cada algoritmo em todos os experimentos.

Tabela 5. Configuração utilizada para cada algoritmo de classificação

Algoritmo	Configuração
MLP	hidden layers= 4, learning rate=0.3, momentum=0.2, epochs=500
Random Forest	bag size percent = 100, batch size = 100, num iterations = 100

De modo a comparar a influência do conteúdo original dos textos no processo de detecção de *bots* sociais, os mesmos algoritmos de classificação utilizados na etapa 4 foram aplicados sobre o conjunto V' sem o enriquecimento realizado pelo método proposto. Também neste caso, foi realizado um processo de validação cruzada com 10 conjuntos.

Portanto, a tabela 6 apresenta os resultados para todos os cenários, onde do 1 ao 4 estão indicados os resultados obtidos pela utilização do método DetBot Alfa e, finalmente, do 5 ao 7 estão indicados os resultados para o método DetBot Beta. Isto posto, vale ressaltar que as acurácias apresentadas nesta tabela foram obtidas pelos algoritmos na classificação de contas utilizando ambos os métodos em duas situações a saber: uma com V' contendo apenas os atributos comportamentais da Tabela 2 e a outra com V' enriquecido pelo método proposto para conter também o percentual de mensagens suspeitas associado a cada conta. Pode-se então perceber que os algoritmos de classificação aplicados em V' enriquecido apresentaram desempenho superior aos mesmos algoritmos aplicados em V' sem enriquecimento, em todos os cenários, exceto no cenário 3, no emprego da rede MLP. Por conseguinte, os resultados figuram como evidências experimentais de que a utilização do conteúdo textual original das mensagens das contas de redes sociais pode contribuir para a confecção de modelos de detecção de *bots* sociais mais eficazes do que aqueles que se baseiam somente nos atributos comportamentais dessas contas, corroborando desta forma as evidências experimentais preliminares apresentadas em [Braz and Goldschmidt 2017].

A fim de confrontar os resultados obtidos pelos métodos de detecção Alfa e Beta, deve-se comparar separadamente o cenário 1 com o cenário 5, e o cenário 2 com o cenário 6, pois são os casos que apresentam o mesmo número de amostras de mensagens. Em ambas as comparações, o método Beta apresentou ganho de acurácia em relação ao Método

Alfa, sinalizando para uma maior adequação do método capaz de lidar com datasets mais volumosos.

Os melhores modelos gerados pela solução poderiam ser utilizados em ambiente de produção como uma solução de detecção de *bots* sociais, visando aumentar a segurança e aperfeiçoando a experiências dos usuários da rede social.

Tabela 6. Resultados: acurácia dos classificadores

Cenário	**Enriquecimento de V'**	**MLP**			**Random Forest**		
		Acc.	FPs	FNs	Acc.	FPs	FNs
1	Não	69.23%	11%	19%	**86.5**%	7%	5%
1	Sim	84.61%	7%	7%	**90.3**%	5%	3%
2	Não	80.5%	13%	6%	**89.3**%	4%	6%
2	Sim	81.76%	8%	9%	**90.5**%	3%	5%
3	Não	81.8%	12%	5%	**90.3**%	4%	5%
3	Sim	81.6%	12%	6%	**90.1**%	4%	5%
4	Não	85.5%	7%	7%	**90.3**%	5%	3%
4	Sim	86.76%	7%	5%	**92.5**%	3%	4%
5	Não	72.35%	10%	17%	**87.4**%	6%	5%
5	Sim	81.15%	7%	8%	**91.5**%	5%	3%
6	Não	82.7%	11%	6%	**89.8**%	4%	5%
6	Sim	83.74%	8%	8%	**90.9**%	3%	5%
7	Não	82.1%	12%	5%	**91.7**%	3%	5%
7	Sim	82.5%	12%	5%	**92.1**%	3%	4%

5. Considerações Finais

As redes sociais estão cada vez mais vulneráveis a ações dos *bots* sociais, contas automatizadas capazes de interagir com outros usuários e reproduzir o comportamento de uma conta legítima, sobretudo ao que tange à confecção de conteúdo de forma autônoma. Tais entidades são empregadas para manipulação de opiniões de usuários das redes sociais por meio de atividades maliciosas como disseminação de notícias falsas, adulteração de estatísticas de percepção pública, furto de contas, etc.

A extração de estatísticas do conteúdo textual das mensagens postadas baliza a grande maioria dos trabalhos voltados para a detecção de *bots* sociais. Visto que esta extração pode incorrer na perda de informação, o presente trabalho objetivou exibir indícios experimentais de que a utilização dos textos originais das mensagens é capaz de aprimorar a precisão da detecção de *bots* sociais. Para tal, foi proposto um método que aplica uma rede neural convolucional para identificar mensagens suspeitas, baseando-se nos atributos comportamentais e no percentual de mensagens depreendidas como suspeitas pelo modelo, no contexto de cada conta, o método emprega amostras mais representativas de mensagens do *dataset* ao identificar características de autoria destas mensagens propagadas. Por fim, o método ainda aplica um classificador binário para detectar a presença de *bots* sociais a partir destas informações já consolidadas. Desta forma, resultados obtidos a partir de uma base de dados do Twitter confirmam a adequação do método aqui proposto,

evidenciando que o emprego do conteúdo textual original das mensagens publicadas pelas contas pode cooperar para produção de modelos mais eficazes do que aqueles que se baseiam apenas em atributos comportamentais e estatísticas dos textos.

Como trabalhos futuros, um estudo mais aprofundado dos clusters para avaliar o ganho de representatividade das mensagens selecionadas. Outrossim, ainda seria válido avaliar a semântica de todas as mensagens confeccionadas. Adicionalmente, seria interessante também avaliar a adequação de outras técnicas de aprendizado profundo para a depreender traços de autoria das mensagens na detecção de *bots* sociais, tais como redes neurais recorrentes do tipo LSTM [Hochreiter and Schmidhuber 1997] em composição com outros modelos de aprendizado profundo tal como exposto em [Kudugunta and Ferrara 2018] e outras formas de representação do texto em [Vosoughi et al. 2016] e [Dhingra et al. 2016].

Referências

Abadi, M., Barham, P., Chen, J., Chen, Z., Yu, Y., and Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283.

Allcott, H. and Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–36.

Alvisi, L., Clement, A., Epasto, A., Lattanzi, S., and Panconesi, A. (2013). Sok: The evolution of sybil defense via social networks. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 382–396. IEEE.

Badri Satya, P. R., Lee, K., Lee, D., Tran, T., and Zhang, J. J. (2016). Uncovering fake likers in online social networks. In *Proceedings of the 25th ACM International on CIKM*, CIKM '16, pages 2365–2370, New York, NY, USA. ACM.

Barbon, S., Igawa, R. A., and Zarpelao, B. B. (2017). Authorship verification applied to detection of compromised accounts on online social networks. *Multimedia Tools and Applications*, 76(3):3213–3233.

Beaudry, N. J. and Renner, R. (2012). An intuitive proof of the data processing inequality. *Quantum Info. Comput.*, 12(5-6):432–441.

Beutel, A., Xu, W., Guruswami, V., Palow, C., and Faloutsos, C. (2013). Copycatch: Stopping group attacks by spotting lockstep behavior in social networks. In *Proceedings of the 22Nd International Conference on World Wide Web*, number - in WWW '13, pages 119–130, New York, NY, USA. ACM. -.

Bezerra, E. (2016). Introdução à aprendizagem profunda. `http://sbbd2016.fpc.ufba.br/sbbd2016/minicursos/minicurso3.pdf`.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. "O'Reilly Media, Inc.".

Boshmaf, Y., Logothetis, D., Siganos, G., Lería, J., Lorenzo, J., Ripeanu, M., and Beznosov, K. (2015). Integro: Leveraging victim prediction for robust fake account detection in osns. In *NDSS*, volume 15, pages 8–11.

Braz, P. and Goldschmidt, R. (2017). Um método para detecção de bots sociais baseado em redes neurais convolucionais aplicadas em mensagens textuais. In *SBSeg 2017()*.

Cao, Q., Sirivianos, M., Yang, X., and Pregueiro, T. (2012). Aiding the detection of fake accounts in large scale social online services. In *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 197–210, San Jose, CA. USENIX.

Chollet, F. et al. (2015). Keras. `https://github.com/fchollet/keras`.

Chu, Z., Gianvecchio, S., Wang, H., and Jajodia, S. (2012). Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *IEEE Transactions on Dependable and Secure Computing*, 9(6):811–824.

Davis, C. A., Varol, O., Ferrara, E., Flammini, A., and Menczer, F. (2016). Botornot: A system to evaluate social bots. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 273–274. International World Wide Web Conferences Steering Committee.

Dhingra, B., Zhou, Z., Fitzpatrick, D., Muehl, M., and Cohen, W. W. (2016). Tweet2vec: Character-based distributed representations for social media. *arXiv preprint arXiv:1605.03481*.

Ferrara, E., Varol, O., Davis, C., Menczer, F., and Flammini, A. (2016). The rise of social bots. *Commun. ACM*, 59(7):96–104.

Freitas, C., Benevenuto, F., and Veloso, A. (2014). Socialbots: Implicaç oes na segurança e na credibilidade de serviços baseados no twitter.

Gilani, Z., Farahbakhsh, R., Tyson, G., Wang, L., and Crowcroft, J. (2017). Of bots and humans (on twitter). In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 349–354. ACM.

Ho, T. K. (1995). Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hwang, T., Pearce, I., and Nanis, M. (2012). Socialbots: Voices from the fronts. *interactions*, 19(2):38–45.

Igawa, R. A., Barbon Jr, S., Paulo, K. C. S., Kido, G. S., Guido, R. C., Júnior, M. L. P., and Silva, I. N. d. (2016). Account classification in online social networks with lbca and wavelets. *Inf. Sci.*, 332(C):72–83.

Jain, A. K. and Dubes, R. C. (1988). Algorithms for clustering data.

Keretna, S., Hossny, A., and Creighton, D. (2013). Recognising user identity in twitter social networks via text mining. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 3079–3082. IEEE.

Kudugunta, S. and Ferrara, E. (2018). Deep neural networks for bot detection. *arXiv preprint arXiv:1802.04289*.

LeCun, Y., Huang, F. J., and Bottou, L. (2004). Learning methods for generic object recognition with invariance to pose and lighting. In *Computer Vision and Pattern Recog-*

nition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 2, pages II–104. IEEE.

Lee, K., Eoff, B. D., and Caverlee, J. (2011). Seven months with the devils: A long-term study of content polluters on twitter.

Liddy, E. D. (2001). Natural language processing.

Ratkiewicz, J., Conover, M., Meiss, M. R., Gonçalves, B., Flammini, A., and Menczer, F. (2011). Detecting and tracking political abuse in social media.

Rosenblatt, F. (1961). Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, CORNELL AERONAUTICAL LAB INC BUFFALO NY.

Salton, G. and McGill, M. J. (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

Stein, T., Chen, E., and Mangla, K. (2011). Facebook immune system. In *Proceedings of the 4th Workshop on Social Network Systems*, page 8. ACM.

Vosoughi, S., Vijayaraghavan, P., and Roy, D. (2016). Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder. In *Proceedings of the 39th International ACM SIGIR*, SIGIR '16, pages 1041–1044, New York, NY, USA. ACM.

Wang, G., Konolige, T., Wilson, C., Wang, X., Zheng, H., and Zhao, B. Y. (2013). You are how you click: Clickstream analysis for sybil detection. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*, pages 241–256.

Wang, G., Mohanlal, M., Wilson, C., Wang, X., Metzger, M., Zheng, H., and Zhao, B. Y. (2012). Social turing tests: Crowdsourcing sybil detection. *arXiv preprint arXiv:1205.3856*.

Wang, G., Zhang, X., Tang, S., Zheng, H., and Zhao, B. Y. (2016). Unsupervised clickstream clustering for user behavior analysis. In *SIGCHI Conference on Human Factors in Computing Systems*.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Xiao, C., Freeman, D. M., and Hwa, T. (2015). Detecting clusters of fake accounts in online social networks. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security*, number - in AISec '15, pages 91–101, New York, NY, USA. ACM. 1/12/2016.

Yang, Z., Wilson, C., Wang, X., Gao, T., Zhao, B. Y., and Dai, Y. (2014). Uncovering social network sybils in the wild. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(1):2.

Yang, Z., Xue, J., Yang, X., Wang, X., and Dai, Y. (2015). Votetrust: Leveraging friend invitation graph to defend against social network sybils. -, -. 16/07/2016.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.