ATAQUES AUTOMATIZADOS DE ENGENHARIA SOCIAL UTILIZANDO O MODELO OCULTO DE MARKOV

Marcos Souza da Silva[1], Rangel Espindola da Rosa Junior[2]

[1]Universidade do Sul de Santa Catarina, Tubarão, Brasil

[2]Universidade do Sul de Santa Catarina, Tubarão, Brasil

marcos.silvase64@gmail.com, rangel.darosa@outlook.com

Abstract. *Given the large amount of data that is shared on social networks nowadays, this article presents an automated application of social engineering attacks that, with base on the collected information on the social network Twitter of a specified person, generates a sentence with a "malicious" link based on the most posted subject of this person. The developed application utilizes the Hidden Markov Model, a chain rewrite system based on the training and mining of collected data. This article is based on this fact and demonstrates that the social networks besides bringing entertainment can also be a big threat to your personal and/or professional life. Besides that, help companies to prevent that your employees become victims of social engineering attacks. Through a qualitative evaluation approach, it achieved in average of 76% of the success rate (correctly constructed sentences). In turn, a quantitative analysis showed that there were one or more accesses in 94% of the links sent.*

Resumo. *Tendo em vista a grande quantidade de dados que é compartilhada nas redes sociais hoje em dia, este artigo apresenta uma aplicação automatizada de ataques de engenharia social que, com base em informações coletadas na rede social Twitter de uma pessoa específica, seja então gerada uma sentença juntamente com um link "malicioso" baseado nos assuntos mais publicados por essa pessoa. A aplicação desenvolvida utiliza o Modelo Oculto de Markov, um sistema de reescrita de cadeias com base no treinamento e mineração de dados coletados. O presente artigo se baseia neste fato e demonstra que as redes sociais além de trazer entretenimento, pode se tornar uma grande ameaça para sua vida pessoal e/ou profissional. Além disso auxiliar empresas para prevenir que seus funcionários sejam vítimas de engenharia social. Através da uma abordagem de avaliação qualitativa foi obtida em média 76% de taxa de acerto (frases construídas corretamente). Por sua vez, uma análise quantitativa demonstrou haver um ou mais acessos em 94% dos links enviados.*

1. Introdução

O compartilhamento de informações se tornou comum e rotineiro e com isso acabamos compartilhando as nossas vidas livremente para a toda ou parte da internet. É de conhecimento mútuo que não existem apenas pessoas bem-intencionadas navegando pelas redes sociais. Contudo, sempre cremos na premissa de que nada de ruim acontecerá conosco. Pensamos que estamos protegidos pela plataforma ao adicionarmos somente quem conhecemos como colegas de trabalho, amigos e familiares. Porém realmente

conhecemos a fundo cada pessoa que adicionamos? Realmente sabemos quem ela diz ser? A plataforma de fato nos mantém seguros?

Um Estudo realizado em 2013 sobre engenharia social nas redes sociais online, analisando o grau de exposição de informações pessoais, revelou que dentre os indivíduos verificados "70% apresentam um indicador de exposição alta e 20% extremamente alta em relação às informações levantadas. Nenhum dos indivíduos pesquisados apresentou zero grau de exposição" (SILVA; ARAÚJO; AZEVEDO; 2013, p.1).

Com o avanço da tecnologia, as empresas foram se adequando conforme ocorria a evolução tecnológica, e com isso focaram-se na segurança na parte de software, acreditando que assim seria impossível alguém violar suas informações, porém esqueceram de um problema antigo que atinge milhares de pessoas e não somente na tecnologia da informação, como em qualquer área de conhecimento, a tão famosa engenharia social que é conhecida por ser um dos elos mais fracos na segurança de qualquer empresa, a capacidade humana de persuadir e enganar o próximo pode levar um funcionário de uma empresa a revelar informações cruciais que podem destruir toda uma barreira de segurança proposta pela empresa, ou seja, o risco na segurança da informação está extremamente maior na falha humana do que na tecnologia em si.

Uma pesquisa realizada pela empresa de segurança Check Point mostra que foram entrevistados 850 profissionais de TI, onde 48% já haviam sido vítimas de engenharia social e consequentemente tiveram 25 ou mais ataques, esses ataques custaram as vítimas de 25.000 a 100.000 dólares por incidente, nessa estatística foi revelado que as empresas estão dando mais importância para a proteção lógica, sendo assim, muitos engenheiros sociais aproveitam para persuadir e subestimar a capacidade humana para coletar informações das empresas, podendo assim acarretar em um grande prejuízo. (SILVA et al., 2012).

Em 2017 a Verizon Enterprise divulgou dados de incidentes de segurança da informação que foram identificados e documentados relacionadas a 65 empresas parceiras que ocorreram no ano de 2016. Foram analisados 42.120 incidentes, nos quais 1.965 resultaram em comprometimento de alguma informação, contudo 43% destes ataques envolviam alguma técnica de engenharia social. (VERIZON ENTERPRISE, 2017).

Tendo em vista a grande quantidade de informações publicadas diariamente nas redes sociais e os ataques de engenharia social que atingem diversas pessoas e empresas, foi desenvolvido uma aplicação automatizada para coletar dados na rede social (Twitter) de uma pessoa alvo. Baseado nesses dados e com a utilização do Modelo Oculto de Markov (RABINER, 1986) será gerado *n* (valor de *n* definido pelo usuário) frases juntamente com um link, onde o usuário da aplicação poderá escolher qual frase será enviada para a vítima ou se assim desejar a aplicação enviará uma automaticamente, com objetivo de despertar um estado instintivo ou emocional sobre ela, com o intuito de realizar uma prova de conceitos, concluindo assim se a pessoa acessaria o link, ou não.

A solução para o problema proposto tem como principal objetivo provar para a sociedade que as redes sociais além de ser uma forma de lazer e passatempo, se não forem utilizadas de maneira adequada, podem ocorrer danos irreparáveis, tanto em sua figura privada quanto pública. Além disso, a aplicação poderá ser utilizada para testar a segurança em ambientes empresariais, analisando o grau de comprometimento gerado pelos seus funcionários.

O presente artigo está estruturado da seguinte forma: a segunda seção apresenta os trabalhos que possuem similaridade ao problema proposto ou a aplicação desenvolvida. A terceira seção trata da contextualização do modelo clássico de Markov, o Modelo Oculto de Markov utilizado neste artigo e a forma de representação do mesmo em diagramas. A quarta seção descreve as ferramentas utilizadas para o desenvolvimento da aplicação e a metodologia computacional. A quinta seção demonstra os testes e os resultados obtidos com a aplicação, assim como discussões relacionadas ao mesmo. A sexta e última seção que enfatiza as conclusões relacionadas ao desenvolvimento da aplicação, além de melhorias e ideais para futuros trabalhos.

2. Trabalhos correlatos

Existem alguns trabalhos relacionados a engenharia social, porém o que mais se destaca no conceito de autômatos é o SNAP_R(Social Media Automated Phishing and Reconnaissance). Seymour e Tully (2016) desenvolveram esta aplicação utilizando inteligência artificial para realizar phishing[1] através do Twitter. Utilizando-se de redes neurais long short-term memory[2] que realizam a atividade de armazenar longas sentenças de dados aparentemente não estruturados - como sentenças - e o modelo de Markov para realizar os ataques de phishing.

Para avaliar o modelo, foram inseridos links encurtados dentro dos tweets gerados, se o alvo acessasse, seria gravado a data, a hora e o usuário para qual o tweet foi enviado. Foram realizados testes em 90 usuários, onde os resultados variam entre 30% e 60% de taxa de sucesso.

Albladi e Weir (2018) enviaram formulários a peritos em segurança da informação. Estes formulários continham três etapas, sendo a primeira etapa as perguntas referentes aos peritos e continham questionamentos como idade, sexo e nível de especialização. Já na segunda etapa da coleta, foram recolhidas informações pertinentes aos fatores e dimensões do framework, pontuando cada fator de zero a cinco. A última etapa tratava-se de perguntas sobre o próprio framework, sendo elas opcionais. Com as informações coletadas tornou-se possível desenvolver o framework que convertia os dados da segunda etapa em 14 fatores principais que então sucederam em 4 perfis dominantes sendo eles: sócio-psicológico, perceptivo, sócio-emocional e habitual.

A conclusão após os experimentos e a divisão das classes evidenciou que os usuários que possuíam conhecimento sobre o risco no qual estavam expostos, também haviam passado alguma experiência envolvendo engenharia social, salientando que a cautela por meio do usuário é extremamente necessária, principalmente nas configurações das redes sociais. O estudo também indicou que a rede social poderia reconhecer usuários mais vulneráveis a este tipo de ataque através dos seus hábitos ou até mesmo de seus atributos sociopsicológicos e conduzir tutoriais ou métodos de conscientização mais eficazes, sendo assim, identificando as intervenções mais eficientes com o intuito de prover proteção, segurança e privacidade aos seus usuários.

Ynoguti (1999) apresenta uma tese baseado em reconhecimento de fala contínua usando Modelos Ocultos de Markov, onde tem como objetivo estudar o problema de

[1] É o tipo de fraude por meio da qual um golpista tenta obter informações pessoais e/ou financeiras de uma pessoa, combinando meios técnicos e de engenharia social.

[2] Redes neurais long short-term memory (Gers et al, 2000)

reconhecimento de fala contínua, que difere do de palavras isoladas, sendo que o locutor não precisa fazer pausas entre as palavras. É analisada a influência de alguns conjuntos de subunidades fonéticas, e dos modelos de dedução e de linguagem no desempenho do sistema.

Para a avaliação do sistema foi confeccionada uma base de dados formada de 200 frases foneticamente balanceadas, com gravações de 40 locutores adultos, sendo 20 de cada sexo. Foi analisado o desempenho do sistema utilizando fones independentes de contexto e a influência do modo de operação do sistema na taxa de acertos.

Os resultados obtidos com os testes foram razoavelmente bons, com um índice de aproximadamente 80% de acerto de palavra para o caso de independente de locutor, chegando a quase 90% no caso dependente de locutor. Em relação aos testes com dependência de sexo, os resultados com locutores masculinos ficaram entre estes dois extremos, para locutores femininos, a taxa de acertos ficou abaixo dos testes realizados com independência de locutor, sendo assim foi analisado e identificado que a causa para este resultado foi a presença de um locutor feminino que o sistema apresentou um resultado muito ruim.

3. Contextualização

3.1. Cadeias de Markov

Segundo Souza (2013) Cadeia de Markov é um processo estocástico, onde esses processos podem ser tanto de tempo discreto como contínuo. O Processo estocástico em intervalos de tempos regulares e discretos, depende apenas do seu último estado para evoluir para o outro independentemente dos demais.

Cadeia de Markov como o próprio nome se refere, é formado por uma cadeia base, onde é possível observar a evolução que ocorre na cadeia, sendo assim o processo evolui de um estado para o outro de acordo com uma matriz de probabilidade pré-definida associada a cada estado, ou seja, é possível observar o processo em um todo até o resultado. (Rabiner, 1989).

Definindo assim um modelo de Cadeia de Markov, como proposto por (Espindula (2009) apud Trivedi (2006)):

- Um espaço de estados $S = \{s1, s2, s3, \cdots, sN\}$
- Uma variável estocástica X a assumir valores do espaço de estados S em diferentes instantes de tempo
- Uma distribuição de probabilidade inicial para cada estado $\Pi = \hat{} \{\pi i\}$, tal que $\pi i = P(X0 = Si)$
- Uma distribuição de probabilidade de transição entre estados $A\hat{} = \{aij\}$, tal que $aij = P(Xt = sj|Xt-1 = si)$

Por fim, as cadeias de Markov como já explicado anteriormente, são processos visíveis, onde ao fim de uma sequência de acontecimentos é possível saber exatamente o caminho percorrido, e a probabilidade final alcançada é dada pela multiplicação das probabilidades definidas nos estados pelos quais foi passado. Este tipo de modelagem é

um tanto restrito comparada ao Modelos Ocultos de Markov, no qual pode ser aplicado problemas mais complexos (Kleper, 2010).

3.2. Modelo Oculto de Markov

Espindola (2009) afirma que a diferença fundamental entre o Modelo Oculto de Markov e as demais técnicas Markovianas está na forma de observar o sistema, onde o modelo oculto observa-se de forma indireta, pois os observáveis são funções probabilísticas de transição entre os estados. Diferente dos demais modelos Markovianos onde a observação é direta.

Segundo Barbosa (2014) um modelo oculto de Markov é formado por uma cadeia de observáveis oculta ao sistema, não sendo possível observar os elementos que serão modelados e que da mesma forma não é possível determinar os parâmetros da cadeia, onde esses parâmetros serão compostos a partir das informações disponíveis para dinamicamente o modelo criar probabilidades de transações para formar observações(saída) finais.

O Modelo Oculto de Markov é formado por um conjunto de estados finitos, onde as transições entre eles são ativadas pela probabilidade, ou seja, cada saída é gerada de acordo com a distribuição de probabilidade treinada pelo modelo (RABINER, 1989).

Para Rabiner (1889) “o modelo resultante (que é chamado de modelo Markov oculto) é um processo estocástico duplamente embutido com um processo estocástico subjacente que não é observável.”

3.3. Representação do modelo markoviano oculto

Para representar de forma clara o funcionamento de um Modelo Oculto de Markov, será mostrado um exemplo considerando uma pequena cadeia de três estados apresentado por Waghabi e Benevides (2009) baseado em exemplos de RABINER (1989) e JUFRASKY E MARTIN (2006), representando a utilização do modelo markoviano oculto para previsões climáticas (Figura 1):

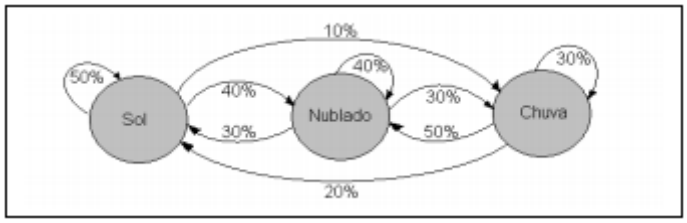


Figura 1. Cadeia de Markov para previsão do tempo (Waghabi e Benevides, 2009).

Tendo em vista o exemplo apresentado na Figura 1, imagine que um profissional que prevê o tempo, não possa observar diretamente o clima. O mesmo trabalha em um ambiente confinado e a única dica de como está o tempo é a roupa que seus colegas de trabalho estão vestindo ou se alguém levou o guarda-chuva. Como nem sempre seus colegas podem prever como vai se comportar o tempo, não significa que vai estar chovendo se um colega levar o guarda-chuva para o trabalho, mas a probabilidade de isso ser verdade é alta, da mesma forma que a probabilidade de estar frio, sendo que um colega apareceu no trabalho de jaqueta é alta, porém nada garante que seja uma verdade (Waghabi e Benevides, 2009).

Com base nessas observações, a cadeia de estados do tempo está oculta ao meteorologista, e o mesmo só conseguirá prever o clima, com as observações adquiridas no decorrer do tempo e consequentemente formando uma rede de probabilidades, conforme mostra a figura 2 a seguir (Waghabi e Benevides, 2009).

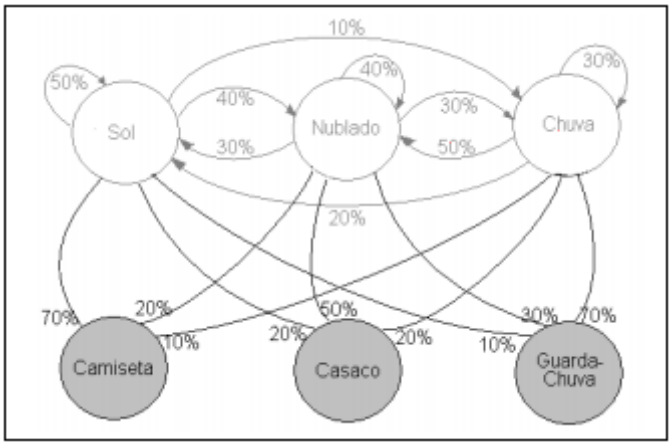


Figura 2. Modelo Oculto de Markov para previsão do tempo (Waghabi e Benevides, 2009).

Diante do modelo ilustrado na figura 2, o meteorologista ao observar o comportamento de seus colegas durante um tempo determinado, conseguirá definir qual a sequência de estado mais provável para produzir a sequência de observações e com base em probabilidade qual estado o sistema está no momento atual e como consequência a previsão para o dia seguinte (Waghabi e Benevides, 2009).

Definindo assim um Modelo Oculto de Markov da seguinte forma $\lambda = \{A, B, \Pi\}$ como composto por (Waghabi e Benevides (2009) apud Jufrasky e Martin (2006)):

- Um conjunto $Q=\{qi\}$ de estados ocultos do modelo, ou apenas estados.
- Uma matriz de probabilidades de transições $A=\{aij\}$ entre os estados qi e qj, onde $aij = [0, 1]$ com $i, j \in [1, |Q|]$, e $\sum aij = 1$ para um mesmo estado qi.
- Um conjunto $O=\{on\}$ de estados observáveis, ou observações.
- Uma matriz de probabilidades $B=\{bin\}$ indicando a chance do estado qi produzir a observação on, onde $bin = [0, 1]$ com $i \in [1, |Q|]$, $n \in [1, |O|]$.
- Uma distribuição $\Pi=\{\pi i\}$ de probabilidades de o modelo iniciar no estado qi, onde $\pi i = [0, 1]$ com $i \in [1, |Q|]$.

Levando a representação do Modelo Oculto de Markov para o problema proposto neste artigo, o modelo vai minerar e observar padrões nos dados coletados de uma determinada pessoa (alvo), e criar probabilidades de criação de frases de acordo com o que a vítima publica, ou seja, se depois da sentença “Eu vou” normalmente a vítima escreve “assistir” e vagamente escreve “jogar”, a probabilidade formada dinamicamente pelo modelo seguirá para a sentença “assistir”, assim montando sentenças com base nas probabilidade de transição treinada pelo algoritmo markoviano oculto.

4. Materiais e métodos

4.1. Ferramentas

Para desenvolver a aplicação, foi utilizada a linguagem de programação Python, com o auxílio da IDE PyCharm. Além de utilização de alguns módulos disponibilizados no Python, a seguir o modulo, sua versão e para que foi utilizado:

- **markovify==0.7.1:** É um pacote que tem a implementação do Modelo Oculto de Markov que é específico para receber grande quantidade de texto como entrada e gerar uma sentença como saída.
- **tweepy==3.6.0:** É uma API do Twitter onde foi utilizada para toda interação junto a rede social, como por exemplo: coleta dos dados e publicações.
- **requests==2.1.9.1:** Utilizada para efetuar requisições a aplicação web do Bitly, para realizar a autenticação, encurtamento de links, etc.
- **emoji==0.5.0:** Aplicado na identificação e remoção de emojis contidos nos dados coletados.
- **schedule==0.5.0:** Utilizado na fase de testes para executar funções determinadas a cada fração de tempo.

A rede social Twitter foi essencial para a realização do artigo, é onde ocorre todo o processamento da aplicação, desde a seleção do alvo, coletada de dados e publicações.

Foi utilizado um servidor na Amazon Web Services para realização de testes e análise de resultados da aplicação desenvolvida.

4.2. Método computacional

4.2.1 Coleta e pré-processamento

Para a construção da aplicação utilizando o Modelo Oculto de Markov(MOM), teve como primeira etapa a coleta de dados de uma pessoa alvo na rede social Twitter, onde o volume de dados depende diretamente do grau de utilização desta rede. A coleta de dados foi realizada utilizando a API do Twitter (Tweepy)[3], o trecho de código contido na figura 3 é utilizado para realizar a mineração tweets das timelines dos usuários e como resultado, terá uma lista contendo tweets como o da figura 4.

```
    timeline = [] # Declaração de uma variável que irá conter todos os tweets minerados

    for page_num in range(page_depth):
        timeline.extend( # Adicionando todo tweet minerado a variável
        api.user_timeline(id=screen_name, # É a pessoa alvo.
                          count=max_timeline_posts, # Quantidade máxima de Tweets por requisição,
                                                    # Limite da API: 200.
#
                          page=page_num,            # É o número da página.
                          include_rts=True))        # Se você deseja ou não minerar retweets.
    return timeline
```

Figura 3. Código responsável pela mineração de dados na timeline do usuário especificado.

[3] Site oficial do Tweepy: www.tweepy.org

{'created_at': 'Sat Oct 20 18:18:01 +0000 2018', 'id': 1053711953632333824, 'id_str': '1053711953632333824', 'text': '@LewisHamilton deixa o @ValtteriBottas ganhar as proximas se for campeão esse fds? #USGP #F1noSporTV #F1', 'truncated': False, 'entities': {'hashtags': [{'text': 'USGP', 'indices': [83, 88]}, {'text': 'F1noSporTV', 'indices': [89, 100]}, {'text': 'F1', 'indices': [101, 104]}], 'symbols': [], 'user_mentions': [{'screen_name': 'LewisHamilton', 'name': 'Lewis Hamilton', 'id': 213969309, 'id_str': '213969309', 'indices': [0, 14]}, {'screen_name': 'ValtteriBottas', 'name': 'Valtteri Bottas', 'id': 1143472657, 'id_str': '1143472657', 'indices': [23, 38]}], 'urls': []}, 'source': 'Twitter for Android', 'in_reply_to_status_id': None, 'in_reply_to_status_id_str': None, 'in_reply_to_user_id': 213969309, 'in_reply_to_user_id_str': '213969309', 'in_reply_to_screen_name': 'LewisHamilton', 'user': {'id': 964646156, 'id_str': '964646156', 'name': 'Vitima', 'screen_name': 'v', 'location': '', 'description': 'Vish....', 'url': None, 'entities': {'description': {'urls': []}}, 'protected': False, 'followers_count': 46, 'friends_count': 281, 'listed_count': 4, 'created_at': 'Thu

Figura 4. Tweet minerado

Logo após a coleta de dados, é necessário realizar a limpeza e pré-processamento dos dados, onde será removido todos os dados inconsistentes e que não agregam nenhum valor, com base em comparações para remover espaços vazios, menções a outros usuários, links encurtados, retweets, hashtag, entre outros. Esta etapa se torna de extrema importância para deixar o texto limpo e com condições de se tornar a entrada da aplicação e garantir a qualidade final da frase gerada (Figura 5).

deixa o ganhar as proximas se for campeão esse fds?
Actually no, since not every driver knowns the car as you would be unfair
Dado a atual situação da williams, melhor não ele so vai denegrir a carreira dele com os resultados...
Comunico que o Tenente-Coronel e Astronauta Marcos Pontes, engenheiro formado no ITA, será indicado para o Ministério da...
Cara ontem foi o dia q o bressan ficou sem emprego pq pqp entrar pra fazer merda ja vi agora pra foder um time primeira vez
Será que o Haddad vai visitar Lula na cadeia amanhã?
É a vitória da verdade contra a mentira, do capitalismo contra o comunismo, do honesto contra o corrupto e do bem contra...
LEWIS HAMILTON WINS 2018 FORMULA ONE WORLD CHAMPIONSHIP AND IS NOW A FIVE-TIME WORLD CHAMPION!!! Pass...
Vc rouba,mata, faz algo ilicito? Então n tem pq ter medo
Imagino a cabeça do Ricciardo sabendo q vai pra Renault ano q vem

Figura 5. Tweets após a etapa de pré-processamento

4.2.2 Aplicação do modelo

Tendo os dados limpos e pré-processados, a próxima etapa é a entrada dos dados na aplicação, onde a mesma tratará do processamento das informações coletadas utilizando o Modelo Oculto de Markov que vai mapear os dados para uma máquina de estados finitos, que dinamicamente criará probabilidades de transições entre os estados e os observáveis, gerando assim *n* sentenças (valor de *n* definido pelo usuário) com similaridades nos assuntos que a pessoa alvo tem interesse, como ilustra a figura 6.

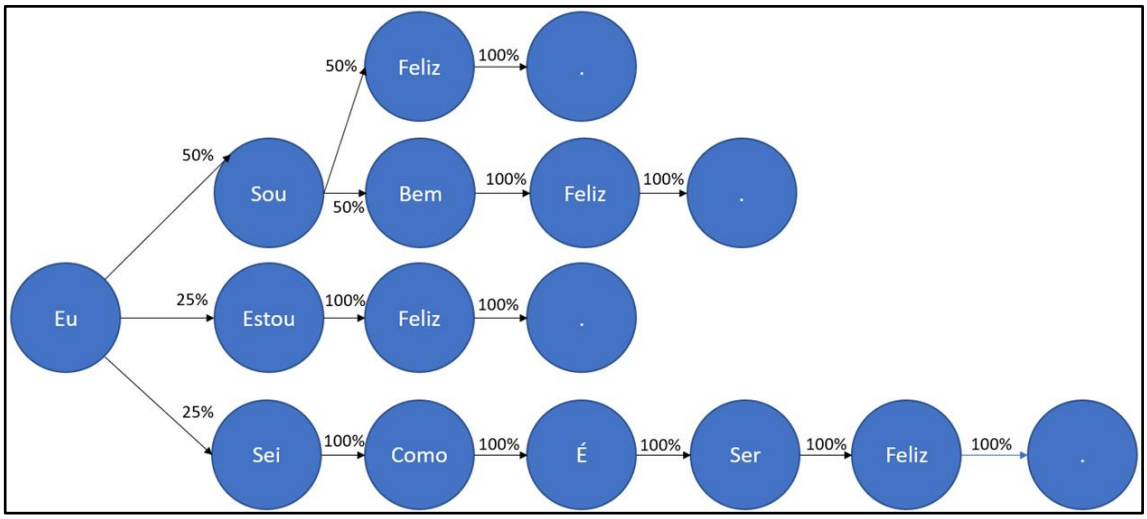


Figura 6. Ilustração do Modelo Oculto de Markov na geração de sentenças.

O pacote Markovify foi utilizado para processar os dados utilizando MOM disponível na linguagem Python e possuindo algumas particularidades que tiveram de ser

modeladas. O Markovify funciona melhor com textos grandes e bem pontuados, no caso da aplicação proposta, que coleta tweets da pessoa alvo e normalmente sem pontuação, foi preciso colocar cada tweet coletado em uma nova linha e utilizado a classe do pacote NewLineText ao invés da classe Text. Além disso o pacote traz três parâmetros interessantes que podem ser modificados conforme a necessidade e quantidade dos dados, conforme podemos observar na figura 7.

```
-h, --help            show this help message and exit
-m, --mine            Mines tweets from a specific target from --target
-mh, --minehashtags   Mines tweets from hashtags found on the target
                      timeline
-g, --generate        ONLY generates sentence USE WITH -f
-n N                  How many sentences to generate
-t TARGET, --target TARGET
                      Target to mine tweets from
-s SHORTLINK, --shortlink SHORTLINK
                      Long url to be shorten
-p, --post            Post the sentence generated
-f FILE, --file FILE  Define file with data to generate the sentence
-mr MAX_RATIO, --max-ratio MAX_RATIO
                      Max overlap ratio to generate the sentence
-mt MAX_TOTAL, --max-total MAX_TOTAL
                      Max overlap total when generating the sentence
-ml MAX_LENGTH, --max-length MAX_LENGTH
                      Max length of the sentence (including shorten link)
-os OUT_SENTENCES, --out-sentences OUT_SENTENCES
                      Exports all generated sentences
-v, --verbose         Verbose
```

Figura 7. Menu da aplicação, onde podem ser alterados os parâmetros.

O primeiro parâmetro a ser analisado "MAX_LENGTH", é a quantidade máxima de caracteres que a frase gerada pelo modelo poderá conter. O segundo "MAX_RATIO" e terceiro "MAX_TOTAL" parâmetro estão diretamente ligados ao método "make_sentence" do Markovify, pois este método realiza tentativas de gerar sentenças distintas das partes originais do texto, por padrão o parâmetro "MAX_RATIO" e o "MAX_TOTAL" tentam suprimir qualquer sentença gerada que sobreponham exatamente o texto original em 70% da contagem das palavras da sentença ou por 15 palavras.

```
python3 main.py -m -mh -p -v -mr 2 -mt 20 -ml 140 -t pessoa_alvo -s www.google.com
```

Figura 8. Sequência de comandos para geração da sentença.

Ao alterar os parâmetros explicados anteriormente, realizando testes e analisando os resultados de maneira empírica, selecionando frases que apresentaram concordância, como será mostrado na próxima seção, foi identificado que um determinado conjunto de parâmetros se adequa melhor para uma certa quantidade de dados e outro conjunto para uma outra quantidade de dados e o último para o meio termo, formando assim 3 módulos de conjuntos de parâmetros que podem se sobressair melhor dependendo da quantidade de dados disponível pelo alvo, conforme mostra o quadro 1. O usuário da aplicação pode definir os parâmetros de maneira manual como mostra a figura 8, onde estão definidos o conjunto de parâmetros para o volume de dados até 700, caso deseje que a aplicação preencha um ou mais parâmetros automaticamente conforme o volume de dados da pessoa alvo, basta não utilizar os parâmetros a serem preenchidos.

Quadro 1. Conjunto de parâmetros ativados conforme o volume de dados

Módulos (conjunto de parâmetros)	Volume de dados
MAX_LENGTH=140 - MAX_RATIO=2 - MAX_TOTAL=20	até 700
MAX_LENGTH=140 - MAX_RATIO=1 - MAX_TOTAL=10	entre 700 e 1500
MAX_LENGTH=140 - MAX_RATIO=0.5 - MAX_TOTAL=5	acima de 1500

Tendo como última etapa do modelo computacional a escolha e o envio da sentença para o alvo selecionado. A seleção da frase é feita através da console, selecionando o número correspondente a frase conforme mostra a figura 9. O envio da sentença é realizado juntamente com link encurtado através da API do Twitter (Tweepy). Para efetivar o encurtamento do link foi utilizado o módulo requests da linguagem Python, permitindo assim realizar as interações e requisições ao Bitly. É possível verificar o resultado deste processo na figura 10.

```
0 - Regenerate

1 - @                O Vettel tem que passar o Kimi sorri hoje?... http://bit.ly
/2CACwgO
2 -                  Tenho certeza de que Hamilton fizesse a parada do Vettel...
. http://bit.ly/2CACwgO
3 - @                E lá vamos ter show de hoje na tv aberta?... http://bit.ly/
2CACwgO
4 - @                Desde 2013 o Kimi e deixar o Kimi tá na cara... http://bit.
ly/2CACwgO
5 - @                Siroco pode ficar fora da corrida da vida da vitória de Rai
kkonen em Austin... http://bit.ly/2CACwgO
6 - @                Kimi é o finlandês com mais entusiasmo essa vitória do Icem
an.... http://bit.ly/2CACwgO
7 - @                Última vitória de Kimi em paz.... http://bit.ly/2CACwgO
8 - @                Equipe informando que o Kimi pro box mesmo... http://bit.ly
/2CACwgO
9 - @                Ainda bem que eu queria que o Vettel psicologicamente nesse
... http://bit.ly/2CACwgO
10 - @                O momento em que eu tô preferin... http://bit.ly/2CACwgO
```

Figura 9. Escolha da sentença para enviar para o alvo.

Figura 10. Sentença enviada para o alvo.

4.2.3 Avaliação e validação do modelo

Como forma de avaliação e validação da aplicação desenvolvida, foi utilizado tanto uma abordagem qualitativa como também quantitativa. Com a abordagem qualitativa foi analisado a qualidade final das sentenças geradas pela aplicação de forma manual, para assim identificar em um universo de *n* frases geradas de uma pessoa alvo, quantas frases tiveram concordância. Para a análise quantitativa foi elaborado um módulo que consiste em cinco etapas principais e automáticas, sendo elas: definir um alvo aleatório, executar a aplicação sobre este alvo, expandir uma base de conhecimento de postagens randômicas, realizar uma postagem randômica e recriar esta base de conhecimento.

Para a definição do alvo, este módulo localizava a hashtag mais utilizada no Brasil naquele momento, coletava dez tweets que interagiram com aquela hashtag e então dentre estes dez tweets definia randomicamente o alvo. Para a segunda etapa este módulo executava a aplicação enviando o nome do alvo definido anteriormente e utilizando como link "google.com/" concatenado ao nome do usuário codificado em um hash MD5[4], juntamente com os parâmetros para minerar a sua linha do tempo e hashtags mais utilizadas, conforme mostra a figura 7. Para a etapa de ampliação da base de dados inserimos os tweets minerados do último usuário alvo em um arquivo especificado, esta etapa é necessária para que houvesse uma maior diversidade de assuntos nos conteúdos publicados. Para as postagens randômicas foi utilizado o modelo oculto de Markov e o arquivo especificado como base de conhecimento para tal. Cada postagem randômica variava de cinco até doze minutos após a última a ser feita, tendo como intuito evitar o sistema de anti-bot do Twitter para que a aplicação não fosse identificada como um bot nocivo a rede social. Por fim, a base de dados de postagens randômicas era recriada a cada duas horas, para que não se acumulasse uma variedade muito grande de assuntos, diminuindo assim a qualidade das postagens feitas (Figura 11).

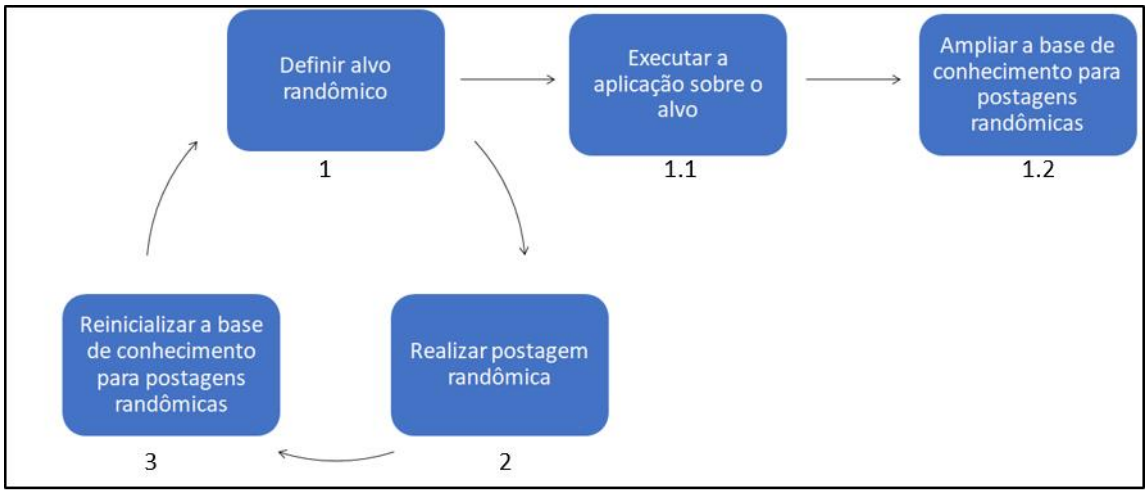


Figura 11. Diagrama do módulo para testes quantitativos.

Todas as sentenças foram enviadas entre os horários 12:39:00 e 02:40:00 UTC ao longo dos dias 31 de outubro e 04 de novembro. Tweets randômicos postados que feriam a ética, foram removidos do Twitter de testes. Vale ressaltar que somente tweets randômicos, ou seja, tweets que foram enviados para alvos não foram apagados, pois como a aplicação gerava sentenças de acordo com que o alvo tinha interesse concluímos que, mesmo contendo termos vulgares ou inapropriados, a sentença seria relevante para o alvo.

5. Resultados e Discussões

A aplicação foi executada em um servidor da Amazon Web Services com sistema operacional Ubuntu 4.15.0-1021-aws, com um processador Intel(R) Xeon(R) CPU ES 2676 v3 @ 2.40Hz, contendo 8 gigabytes de memória física e com 1 gigabyte de memória volátil.

[4] Message-Digest algorithm 5 (MD5): é uma função de dispersão criptográfica unidirecional de 128bits.

Foram realizados dois testes para a validação da aplicação, sendo um qualitativo onde eram analisadas as frases geradas e outro utilizando a abordagem quantitativa, realizando uma prova de conceito, enviando sentenças juntamente com um link para 100 pessoas escolhidas aleatoriamente com o intuito de saber quantas pessoas clicariam nos links enviados.

5.1 Abordagem qualitativa

O primeiro teste realizado foi através da abordagem de avaliação qualitativa, sendo que foi utilizado uma amostra de 10 pessoas, onde foram geradas 100 frases para cada uma delas e então avaliado a qualidade destas frases geradas de forma manual, selecionando frases com concordância que se aproximam de uma frase formada por um ser humano, conforme mostra a figura 12.

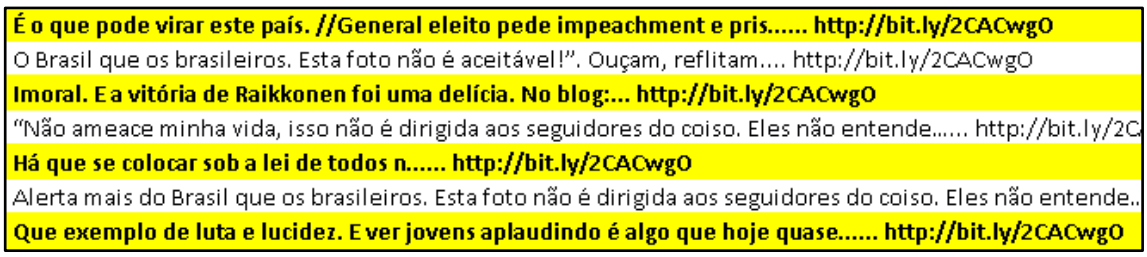


Figura 12. Sentenças selecionadas qualitativamente no Excel.

Ao realizar esta análise qualitativamente foi possível perceber que a qualidade dos dados influencia mais no resultado final do que a quantidade dos dados coletados, pois ao realizar testes em pessoas que digitam muito mal e a grande maioria dos seus tweets continham expressões curtas, contendo ou não links, a qualidade do resultado final era pior do que com pessoas que continham maiores expressões textuais e com um português melhor. Sendo assim, tendo um bom resultado para bases de conhecimento menores tanto quanto maiores, conforme mostra o quadro 2. No universo de 100 frases geradas para cada alvo, a média da taxa de acerto - frases com concordância - foi de 76% como ilustra o gráfico na figura 13.

Quadro 2. Resultado do teste qualitativo realizado nas sentenças geradas pela aplicação.

Teste Qualitativo			
Alvos	**Quant. dados (tweets)**	**Taxa de Acerto**	**Taxa de Erro**
Alvo 1	2833	68%	32%
Alvo 2	693	67%	33%
Alvo 3	3099	69%	31%
Alvo 4	3934	80%	20%
Alvo 5	4719	76%	24%
Alvo 6	3896	73%	27%
Alvo 7	498	80%	20%
Alvo 8	262	79%	21%
Alvo 9	2686	79%	21%
Alvo 10	2905	89%	11%

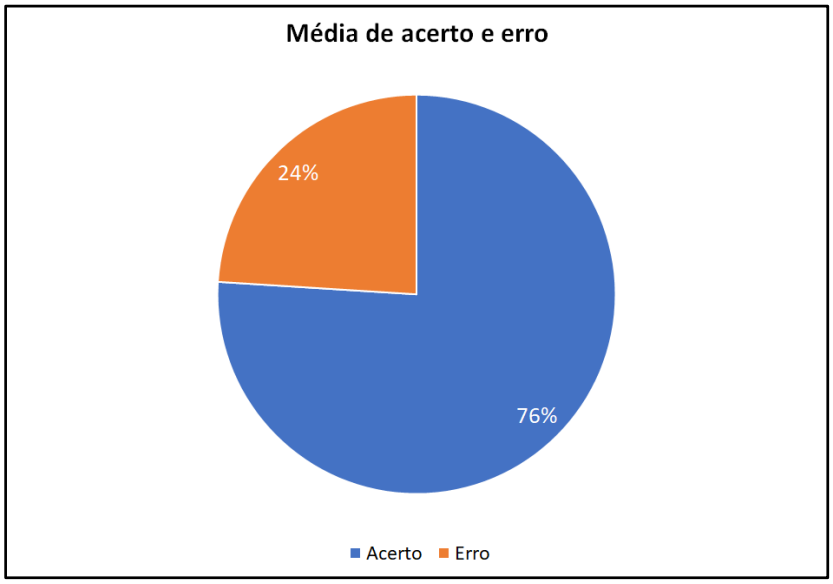


Figura 13. Média da taxa de acerto e erro do teste qualitativo realizado referente a quadro 1.

5.2 Abordagem quantitativa

No segundo teste utilizamos a abordagem de avaliação quantitativa, sendo realizado em uma amostra de 100 pessoas aleatórias, onde para cada pessoa foi gerado um link e uma sentença que logo após foi enviado através do Twitter, sendo uma das etapas do módulo citado na seção anterior.

Os resultados obtidos com base nesse teste foram interessantes, obtendo um ou mais acessos em 94% dos links gerados e enviados, ou seja, ao enviar uma sentença juntamente com um link para uma determinada pessoa no Twitter através de uma postagem, não impede que outras pessoas cliquem no mesmo, sendo assim, impossibilitando identificar se foi somente a pessoa alvo que clicou e se ela clicou, tornando o que poderia ser um ataque direcionado a uma pessoa passar a ser um ataque direcionado a uma comunidade (Figura 14).

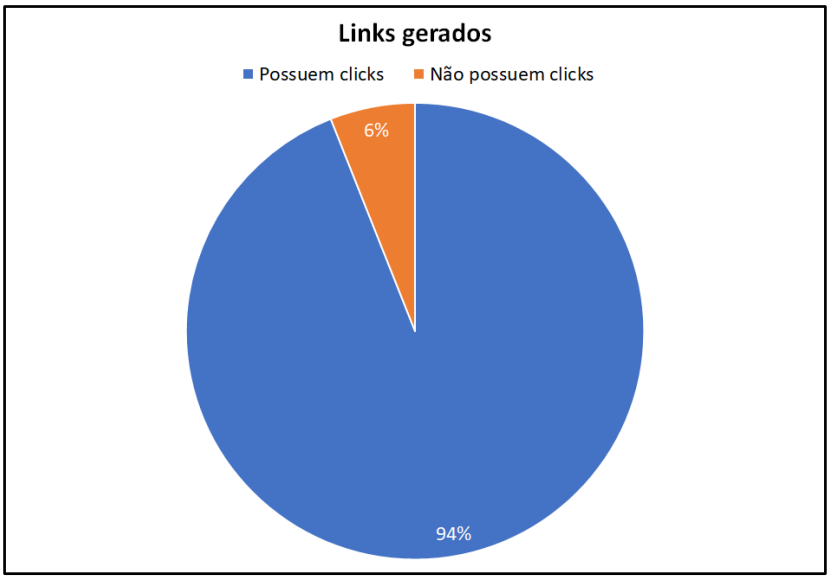


Figura 14. Acessos obtidos com os links enviados.

Ao analisar os resultados obtidos com a API do Bitly, foi possível observar que mesmo ao minerar somente hashtags brasileiras, foi obtido apenas 38 acessos com origem do Brasil, sendo assim, os resultados obtidos pela aplicação tiveram uma repercussão maior do que o esperado, como ilustra o gráfico na figura 15.

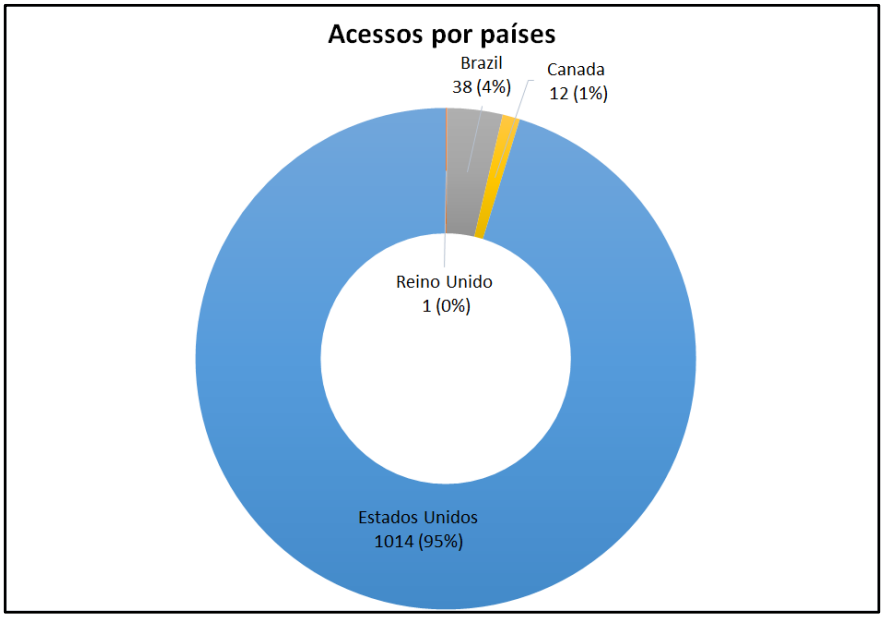


Figura 15. Origem dos acessos dos links enviados.

Outra característica que foi possível identificar ao analisar os resultados obtidos com realização do teste quantitativo, foi a importância da qualidade final da sentença gerada para a pessoa alvo, onde os links juntamente com frases com maior concordância e assuntos relevantes do momento tiveram uma maior quantidade de acessos. Como exemplo foi separado três frases com os maiores números de acessos e três frases que obtiveram nenhum acesso, conforme mostra o quadro 3. Ao analisar as frases que não obtiveram nenhum acesso, foi possível identificar a falta de concordância ou de um direcionamento para um assunto específico e em contrapartida as frases que obtiveram maiores acessos tinham uma maior concordância e também eram direcionadas a um assunto específico e de preferência um assunto do momento.

Quadro 3. Exemplos de sentenças enviadas para os alvos.

Sentenças Enviadas	Acessos
to sim, e estou com uma monta..... http://bit.ly/2yLVGMV	0
Eu precisarei muito do twitter estamos como:... http://bit.ly/2yQMIxZ	0
Realiza: Moro, ao lado dos mais pobres. Por João Sicsú... http://bit.ly/2yHHlky	0
o eleitor democrata do Alckmin, Meirelles, Marina, Amoedo e seus apoiadores doen..... htt	55
Não é questão de direita acaba de n..... http://bit.ly/2OobgmX	42
Algo me diz que negociação com o América em casa... http://bit.ly/2yPdBme	33

Os resultados obtidos ao realizar o teste utilizando a abordagem quantitativa teve como média 10,65 acessos para os links gerados, conforme ilustra o gráfico na figura 16.

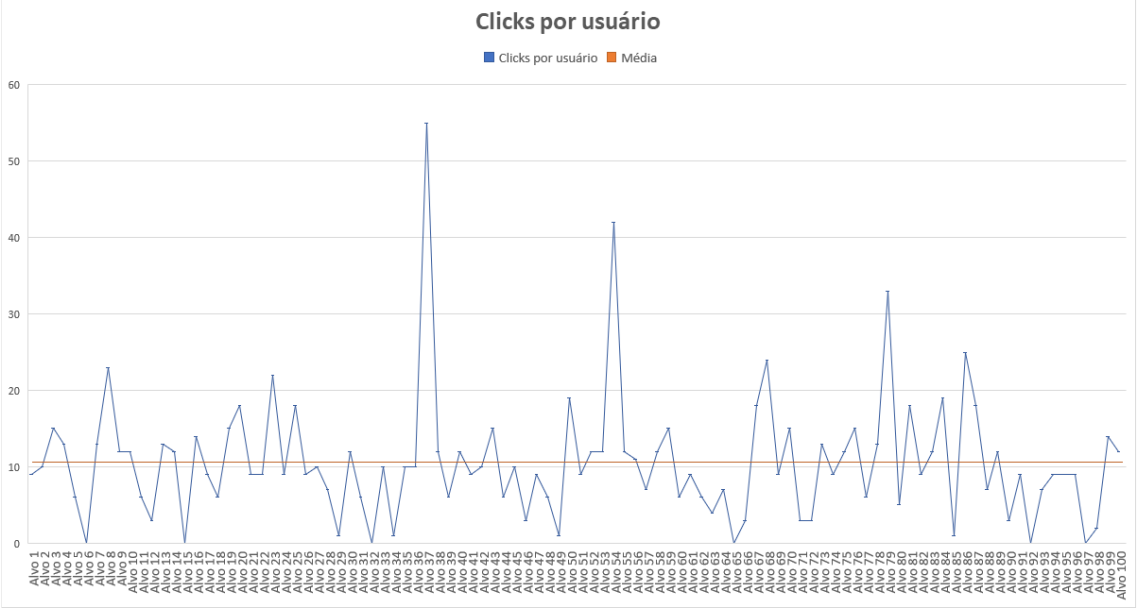


Figura 16. Todos os acessos para cada link enviado com base na média.

Um ponto a ser ressaltado são as interações dos usuários da rede social com as sentenças geradas pela aplicação automatizada ao longo dos testes, totalizando 29 interações, entre curtidas, respostas, retweets e seguidas, ilustrado no gráfico da figura 17 e apresentados no apêndice.

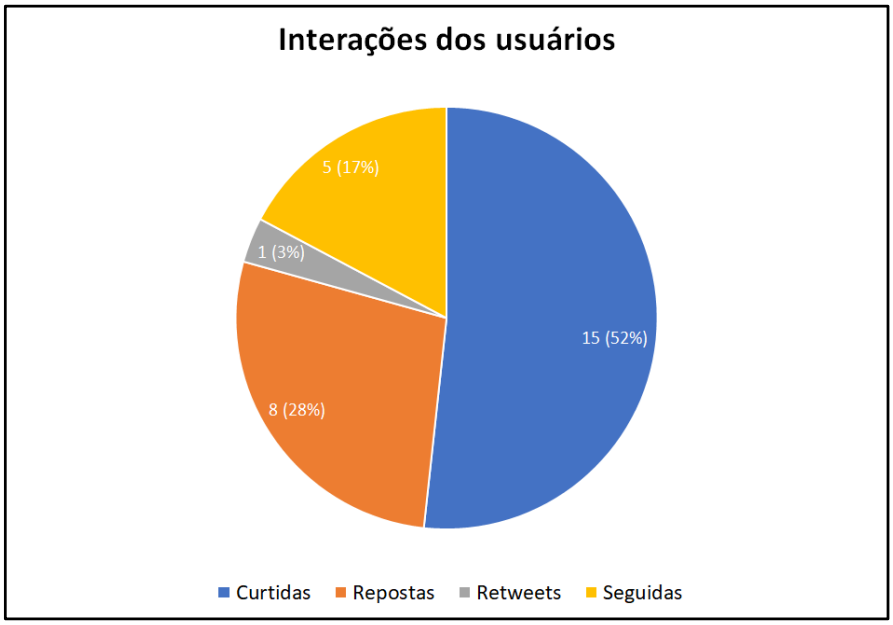


Figura 17. Interações dos usuários.

6. Conclusão

A engenharia social é um problema que afeta milhares de pessoas no mundo todo, a capacidade de persuasão e de enganar o próximo com ou sem o auxílio de um software, tendo como objetivo principal acessar informações valiosas e se beneficiar financeiramente com os dados obtidos. Sendo assim, com o avanço da tecnologia e o desenvolvimento dos softwares de segurança, aperfeiçoando-se cada vez mais para

identificar ameaças virtuais, a engenharia social ficou conhecida como um dos elos mais fracos na segurança, seja no ambiente empresarial ou pessoal.

Verificando os testes qualitativos e quantitativos podemos concluir que a aplicação obteve resultados melhores do que eram esperados, pois nos testes quantitativos obtivemos um ou mais acessos em 94% dos links gerados e enviados, ou seja, a aplicação atingiu o seu propósito de despertar um estado instintivo ou emocional sobre uma ou mais pessoas fazendo com que elas acessassem o link. Observando os testes qualitativos, identificamos que frases com concordância são constantes e dependem diretamente da qualidade dos dados coletados.

Conforme citado na seção de trabalhos correlatos, o trabalho que mais se assemelha a este é o SNAP_R desenvolvido por Seymour e Tully (2016), tendo como principal diferença o foco nas técnicas de inteligência artificial, pois focamos no modelo oculto de Markov e em seu funcionamento, possibilitando que ao longo dos testes qualitativos identificássemos conjuntos de parâmetros utilizados no algoritmo de Markov que correspondiam de maneira mais eficiente quando correlacionados a uma quantidade específica de dados, permitindo assim que fosse elaborado três conjuntos de parâmetros definidos automaticamente para as quantidades de dados disponíveis de cada usuário. Outro fator é que esta aplicação possui uma variedade maior de parâmetros que podem ser utilizados, permitindo que o usuário defina quantas frases serão geradas e inclusive que o mesmo escolha a sentença a ser enviada, também viabilizando que o usuário defina os parâmetros que serão utilizados no modelo oculto de Markov, caso opte por não utilizar os 3 conjuntos de parâmetros pré-definidos. Esta aplicação também proporciona ao usuário a possibilidade de gerar sentenças com base em um arquivo, o que oportuniza a aplicação minerar os tweets de um alvo somente uma vez e reutiliza-los para gerar sentenças quantas vezes forem necessários e exportar as sentenças geradas, caso deseje.

Como ideias para trabalhos futuros, pretendemos nos aprofundar e aperfeiçoar dois pontos na rede social Twitter, que são: Enviar a sentença através de uma mensagem privada para o alvo e lidar melhor com as menções aos usuários, pois sem as menções muitas frases ficavam incompletas o que prejudicava o desempenho do modelo oculto de Markov. Além destes dois pontos, pretendemos ampliar a aplicação desenvolvida neste artigo para que possa incorporar outras redes sociais.

7. Referências

ALBLADI, S. M. e WEIR, G. R. S. User characteristics that influence judgment of social engineering attacks in social networks. Hum. Cent. Comput. Inf. Sci. (2018) 8:5. Disponível em: <https://hcis-journal.springeropen.com/track/pdf/10.1186/s13673-018-0128-7>

Barbosa, Bruno de Lima. O uso de Modelos Ocultos de Markov no estudo do fluxo de rios intermitentes. Natal, Outubro de 2014. Disponível em: <https://repositorio.ufrn.br/jspui/bitstream/123456789/19513/1/BrunoDeLimaBarbosa_DISSERT.pdf>

Espindola, Luciana da Silveira. Um Estudo sobre Modelos Ocultos de Markov

HMM - Hidden Markov Model. Porto Alegre, junho de 2009. Disponível em: <https://www.inf.pucrs.br/peg/pub/tr/TI1_Luciana.pdf>

Gers, Felix A., Schmidhuber, Jürgen e Cummins, Fred. "Learning to forget: Continual prediction with LSTM." Neural Computation 12.10 (2000): 24512471.

Kepler, Fábio Natanael. Modelagem de contextos para aprendizado automático aplicado à análise morfossintática. São Paulo, julho de 2010. Disponível em: <http://www.teses.usp.br/teses/disponiveis/45/45134/tde-05082010-223515/pt-br.php>. Acessado 31/10/2018.

RABINER, L., 1989, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE, 77 (2), p. 257–286.

Rosenberg, M. e Frenkel, S. Facebook's Role in Data Misuse Sets Off Storms on Two Continents. 10 de março de 2018. Disponível em: <https://www.nytimes.com/2018/03/18/us/cambridge-analytica-facebook-privacy-data.html?rref=collection%2Fbyline%2Fmatthew-rosenberg>. Acessado em: 23/03/2018.

Seymour, J e Tully, P. Weaponizing data science for social engineering: Automated E2E spear phishing on Twitter. DEFCON, 2016. Disponível em: <https://media.defcon.org/DEF%20CON%2024/DEF%20CON%2024%20presentations/DEFCON-24-Seymour-Tully-Weaponizing-Data-Science-For-Social-Engineering-WP.pdf>

Silva, Narjara Bárbara Xavier; Araújo, Wagner Junqueira e Azevedo, Patrícia Morais de. Engenharia social nas redes sociais online: um estudo de caso sobre a exposição de informações pessoais e a necessidade de estratégias de segurança da informação. RICI: R.Ibero-amer. Ci. Inf., ISSN 1983-5213, Brasília, v. 6, n. 2, p. 37-55, ago./dez. 2013. Disponível em: <http://eprints.rclis.org/23215/1/Engenharia%20social%20nas%20redes%20sociais%20online.pdf>.

Silva, Clayton Silvestre; Rosa, Adriano Carlos Moraes; Chaim, Daniel Faria; Carvalho, Roberto José e Chimendes, Vanessa Cristhina Gatto. ENGENHARIA SOCIAL: O ELO MAIS FRÁGIL DA SEGURANÇA NAS EMPRESAS. Revista REAVI, Nº 02 – Dezembro 2012. Disponível em: <www.revistas.udesc.br/index.php/reavi/article/download/2840/2172>

Souza, Daniel Morais de. Modelos ocultos de Markov: uma Abordagem em Controle de Processos. Juiz de Fora, 2013. Disponível em: <http://www.ufjf.br/cursoestatistica/files/2014/04/Modelos-ocultos-de-Markov_-uma-Abordagem-em-Controle-de-Processos.pdf>

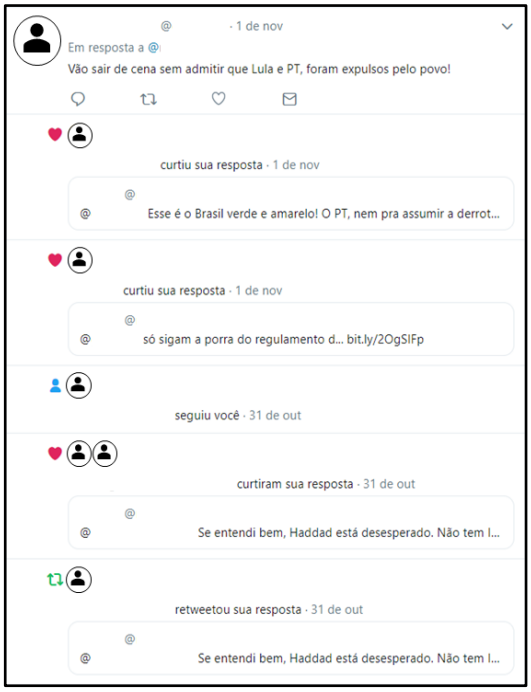
Verizon Enterprise, 2017 Data Breach Investigations Report 10th Edition. 2017. Disponível em: <http://www.verizonenterprise.com/resources/reports/rp_DBIR_2017_Report_en_xg.pdf>

Waghabi, Eduardo R. e Benevides, Mario R. F. Aplicação de Modelos Ocultos de Markov na Teoria dos Jogos. Rio de Janeiro, 2009. Disponível em: <http://csbc2009.inf.ufrgs.br/anais/pdf/enia/st02_05.pdf>.

Ynoguti, Carlos Alberto. Reconhecimento de Fala Contínua Usando Modelos Ocultos de Markov. Campinas, maio de 1999. Disponível em:

<http://www.decom.fee.unicamp.br/lpdf/teses_pdf/Tese-Doutorado-Carlos_Alberto_Ynoguti.pdf>

8. APÊNDICE




@ · 3 de nov
Em resposta a @
oq?
@ · 3 de nov
Em resposta a @
a
Google
404 Isso é um erro.
O URL solicitado /9c037c8bce12ae9eb07f2029516deedenão foi encontrado neste servidor. Isso é tudo que sabemos.