



**ESTRATÉGIAS DE ATAQUES ADVERSARIAIS EM UM
CLASSIFICADOR DE SENTIMENTO LÉXICO:
UMA ABORDAGEM DE MÍDIA SOCIAL**

GILDÁSIO ANTONIO DE OLIVEIRA JÚNIOR

**TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA**

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**ADVERSARIAL ATTACKS STRATEGIES IN A LEXICAL
SENTIMENT CLASSIFIER: A SOCIAL MEDIA APPROACH**

**ESTRATÉGIAS DE ATAQUES ADVERSARIAIS EM UM
CLASSIFICADOR DE SENTIMENTO LÉXICO: UMA ABORDAGEM
DE MÍDIA SOCIAL**

GILDÁSIO ANTONIO DE OLIVEIRA JÚNIOR

ORIENTADOR: RAFAEL TIMÓTEO DE SOUSA JÚNIOR, DR.

**TESE DE DOUTORADO EM ENGENHARIA
ELÉTRICA**

PUBLICAÇÃO: PPGENE.DM-172/20

BRASÍLIA/DF: DEZEMBRO - 2020

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**ESTRATÉGIAS DE ATAQUES ADVERSARIAIS EM UM
CLASSIFICADOR DE SENTIMENTO LÉXICO:
UMA ABORDAGEM DE MÍDIA SOCIAL**

GILDÁSIO ANTONIO DE OLIVEIRA JÚNIOR

**TESE DE DOUTORADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA
DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR.**

APROVADA POR:

**Prof. Rafael Timóteo de Sousa Júnior, Dr. – ENE/UnB
Orientador**

**Prof. William Ferreira Giozza, Dr. – ENE/UnB
Membro Interno**

**Profa. Edna Dias Canedo, Dra. – CIC/UnB
Membro Externo**

**Prof. Raul Ceretta Nunes, Dr. – CT/UFSM
Membro Externo**

BRASÍLIA, 21 DE DEZEMBRO DE 2020.

FICHA CATALOGRÁFICA

OLIVEIRA JÚNIOR, GILDÁSIO ANTONIO

Estratégias de Ataques Adversariais em um Classificador de Sentimento Léxico: uma Abordagem de Mídia Social [Distrito Federal] 2020.

xix, 122p., 210 x 297 mm (ENE/FT/UnB, Doutor, Engenharia Elétrica, 2020).

Tese de doutorado – Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

- | | |
|---------------------------|---------------------------------------|
| 1. Análise de Sentimentos | 2. Ataques Adversariais |
| 3. Engenharia Reversa | 4. Processamento de Linguagem Natural |
| I. ENE/FT/UnB | II. Título (série) |

REFERÊNCIA BIBLIOGRÁFICA

OLIVEIRA JÚNIOR, G. A. (2020). Estratégias de Ataques Adversariais em um Classificador de Sentimento Léxico: uma Abordagem de Mídia Social. Tese de doutorado em Engenharia Elétrica, Publicação PPGENE.DM-172/20, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 122p.

CESSÃO DE DIREITOS

AUTOR: Gildásio Antonio de Oliveira Júnior

TÍTULO: Estratégias de Ataques Adversariais em um Classificador de Sentimento Léxico: uma Abordagem de Mídia Social.

GRAU: Doutor ANO: 2020

É concedida à Universidade de Brasília permissão para reproduzir cópias desta tese de doutorado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa tese de doutorado pode ser reproduzida sem autorização por escrito do autor.

Gildásio Antonio de Oliveira Júnior

Departamento de Engenharia Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

*Dedico este trabalho aos meus pais,
à minha esposa, às minhas filhas e
aos pesquisadores deste país*

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me ajudar nos momentos difíceis e me mostrar novas oportunidades que contribuirão para o meu crescimento pessoal e profissional.

Aos meus pais pela excelente formação e educação, e especialmente, a minha esposa Sirrame, minhas filhas Lara e Luísa e a minha sobrinha Emilly, pela motivação e apoio incondicional nas horas difíceis.

Ao meu orientador Prof. Dr. Rafael Timóteo de Sousa Júnior, pelo constante apoio, incentivo e dedicação. Estes requisitos foram essenciais para o desenvolvimento deste trabalho e para o meu desenvolvimento como pesquisador.

Ao Prof. Dr. Robson de Oliveira Albuquerque, por acreditar no meu trabalho, pela disponibilidade, ideias e ensinamentos. Suas opiniões foram fundamentais para realização deste trabalho.

Aos meus amigos Marco Antônio, José Paulo Felipe, Adriano Bossonaro, Rômulo Bolson, Júlio Adilson, Francisco Regis, Rafael Ferraz, Claudinei Morin e Edson Alves, companheiros de trabalho e irmãos na amizade.

Aos professores da UNB, pesquisadores deste país, que de certa forma contribuíram para realização deste trabalho.

Agradeço também o apoio técnico e computacional do Laboratório de Tecnologias para Tomada de Decisão - LATITUDE, da Universidade de Brasília, que conta com apoio do CNPq - Conselho Nacional de Pesquisa (Outorgas 312180/2019-5 PQ-2, BRICS2017-591 LargEWiN e 465741/2014-2 INCT em Cibersegurança), da CAPES - Coordenação de Aperfeiçoamento do Pessoal de Nível Superior (Outorgas PROAP PPGEE/UnB, 23038.007604/2014-69 FORTE, e 88887.144009/2017-00 PROBRAL), da FAP-DF - Fundação de Amparo à Pesquisa do Distrito Federal (Outorgas 0193.001366/2016 UIoT e 0193.001365/2016 SSDDC), do Ministério da Economia (Outorgas 005/2016 DIPLA e 083/2016 ENAP), da Secretaria de Segurança Institucional da Presidência da República do Brasil (Outorga ABIN 002/2017), do Conselho Administrativo de Defesa Econômica (Outorga CADE 08700.000047/2019-14), da Advocacia Geral da União (Outorga AGU 697.935/2019) e dos DPI/DPG/UnB - Decanatos de Pesquisa e Inovação e de Pós-Graduação da Universidade de Brasília.

RESUMO

Título: Estratégias de Ataques Adversariais em um Classificador de Sentimento Léxico: uma Abordagem de Mídia Social

Autor: Gildásio Antonio de Oliveira Júnior

Orientador: RAFAEL TIMÓTEO DE SOUSA JÚNIOR, Dr.

Programa de Pós-Graduação em Engenharia Elétrica

Brasília, 21 de dezembro de 2020

As mídias sociais tornaram-se fonte de informações de relevância para diversos órgãos de Governo e empresas dos mais variados tipos. Essas informações são úteis para processos de tomada de decisão e definição de estratégias de negócio. Considerando este ponto de vista, várias técnicas de análise de sentimentos são empregadas para transformar dados coletados em conhecimentos que podem ser aplicados na inteligência. Um dos problemas é que os classificadores de sentimento utilizados nestes ambientes de coleta devem ser estudados e observados com cuidado, antes de serem instalados e confiados indiscriminadamente em sistemas de apoio à decisão. Pesquisas sobre as técnicas utilizadas em classificadores de sentimento tem se tornado um ponto crítico com novos métodos de ataques adversariais, onde pequenas perturbações podem ser criadas por usuários mal-intencionados para enganar os classificadores de sentimentos, gerando percepção diferente da realmente observada no ambiente. Nesse contexto, este trabalho apresenta o desenvolvimento de ataques adversariais em um classificador léxico de linguagem natural. Esse classificador, objeto dos ataques, é utilizado para calcular o sentimento dos dados postados e coletados por usuários em diversas aplicações de mídia social. Os resultados indicam que as vulnerabilidades encontradas, se exploradas por usuários mal-intencionados em aplicações que utilizam o mesmo classificador léxico, invertem ou anulam a percepção dos classificadores, gerando informação que não corresponde à realidade para a tomada de decisão. Esse trabalho ainda propõe algumas contramedidas que, se empregadas corretamente, são capazes de mitigar os ataques implementados.

Palavras-chave: Análise de Sentimentos, Ataques Adversariais, Engenharia Reversa, Processamento de Linguagem Natural.

ABSTRACT

Title: Adversarial Attacks Strategies in a Lexical Sentiment Classifier: a Social Media Approach

Author: Gildásio Antonio de Oliveira Júnior

Supervisor: RAFAEL TIMÓTEO DE SOUSA JÚNIOR, Dr.

Graduate Program in Electrical Engineering

Brasília, December 21th, 2020

Social media has become a source of relevant information for various government agencies and companies of the most varied types. This information is useful for decision-making processes and the definition of business strategies. Regarding this point of view, several sentiment analysis techniques are used to transform collected data into knowledge that can be applied to intelligence. One of the problems is that the sentiment classifiers used in these collection environments must be studied and observed before indiscriminately gaining trust and being installed in decision support systems. Research on the techniques used in sentiment classifiers has become a critical point with new methods of adversarial attacks, in which small perturbations can be created by malicious users to deceive the sentiment classifiers, generating a different perception from the one observed in the environment. In this context, this work presents the development of adversarial attacks in a lexical natural language classifier. This classifier, the object of the attacks, is used to calculate the sentiment of the data posted and collected by users in various social media applications. The results indicate that the vulnerabilities found, if exploited by malicious users in applications that use the same lexical classifier, could invert or cancel the classifiers' perception, generating information that does not correspond to the reality for decision making. This work also proposes some countermeasures that, if used correctly, can mitigate the implemented attacks.

Keywords: Sentiment Analysis, Adversarial Attacks, Reverse Engineering, Natural Language Processing.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	OBJETIVOS	2
1.2	RESUMO DAS PRINCIPAIS CONTRIBUIÇÕES	3
1.3	MOTIVAÇÃO E JUSTIFICATIVA	3
1.4	PUBLICAÇÕES.....	4
1.5	ESTRUTURA DO TRABALHO	4
2	ESTADO DA ARTE E TRABALHOS RELACIONADOS.....	6
2.1	ANÁLISE DE DADOS DE MÍDIA SOCIAL.....	6
2.2	BOT SOCIAL	7
2.3	SISTEMAS DE ANONIMATO	7
2.4	SISTEMAS DISTRIBUÍDOS	8
2.4.1	ELASTICSEARCH	8
2.4.2	KIBANA.....	10
2.5	VISUALIZAÇÃO DE DADOS.....	10
2.6	ANÁLISE DE SENTIMENTOS NA CLASSIFICAÇÃO DE TEXTOS	12
2.6.1	CLASSIFICAÇÃO COM ABORDAGEM LÉXICA	13
2.6.2	CLASSIFICAÇÃO COM TÉCNICAS DE APRENDIZADO DE MÁQUINA ...	17
2.6.3	CLASSIFICAÇÃO COM ABORDAGEM HÍBRIDA.....	21
2.6.4	COMPARAÇÃO DAS ABORDAGENS DE ANÁLISE DE SENTIMENTOS ...	25
2.7	ATAQUES ADVERSARIAIS EM ALGORITMOS DE CLASSIFICAÇÃO	28
2.8	CONTRIBUIÇÕES PRINCIPAIS DESTE TRABALHO	33
2.8.1	ESTRATÉGIAS DE ATAQUES ADVERSARIAIS	33
2.8.2	ARQUITETURA DE COLETA.....	36
2.9	RESUMO DO CAPÍTULO 2	37
3	DESCRIÇÃO DO PROBLEMA.....	38
3.1	PROPOSTA DA ARQUITETURA DE COLETA.....	39
3.2	DESCRIÇÃO DAS FASES DE IMPLEMENTAÇÃO.....	42
3.2.1	FASE 1: CAMADA DE COLETA DOS DADOS	42
3.2.2	FASE 2: SUBCAMADA DE PROCESSAMENTO DE DADOS	42
3.2.2.1	TRADUÇÃO E CORREÇÃO DOS DADOS TEXTUAIS.....	43
3.2.2.2	STOPWORDS E CARACTERES ESPECIAIS	43
3.2.2.3	TOKENIZAÇÃO	44
3.2.3	FASE 3: SUBCAMADA DE CLASSIFICAÇÃO	45
3.2.3.1	ANÁLISE DE SENTIMENTOS	45

3.2.4 FASE 4: CAMADA DE ARMAZENAMENTO DISTRIBUÍDO	45
3.2.5 FASE 5: CAMADA DE VISUALIZAÇÃO	46
3.3 EFICIÊNCIA DA ARQUITETURA DE COLETA	46
3.3.1 ESTUDO DE CASO I: COPA DO MUNDO FIFA 2018	47
3.3.1.1 COLETA DOS DADOS	47
3.3.1.2 APRESENTAÇÃO DO RESUMO GERAL DA COLETA	47
3.3.1.3 ANÁLISE DE TWEETS E RETWEETS	48
3.3.1.4 ANÁLISE DE HASHTAGS	49
3.3.1.5 APLICAÇÃO DE FILTROS	50
3.3.1.6 ANÁLISE DE USUÁRIOS	51
3.3.1.7 ANÁLISE DE SENTIMENTOS	52
3.3.1.8 ANÁLISE DE VÍNCULOS.....	55
3.3.1.9 ANÁLISE DA HASHTAG MAIS COMENTADA NAS QUARTAS DE FINAL.....	57
3.3.1.10 ANÁLISE DE OUTLIERS	58
3.3.1.11 ANÁLISE DE BOTNET	60
3.3.2 ESTUDO DE CASO II: COVID-19.....	63
3.3.2.1 COLETA DOS DADOS	63
3.3.2.2 APRESENTAÇÃO DO RESUMO GERAL DA COLETA	63
3.3.2.3 ANÁLISE DE SENTIMENTOS	64
3.3.2.4 ANÁLISE DE SENTIMENTOS POR IDIOMA	66
3.4 RESUMO DO CAPÍTULO 3	69
4 ENGENHARIA REVERSA NO ALGORITMO PATTERN ANALYZER DA BIBLIOTECA TEXTBLOB.....	71
4.1 BIBLIOTECA TEXTBLOB	71
4.1.1 ALGORITMO PATTERN ANALYZER.....	72
4.1.1.1 CONJUNTO DE DADOS LÉXICO	72
4.1.1.2 PALAVRAS DESCONHECIDAS.....	74
4.1.1.3 CÁLCULO DA POLARIDADE E SUBJETIVIDADE	74
4.1.1.4 POLARIDADE E SUBJETIVIDADE EM FRASES	76
4.1.1.5 POLARIDADE E SUBJETIVIDADE EM FRASES COM PALA- VRAS DE NEGAÇÃO.....	77
4.1.1.6 POLARIDADE E SUBJETIVIDADE EM FRASES COM ADVÉR- BIOS	78
4.1.1.7 POLARIDADE E SUBJETIVIDADE EM FRASES COM PALA- VRAS DE NEGAÇÃO E ADVÉRBIOS	81
4.1.1.8 POLARIDADE E SUBJETIVIDADE EM FRASES COM EMOTI- CONS.....	82

<i>SUMÁRIO</i>	x
4.2 RESUMO DO CAPÍTULO 4	84
5 ATAQUES ADVERSARIAIS EM ANÁLISE DE SENTIMENTO.....	85
5.1 ESTRATÉGIA DE ATAQUE I: INSERÇÃO DE CARACTERES	85
5.1.1 INSERÇÃO DE CARACTERES EM FRASES COM IDIOMA INGLÊS	86
5.1.1.1 CONTRAMEDIDA PARA OS ATAQUES DE INSERÇÃO EM FRASES COM IDIOMA INGLÊS.....	89
5.1.2 INSERÇÃO DE CARACTERES EM FRASES COM OUTROS IDIOMAS	90
5.1.2.1 CONTRAMEDIDA PARA OS ATAQUES DE INSERÇÃO EM FRASES COM OUTROS IDIOMAS	92
5.2 ESTRATÉGIA DE ATAQUE II: SUBSTITUIÇÃO DE PALAVRAS	94
5.2.1 SUBSTITUIÇÃO EM FRASES NEGATIVAS OU POSITIVAS	94
5.2.2 SUBSTITUIÇÃO EM FRASES COM PALAVRAS DE NEGAÇÃO	98
5.2.3 CONTRAMEDIDA PARA OS ATAQUES DE SUBSTITUIÇÃO	100
5.3 RESUMO DO CAPÍTULO 5	102
6 REPRESENTAÇÃO VISUAL DOS ATAQUES NO OCTOPUSVIZ.....	103
6.1 APLICAÇÃO DOS ATAQUES NO OCTOPUSVIZ.....	104
6.1.1 ATAQUES DE INSERÇÃO E SUBSTITUIÇÃO	105
6.1.2 REPRESENTAÇÃO VISUAL DOS ATAQUES	106
6.2 CONTRAMEDIDAS PARA MITIGAR OS ATAQUES NO OCTOPUSVIZ	109
6.3 RESUMO DO CAPÍTULO 6	111
7 CONCLUSÕES E TRABALHOS FUTUROS	112
7.1 ESTRATÉGIAS DE ATAQUES ADVERSARIAIS	112
7.2 ARQUITETURA DE COLETA	113
7.3 TRABALHOS FUTUROS	114
REFERÊNCIAS BIBLIOGRÁFICAS	114

LISTA DE FIGURAS

3.1	Arquitetura <i>OctopusViz</i>	40
3.2	Dados coletados no <i>Twitter</i> entre os dias 15 junho de 2018 e 31 de julho de 2018. <i>Histogram</i> com a quantidade de <i>tweets</i> e <i>retweets</i> coletados por dia (a). Quantidade total de <i>tweets</i> e <i>retweets</i> (b). Quantidade total de usuários e <i>hashtags</i> (c).	48
3.3	Dados coletados no <i>Twitter</i> entre os dias 15 junho de 2018 e 31 de julho de 2018. Classificação das mensagens em <i>tweets</i> e <i>retweets</i> (a). <i>Histogram</i> com a classificação das mensagens em <i>tweets</i> e <i>retweets</i> coletados por dia (b).....	49
3.4	Nuvem de palavras com a identificação das <i>hashtags</i> mais referenciadas nos <i>tweets</i> e <i>retweets</i>	49
3.5	Nuvem com as <i>hashtags</i> mais referenciadas. (a) indica <i>tweets</i> e (b) <i>retweets</i> . ..	50
3.6	Usuários que mais publicaram <i>tweets</i> e <i>retweets</i>	51
3.7	<i>Retweets</i> publicado pelo usuário <i>InfosFutebol</i>	52
3.8	Classificação (positivo, negativo e neutro) dos <i>tweets</i> e <i>retweets</i> por dia (a). Classificação (positivo, negativo e neutro) dos <i>tweets</i> e <i>retweets</i> por hora (b)...	53
3.9	Classificação dos <i>tweets</i> e <i>retweets</i> por dia com as polaridades neutra (a), positiva (b) e negativa (c).	54
3.10	Classificação geral dos <i>tweets</i> e <i>retweets</i> conforme o algoritmo <i>Pattern Analyzer</i>	55
3.11	Relacionamento entre as entidades polaridade (positiva, negativa e neutra) e usuário (a). Usuários que mais publicaram <i>tweets</i> e <i>retweets</i> com a polaridade positiva (b).	56
3.12	Relacionamento entre as entidades <i>hashtag</i> e usuário (a). <i>Hashtag</i> mais referenciada nos <i>tweets</i> e <i>retweets</i> (b).	56
3.13	Quantidade de usuários, <i>hashtags</i> (a), <i>tweets</i> e <i>retweets</i> (b). <i>Hashtag</i> mais comentada (c). Classificação das mensagens que foram incluídas com essa <i>hashtag</i> em <i>tweets</i> e <i>retweets</i> (d). Usuários que mais enviaram mensagens com essa <i>hashtag</i> (e).	57
3.14	Polaridade dos <i>tweets</i> e <i>retweets</i> (a). Grafo com o relacionamento entre as entidades <i>hashtag</i> e polaridade (b). <i>Histogram</i> com a quantidade de <i>tweets</i> e <i>retweets</i> coletados por dia com essa <i>hashtag</i> (c).	58
3.15	Dados coletados no <i>Twitter</i> entre os dias 15 junho de 2018 e 21 de junho de 2018. Usuários discrepantes.	59

3.16	<i>Dashboard</i> com informações do usuário <i>dobresdelena</i> . Quantidade de <i>hashtags</i> (a), <i>tweets</i> e <i>retweets</i> (b). Usuário discrepante (c). Quantidade de mensagens classificadas como <i>retweets</i> (d). <i>Histogram</i> com a quantidade de <i>retweets</i> coletados por dia (e).....	60
3.17	Informações sobre a polaridade dos <i>retweets</i> do usuário <i>dobresdelena</i> (a). Relação entre as entidades polaridade (positiva) e o usuário (b). Classificação (positiva) de <i>retweets</i> por hora (c).....	60
3.18	<i>Retweets</i> publicados pelo usuário <i>dobresdelena</i>	61
3.19	Análise reversa da imagem nas ferramentas <i>TinEye</i> (a) e <i>Google Images</i> (b)....	62
3.20	Análise através da ferramenta <i>TweetBotOrNot</i>	63
3.21	Dados coletados no <i>Twitter</i> entre os dias 01 abril de 2020 e 06 de setembro de 2020. <i>Histogram</i> com a quantidade de <i>tweets</i> e <i>retweets</i> coletados por dia (a). Idiomas mais utilizados para publicar os <i>tweets</i> e <i>retweets</i> (b). Quantidade de usuários, <i>hashtags</i> (c), <i>tweets</i> e <i>retweets</i> (d). <i>Hashtag</i> mais comentada (e). Classificação das mensagens em <i>tweets</i> e <i>retweets</i> (f). Usuários que mais enviaram mensagens sobre o tema (g).....	64
3.22	Classificação dos <i>tweets</i> e <i>retweets</i> por semana.	65
3.23	Classificação geral dos <i>tweets</i> e <i>retweets</i> conforme o algoritmo <i>Pattern Analyzer</i>	65
3.24	<i>Tweets</i> publicados com os idiomas inglês, português e chinês.	66
3.25	Quantidade de <i>tweets</i> e <i>retweets</i> publicados pelos idiomas inglês, espanhol, português e chinês.	67
3.26	Sentimento dos usuários que publicaram <i>tweets</i> e <i>retweets</i> com o idioma inglês (a). Sentimento dos usuários que publicaram <i>tweets</i> e <i>retweets</i> com o idioma espanhol (b). Sentimento dos usuários que publicaram <i>tweets</i> e <i>retweets</i> com o idioma português (c). Sentimento dos usuários que publicaram <i>tweets</i> e <i>retweets</i> com o idioma chinês (d).....	68
4.1	Estrutura da biblioteca <i>TextBlob</i>	72
5.1	Ataque adversarial através de ruído com erros de ortografia no algoritmo <i>Pattern Analyzer</i>	87
5.2	Representação visual do ataque de inserção em frases com o idioma inglês.	88
5.3	Correção do ataque adversarial através do método <i>correct()</i>	90
5.4	Ataque adversarial através de ruído com erros de ortografia em frases com outros idiomas no algoritmo <i>Pattern Analyzer</i>	91
5.5	Representação visual do ataque de inserção em frases com outros idiomas.	92
5.6	Correção do ataque adversarial através dos métodos <i>correct_language_portuguese()</i> , <i>translate()</i> e <i>correct()</i>	93
5.7	Ataque adversarial através da substituição de palavras por seus sinônimos.	96

5.8	Representação visual do ataque de substituição em frases negativas ou positivas.	97
5.9	Ataque adversarial através da substituição de palavras de negação por seus sinônimos.	99
5.10	Representação visual do ataque de substituição em frases com palavras de negação.	100
5.11	Contramedida para o ataque de substituição em frases positivas e negativas. ...	101
5.12	Contramedida para o ataque de substituição em frases com palavras de negação.	102
6.1	Metodologia de ataque que pode ser aplicada por um usuário mal-intencionado.	103
6.2	Perfil da conta criada no <i>Twitter</i> para publicar os <i>tweets</i> originais e adversariais.	104
6.3	Aplicação dos ataques no ambiente <i>OctopusViz</i>	105
6.4	<i>Tweets</i> original e adversarial publicados pela conta <i>cbxyz</i>	105
6.5	<i>Tweets</i> original e adversarial publicados pela conta <i>cbxyz</i>	106
6.6	<i>Histogram</i> com os <i>tweets</i> publicados pela conta <i>cbxyz</i>	106
6.7	<i>Tweets</i> originais e adversariais publicados pela conta <i>cbxyz</i>	107
6.8	Representação visual dos ataques de inserção e substituição no ambiente <i>OctopusViz</i>	107
6.9	<i>Tweets</i> adversariais publicados pela conta <i>cbxyz</i>	110
6.10	<i>Histogram</i> com os <i>tweets</i> adversariais publicados pela conta <i>cbxyz</i>	110
6.11	<i>Tweets</i> adversariais mitigados pelos métodos de correção, tradução e inserção de palavras de negação no código do classificador.	110

LISTA DE TABELAS

2.4	Trabalhos vulneráveis para os ataques de inserção e substituição.	34
2.5	Diferença entre este trabalho (T) e os trabalhos relacionados.	36
2.6	Diferença entre o <i>OctopusViz</i> e os trabalhos apresentados.	37
3.1	Características do <i>host</i>	41
3.2	Sistemas convidados e suas configurações.	41
3.3	Função para tradução e correção dos <i>tweets</i>	43
3.4	Palavras do <i>corpus stopwords</i> e caracteres especiais.	44
3.5	Função para limpeza dos <i>tweets</i>	44
3.6	Função para tokenização dos <i>tweets</i>	44
3.7	Função para classificação dos <i>tweets</i>	45
3.8	As cinco <i>hashtags</i> mais referenciadas entre os dias 15 junho de 2018 e 31 de julho de 2018.	50
3.9	<i>Hashtags</i> mais referenciadas por <i>tweets</i> ou <i>retweets</i>	50
3.10	Classificação dos <i>tweets</i> e <i>retweets</i> por usuário.	52
3.11	Picos das polaridades dos <i>tweets</i> e <i>retweets</i> por dia.	54
3.12	Polaridade dos <i>tweets</i> e <i>retweets</i>	55
3.13	Quantidade de <i>tweets</i> e <i>retweets</i> por usuário.	59
3.14	Polaridade dos <i>tweets</i> e <i>retweets</i>	65
3.15	<i>Tweets</i> e <i>retweets</i> publicados com os idiomas <i>en</i> , <i>es</i> , <i>pt</i> e <i>zh</i>	67
3.16	Polaridade dos <i>tweets</i> e <i>retweets</i> publicados com o idioma inglês.	68
3.17	Polaridade dos <i>tweets</i> e <i>retweets</i> publicados com o idioma espanhol.	69
3.18	Polaridade dos <i>tweets</i> e <i>retweets</i> publicados com o idioma português.	69
3.19	Polaridade dos <i>tweets</i> e <i>retweets</i> publicados com o idioma chinês.	69
4.1	Classe gramatical e POS <i>Tagger</i> padrão utilizado no conjunto de dados léxico.	73
4.2	Palavras do conjunto de dados léxico.	74
4.3	Relação de algumas palavras que não estão no conjunto de dados léxico.	74
4.4	Entradas da palavra " <i>great</i> " no conjunto de dados léxico.	75
4.5	Cálculo da polaridade e subjetividade da palavra " <i>great</i> " pelo algoritmo <i>Pattern Analyzer</i>	76
4.6	Polaridade e subjetividade das palavras da frase " <i>The world is beautiful and great</i> " no conjunto de dados léxico.	76
4.7	Cálculo da polaridade e subjetividade da frase " <i>The world is beautiful and great</i> " pelo algoritmo <i>Pattern Analyzer</i>	77

4.8	Cálculo da polaridade e subjetividade da frase negativa "Not great" pelo algoritmo <i>Pattern Analyzer</i>	78
4.9	Cálculo da polaridade e subjetividade da frase "Not not never no great" pelo algoritmo <i>Pattern Analyzer</i>	78
4.10	Polaridade, subjetividade e intensidade das palavras "Very", "high" e "action" no conjunto de dados léxico.	79
4.11	Cálculo da polaridade e subjetividade da frase "Very high" pelo algoritmo <i>Pattern Analyzer</i>	80
4.12	Cálculo da polaridade e subjetividade da frase "Very action" pelo algoritmo <i>Pattern Analyzer</i>	80
4.13	Cálculo da polaridade e subjetividade da frase "The world is very beautiful and great" pelo algoritmo <i>Pattern Analyzer</i>	81
4.14	Cálculo da polaridade e subjetividade da frase "Not very great" pelo algoritmo <i>Pattern Analyzer</i>	82
4.15	<i>Emoticons</i> utilizados pelo algoritmo para entrar no cálculo da polaridade.	82
4.16	Cálculo da polaridade e subjetividade da frase "<3 the world" pelo algoritmo <i>Pattern Analyzer</i>	83
4.17	Entradas da palavra "love" no conjunto de dados léxico.	83
4.18	Cálculo da polaridade e subjetividade da frase "love the world" pelo algoritmo <i>Pattern Analyzer</i>	84
5.1	POS <i>Tagger</i> de cada palavra da frase "This house is very nice and elegant".....	86
5.2	Polaridade, subjetividade e intensidade das palavras da frase "This house is very nice and elegant" no conjunto de dados léxico.	86
5.3	Cálculo da polaridade e subjetividade da frase "This house is very nice and elegant" pelo algoritmo <i>Pattern Analyzer</i>	87
5.4	Ataque adversarial através de ruído com erros de ortografia na frase "This house is veery niice and eleggant".	88
5.5	Correção do ataque adversarial através do método <i>correct()</i> da biblioteca <i>TextBlob</i> na frase "This house is veery niice and eleggant".	89
5.6	Ataque adversarial através de ruído com erros de ortografia na frase "Esta casa é mu:into agr/adável e eleg?ante".	91
5.7	POS <i>Tagger</i> de cada palavra da frase "His house is my: into air / adve e leg? Ante".	92
5.8	Correção do ataque adversarial através dos métodos <i>correct_language_portuguese()</i> , <i>translate()</i> e <i>correct()</i> na frase "Esta casa é mu:into agr/adável e eleg?ante".	93
5.9	POS <i>Tagger</i> de cada palavra da frase "The girl is evil".....	94
5.10	Entradas da palavra "evil" no conjunto de dados léxico.	95

5.11	Polaridade, subjetividade e intensidade das palavras da frase " <i>The girl is evil</i> "no conjunto de dados léxico.....	95
5.12	Cálculo da polaridade e subjetividade da frase " <i>The girl is evil</i> "pelo algoritmo <i>Pattern Analyzer</i>	95
5.13	Ataque adversarial através da substituição de palavras na frase " <i>The girl is wicked</i> ".	96
5.14	Cálculo da polaridade e subjetividade da frase " <i>She is charming</i> "pelo algoritmo <i>Pattern Analyzer</i>	97
5.15	Ataque adversarial através da substituição de palavras na frase " <i>She is seductive</i> ".	97
5.16	POS Tagger, polaridade, subjetividade e intensidade das palavras da frase " <i>It's not beautiful what you are doing Alice</i> "no conjunto de dados léxico.....	98
5.17	Cálculo da polaridade e subjetividade da frase " <i>It's not beautiful what you are doing Alice</i> "pelo algoritmo <i>Pattern Analyzer</i>	99
5.18	Ataque adversarial através da substituição de palavras na frase " <i>Nothing beautiful what are you doing Alice</i> ".	100
5.19	Inserção das palavras " <i>wicked</i> "e " <i>seductive</i> "no conjunto de dados léxico.	101
6.1	Cálculo da polaridade e subjetividade do <i>tweet</i> original " <i>A eleição nos Estados Unidos não foi muito agradável</i> "pelo algoritmo <i>Pattern Analyzer</i>	108
6.2	Cálculo da polaridade e subjetividade do <i>tweet</i> adversarial " <i>A eleição nos Estados Unidos n.ã.o foi muito agradável</i> "pelo algoritmo <i>Pattern Analyzer</i>	109

LISTA DE SÍMBOLOS

Cálculo da Polaridade e Subjetividade

- \bar{P} Polaridade de uma palavra do conjunto de dados léxico
 \bar{S} Subjetividade de uma palavra do conjunto de dados léxico

Cálculo da Polaridade e Subjetividade em Frases

- PFA Polaridade de uma frase com advérbios
 $PFNA$ Polaridade de uma frase com palavras negativas e advérbios
 PFN Polaridade de uma frase com palavras negativas
 SFA Subjetividade de uma frase com advérbios
 $SFNA$ Subjetividade de uma frase com palavras negativas e advérbios
 \overline{PF} Polaridade de uma frase
 \overline{SF} Subjetividade de uma frase

LISTA DE ACRÔNIMOS E ABREVIACÕES

AG	Conjunto de Dados com Artigos de Notícias. 32
API	Interface de Programação de Aplicação. 1, 8, 16, 30, 42, 43, 72
AWS	Amazon Web Services. 30
BeR	Classificador Baseado em Regras. 22
BERT	Bidirectional Encoder Representations from Transformers. 32
BOW	Bag Of Words. 13
CNN	Convolutional Neural Network. 32
CPU	Unidade Central de Processamento. 26
DCT	Descoberta de Conhecimento em Texto. 22
DLTU	Deep Learning-based Text Understanding. 30
DMZ	Zona Desmilitarizada. 40, 41
DPI	Deep Packet Inspection. 12
ELK	Elasticsearch, Logstash e Kibana. 9, 10, 12
FIFA	Federação Internacional de Futebol. 47
GAN	Generative Adversarial Networks. 29
IMDB	Internet Movie Database. 30–32
IoT	Internet of Things. 9
ISP	Internet Service Provider. 8
JSON	JavaScript Object Notation. 9
MR	Calgary-Campinas Public Brain MR Dataset. 30
NLP	Processamento de Linguagem Natural. 13, 14, 28, 30, 42, 71
NLTK	Natural Language Toolkit. 17, 18, 42, 43, 71
OMS	Organização Mundial da Saúde. 16, 63
OSINT	Open Source Intelligence. 103

POS	Part-of-Speech. xiv–xvi, 13, 22, 71–76, 78, 79, 83, 86, 91, 92, 94, 98
ROC	Característica de Operação do Receptor. 26, 27
RSS	Rich Site Summary. 22
SVM	Máquinas de Vetores de Suporte. 17, 21, 22, 24–26, 72
TOR	The Onion Router. 114
URL	Uniform Resource Locator. 43
VADER	Valence Aware Dictionary and sEntiment Reasoner. 25, 26
VPN	Virtual Private Network. 7, 8, 37, 40, 42, 69
WSD	Word Sense Disambiguation. 21
XML	Extensible Markup Language. 73

A utilização generalizada das redes sociais oferece novas oportunidades de fonte de informações para órgãos de Governo e empresas. Publicar ou divulgar uma ideia tornou-se prática comum nas redes sociais. A disseminação da opinião e expressão individual através dos diversos canais, leva à formação de bases que são úteis para gerar conhecimento sobre os eventos e mudanças do mundo atual. Monitorar, analisar dados e sentimentos sobre empresas, prever cenários que impactam a opinião pública sobre greves, protestos, *marketing*, ataques cibernéticos, eleições, operações militares e pesquisas de mercado são exemplos de como as informações extraídas das redes sociais podem ser utilizadas para antecipar cenários possíveis dentro de um determinado contexto de interesse de análise. Esse tipo de capacidade, permite aos interessados entenderem um determinado assunto em discussão e levantar possibilidades de desdobramentos sobre uma questão em particular.

Conforme Marques e Vidgal [1], as mídias sociais mostraram através da WEB 2.0 uma nova forma de obtenção de dados e desenvolvimento de aplicações. As pessoas começaram a compartilhar suas experiências e opiniões em grande quantidade de forma *online*. Para Pereira-Kohatsu et al. [2], essas mídias sociais representam sensores no mundo real que podem ser usados para medir o pulso da sociedade. Esta enorme massa de dados, disponíveis para os desenvolvedores de sistemas através de Interfaces de Programação de Aplicativos (API's) ¹, tem despertado grande interesse de pesquisadores e atrai diversos estudos acerca de mineração de dados, análise de sentimentos, visualização, entre outras áreas de pesquisa.

No *Twitter* [3, 4] ², por exemplo, qualquer usuário pode publicar uma mensagem curta (*tweet*) com um comprimento máximo de 280 caracteres. Existe uma linha do tempo pública, que transmite os *tweets* de todos os usuários em todo o mundo como um extenso fluxo de informações em tempo real de mais de um milhão de mensagens por hora, especialmente durante eventos que se tornam significativos dependendo de contextos sociais, econômicos ou políticos. Russel [6], cita a curiosidade humana e a necessidade de compartilhar ideias e experiências, fazer perguntas, interagir de maneira rápida como característica marcante dessa rede social. O *Twitter* propicia de maneira dinâmica que todos esses aspectos sejam possíveis de serem realizados com velocidade impressionante. Além disso, esta rede social possui um diferencial das demais, pelo seu modelo assimétrico de seguidores, onde qualquer usuário pode ficar por dentro dos últimos acontecimentos, mesmo que ele não siga o autor da postagem, enquanto em outras redes sociais, como o *Facebook* e *LinkedIn*, é preciso uma aceitação de conexão entre os seus usuários.

¹Conjunto de rotinas e padrões de programação para acesso a softwares.

²Plataforma de *microblogging* lançada em 2006, com mais de 25 milhões de visitantes únicos mensais [5].

Em razão da rapidez com que as informações são transmitidas na Internet, técnicas de extração de conhecimento são utilizadas para automatizar a busca e processamento de textos que, acompanhadas de técnicas de análise de sentimentos, tornam possível a descoberta do julgamento dos usuários com relação aos produtos, serviços e companhias. Conseqüentemente, as organizações são capazes de realizar melhorias e adotar práticas de acordo com a opinião de seu público-alvo. Fontes como o *Twitter*, por exemplo, que geram grandes volumes de dados a cada momento - também conhecidos como *big data*, tem o potencial de facilitar a pesquisa sobre fenômenos sociais com base na análise de sentimentos [7], assim como a busca por novas soluções que auxiliem na extração de conhecimento útil a partir dessas grandes bases de dados.

Entretanto, uma questão importante que deve ser levada em consideração com relação as abordagens de análise de sentimento (léxica, com técnicas de aprendizado de máquina ou híbrida), aplicadas nestes ambientes de coleta, são as estratégias de ataques adversariais. Neste caso, um usuário mal-intencionado pode estrategicamente manipular a entrada de dados com pequenas perturbações para alterar o resultado da saída do classificador de sentimento [8]. Isso gera para o tomador de decisão a percepção equivocada da realidade, o que pode levar a situações em que decisões erradas serão tomadas, causando danos dos mais diversos tipos, dependendo do negócio. Entende-se que isso é um aspecto importante e que é necessária uma avaliação mais precisa. Esse trabalho faz essa avaliação ao longo dos ataques implementados e suas possibilidades de exploração.

Uma diferença importante nesta pesquisa em comparação com outras formas de ataques adversariais está no fato de que o seu foco é aplicado em algoritmos de classificação que utilizam abordagem léxica. O esforço realizado na engenharia reversa identificou vulnerabilidades no código do classificador e no conjunto de dados léxico. Isso permitiu se ter uma visão clara sobre as vulnerabilidades e dos possíveis ataques adversariais no classificador léxico de linguagem natural. Ao se considerar esses pontos, foram desenvolvidas duas estratégias de ataques adversariais *white-box* (algoritmo e conjunto de dados léxico conhecidos pelo usuário [9]) e algumas contramedidas que podem ser utilizadas para mitigar esses ataques.

1.1 OBJETIVOS

O objetivo principal deste trabalho é identificar e analisar as vulnerabilidades de um classificador de sentimento léxico sob diferentes estratégias de ataques adversariais e propor contramedidas que podem ser utilizadas para mitigar esses ataques. Para isso, os seguintes objetivos específicos podem ser listados:

- Desenvolver uma arquitetura de coleta de informações de redes sociais e implementar

na camada de classificação dessa arquitetura um classificador de sentimento léxico;

- Utilizar técnicas de engenharia reversa para identificar e analisar as vulnerabilidades no conjunto de dados léxico e no código do classificador de sentimento léxico;
- Propor e realizar duas estratégias de ataques adversariais (inserção de caracteres e substituição de palavras) na arquitetura de coleta;
- Propor contramedidas que poderão ser utilizadas para mitigar os efeitos de diferentes estratégias de ataques em aplicações que utilizam o mesmo algoritmo para classificar textos.

1.2 RESUMO DAS PRINCIPAIS CONTRIBUIÇÕES

Este trabalho propõe duas estratégias de ataques adversariais *white-box* em um classificador léxico de linguagem natural. O classificador léxico em estudo nesse trabalho é o algoritmo *Pattern Analyzer* da biblioteca *TextBlob* [10], que é utilizado em vários ambientes para classificar textos. Nesses ambientes, esse tipo de classificador gera a percepção do sentimento dos usuários em diversas aplicações de mídia social.

De maneira resumida, pode-se considerar como contribuições da presente tese:

- A viabilidade de anular os valores da polaridade através da inserção de caracteres nas palavras escritas em frases com o idioma inglês ou outros idiomas;
- A possibilidade de inverter a percepção do classificador através da substituição de palavras por sinônimos que não estão no conjunto de dados léxico (palavras negativas ou positivas) ou que não são utilizados pelo algoritmo *Pattern Analyzer* (palavras de negação);
- A capacidade de mitigar os dois tipos de ataques através das contramedidas propostas.

Para mitigar o ataque de inserção de caracteres utilizamos dois métodos de correção: um para o idioma inglês e outro para o idioma português. Mitigamos o ataque de substituição através da inserção de sinônimos (palavras semelhantes) no conjunto de dados léxico e no código do algoritmo *Pattern Analyzer*. As principais contribuições deste trabalho estão detalhadas na Seção 2.8.

1.3 MOTIVAÇÃO E JUSTIFICATIVA

A revisão bibliográfica deste trabalho aponta que existem diversas aplicações que utilizam técnicas de análise de sentimentos para classificar textos. Entretanto, por suas características, existem também lacunas que fazem com que essas técnicas sejam exploradas por

usuários mal-intencionados na tentativa de identificar vulnerabilidades para realizar ataques adversariais. Assim, é necessário incluir este ponto de vista para uma avaliação mais precisa.

Isso também se torna uma motivação para estudar classificadores de sentimento, em particular, os que utilizam a abordagem léxica. O método implica em identificar vulnerabilidades para realizar estratégias de ataques adversariais e propor contramedidas que poderão ser aplicadas para mitigar esses ataques em aplicações que empregam essas técnicas para fazer análise de sentimento ou classificar textos.

1.4 PUBLICAÇÕES

As publicações relacionadas a esta tese estão divididas em três grupos, a saber: trabalhos associados às tecnologias de processamento e visualização de dados (Revista *Applied Sciences*; Fator de Impacto: 2.474 / A4); trabalhos relacionados a classificadores de sentimento (Revista *Sensors*; Fator de Impacto: 3.275, *Qualis* 2013-2016 / Engenharias IV: A1); e, trabalhos relacionados a estratégias de ataques adversariais.

- Pimenta Rodrigues, Gabriel; de Albuquerque, Robson; Gomes de Deus, Flávio; de Sousa Júnior, Rafael; **de Oliveira Júnior, Gildásio**; García Villaba, Luis; Kim, Tai-Hoon (2017). *Cybersecurity and Network Forensics: Analysis of Malicious Traffic towards a Honeynet with Deep Packet Inspection*. Applied Sciences-Basel JCR, v. 7, p. 1082;
- **de Oliveira Júnior, G.A.**; de Oliveira Albuquerque, R.; Borges de Andrade, C.A.; de Sousa, R.T., Jr.; Sandoval Orozco, A.L.; García Villalba, L.J. *Anonymous Real-Time Analytics Monitoring Solution for Decision Making Supported by Sentiment Analysis*. Sensors 2020, 20, 4557;
- O trabalho sobre as estratégias de ataques adversariais no classificador de sentimento léxico está em processo de revisão (Revista *Computer Communications*; Fator de Impacto: 2.816, *Qualis* 2013-2016 / Engenharias IV: A1).

1.5 ESTRUTURA DO TRABALHO

Este trabalho está organizado da seguinte forma: no Capítulo 2, são apresentados alguns conceitos básicos importantes para a compreensão da proposta da arquitetura de coleta. Além disso, apresenta-se também o estado da arte, onde os trabalhos relacionados são revisados, bem como as principais contribuições da solução proposta e das estratégias de ataques adversariais; a descrição do problema, com as justificativas para as estratégias de ataques adversariais, é apresentada juntamente com os detalhes e eficiência da proposta da arquitetura

de coleta no Capítulo 3; o Capítulo 4 apresenta com detalhes, através de engenharia reversa, o funcionamento do classificador léxico utilizado na arquitetura de coleta; o Capítulo 5, mostra a aplicação de dois modelos de ataques adversariais e algumas contramedidas que poderão ser utilizadas para mitigar esses ataques; o Capítulo 6 mostra os resultados dos ataques adversariais na arquitetura de coleta e por fim, o Capítulo 7 apresenta as considerações finais e propostas de trabalhos futuros.

2 ESTADO DA ARTE E TRABALHOS RELACIONADOS

Neste capítulo são apresentados os conceitos necessários para o desenvolvimento e entendimento do ambiente proposto e das estratégias de ataques adversariais. As Seções 2.1, 2.2, 2.3, 2.4 e 2.5 apresentam a fundamentação teórica sobre análise de dados de mídia social, *bot* social, sistemas de anonimato, sistemas distribuídos e visualização de dados, utilizados pela arquitetura de coleta para auxiliar na apresentação, compreensão e coleta dos dados. A Seção 2.6 apresenta a fundamentação teórica sobre análise de sentimentos (camada de classificação da arquitetura de coleta) dando ênfase a três abordagens de classificação: a) léxica; b) aprendizado de máquina; e c) híbrida. Esta seção mostra também uma comparação entre as abordagens léxica e aprendizado de máquina. A Seção 2.7 aborda as estratégias de ataques adversariais em algoritmos de classificação. A Seção 2.8 mostra as principais contribuições deste trabalho.

2.1 ANÁLISE DE DADOS DE MÍDIA SOCIAL

Profissionais que trabalham com dados têm uma infinidade de ferramentas computacionais disponíveis para auxiliar na coleta, limpeza, análise e apresentação de dados. Exemplos dessas ferramentas, como o Planilhas *Google*, o *Web Scraper*, o *OpenRefine*, o *Infogram*, o *Quadrigam*, o *Google Analytics*, o *Tableau*, o *Gephi* etc., são abundantes. No entanto, de acordo com Brooks [11], incompatibilidades e limitações de *design* podem exigir habilidades especializadas desses profissionais, como realizar adaptações extensas, encontrar soluções frágeis ou mudanças de contexto que possam impedir seu progresso. Isso pode restringir a participação de indivíduos nesta área de pesquisa emergente.

Conforme Heer e Shneiderman [12], a análise requer julgamento humano contextualizado com relação ao significado específico de domínio de *clusters*, tendências e *outliers* descobertos em dados. “Como eles organizam suas informações para análise? Quais ferramentas computacionais eles aplicam? Como eles colaboram com os outros? Quais são seus produtos de análise?” Essas são perguntas que Chin Jr., Kuchar e Wolf [13] apresentam para instigar a pesquisa, desenvolvimento e implantação de tecnologias de informação para apoiar a análise de inteligência.

Alguns pesquisadores realizaram estudos para compreender as práticas de análise de dados em diversos domínios, como a análise de inteligência [13] e análise de dados de mídia social [11, 14, 15]. Neste estudo, especificamente, nos concentramos nas práticas de análise de dados de profissionais que trabalham com dados de mídia social para assessoramento de tomadores de decisões.

Assim, os pesquisadores em dados de mídia social enfrentam algumas barreiras metodológicas e técnicas e questões sobre como a pesquisa com dados sociais *online* deve ser feita, garantindo, por exemplo, validade, ética e reprodutibilidade. Portanto, o *design* de ferramentas de análise de dados é um desafio interdisciplinar que requer a compreensão do domínio no qual o analista de dados trabalha em outros campos técnicos, como mídia e jornalismo [11, 16].

2.2 BOT SOCIAL

Segundo Ferrara et al. [17], um *bot* social é um algoritmo de computador que produz automaticamente conteúdo e interage com os humanos nas mídias sociais, tentando imitar e possivelmente alterar seu comportamento. Os *bots* sociais habitaram as plataformas de mídia social nos últimos anos.

Para Kitzie [18], além de potencialmente colocar em risco a democracia, causar pânico durante emergências e afetar o mercado de ações, os *bots* sociais podem prejudicar nossa sociedade de maneira ainda mais sutis. Um estudo feito por Boshmaf et al. [19] demonstrou a vulnerabilidade dos usuários de mídia social a um *botnet* social projetado para expor informações privadas, como números de telefone e endereços.

Conforme Hwang et al. [20], esse tipo de vulnerabilidade pode ser explorado pelo cibercrime e causar a erosão da confiança nas mídias sociais. *Bots* também podem impedir o avanço da política pública criando a impressão de um movimento de base contrário, ou contribuir para a forte polarização da discussão política observada nas mídias sociais [21]. Eles podem alterar a percepção da influência da mídia social, aumentando artificialmente o público de algumas pessoas, [22] ou podem arruinar a reputação de uma empresa, para fins comerciais ou políticos [23]. Um estudo feito por Kramer et al. [24] demonstrou que as emoções são contagiosas nas redes sociais: *bots* indescritíveis poderiam facilmente se infiltrar em uma população de zumbis inconscientes.

2.3 SISTEMAS DE ANONIMATO

Conforme Edman et al. [25], os sistemas de anonimato fornecem "desassociação" entre o remetente e os destinatários e entre o receptor e remetentes. Os sistemas de anonimato se enquadram em duas classificações: i) alta latência, utilizados em aplicativos baseados em mensagens que toleram atrasos; e ii) baixa latência para aplicativos que trabalham em tempo real [25].

As *Virtual Private Networks* (VPNs) são consideradas sistemas de anonimato de baixa

latência. A utilização de uma VPN transfere a confiança do usuário e do *Internet Service Provider* (ISP) para o provedor de VPN, já que a primeira linha de identificação será a saída da VPN para a Internet. Isso, de certa maneira, proporciona a privacidade [26]. Este aspecto é importante porque não interfere na coleta, mas também gera a garantia de que não é necessário alteração de comportamento por parte dos usuários das redes sociais, trazendo maior realidade no sentimento da publicação.

Segundo Çalişkan et al. [27], as VPNs fornecem também comunicação segura para garantir a confidencialidade dos dados trafegados. Assim, usuários mal-intencionados observarão apenas dados criptografados. A integridade da comunicação é fornecida para garantir que qualquer tipo de adulteração no tráfego seja detectada e descartada [27]. Além disso, alguns provedores de serviço de VPN suportam também alta largura de banda, baixa latência, alto *throughput*, múltiplas conexões simultâneas e pagamentos onde não é necessário a identificação da origem [28].

Outro ponto a ser considerado é que as VPNs resolvem o problema de restrições impostas sobre endereços IP que algumas aplicações de mídias sociais utilizam para limitar coletas. Por exemplo, a API do *Twitter* autoriza apenas para cada usuário, 900 solicitações a cada 15 minutos [29]. Este problema de restrição do *Twitter* pode ser resolvido com a utilização de VPNs (endereços IP de outros países onde essa restrição não se aplica).

2.4 SISTEMAS DISTRIBUÍDOS

Conforme Dhulavvagol et al. [30], os sistemas distribuídos são compostos por vários computadores que interagem entre si em uma rede para realizar operações de processamento de dados. Para Joyce et al. [31], o monitoramento de sistemas distribuídos envolve coleta, interpretação e visualização de informações sobre interações entre processos em execução simultânea. O *Elasticsearch* e *Kibana* [32] podem ser utilizados como mecanismo de um sistema distribuído para trabalhar com processamento de dados em grande escala e operações de pesquisa.

2.4.1 Elasticsearch

O *Elasticsearch* é um projeto *open source* desenvolvido sobre o *Apache Lucene* que pode ser utilizado como mecanismo de busca e análise de dados distribuído [32]. O *Elasticsearch* pode armazenar e indexar todo tipo de dado, desde texto estruturado, não estruturado, dados numéricos ou dados geoespaciais. A velocidade (latência do momento em que um documento indexado fica disponível para busca – normalmente um segundo), escalabilidade e capacidade de indexar muitos tipos de conteúdo faz com que essa plataforma possa ser em-

pregada em vários estudos de caso, a saber: busca em aplicação, *website* e empresarial; *logging* e análise de dados de *log*; métricas de infraestrutura e monitoramento; análise e visualização de dados geoespaciais; e análise de segurança.

Conforme Oliveira Junior et al. [33], a indexação e a busca são basicamente as duas ações executadas pelo *Elasticsearch*. Na indexação os documentos são inseridos, alterados e excluídos. Na busca são utilizadas diversas *features*, como busca por sinônimos, agrupamentos e contagem (*max*, *min* e *avg*) de determinados eventos e expressões lógicas.

O *Elasticsearch* trabalha com fragmentos (*shards*), documentos e índices [32]. Um índice é um agrupamento lógico de um ou mais fragmentos físicos. Os documentos são as estruturas onde eventos e outras informações são estruturadas e armazenadas. Esses documentos são distribuídos em vários fragmentos no índice. Para garantir redundância e aumentar a capacidade de consulta o *Elasticsearch* distribui esses fragmentos em vários nós (*cluster*).

Para Dhulavvagol et al. [30], técnicas de fragmentação são utilizadas para aprimorar os problemas de escalabilidade, consistência e tolerância a falhas. Os autores desenvolveram no *Elasticsearch* um mecanismo para lidar com processamento de dados em grande escala e operações de pesquisa usando uma técnica de fragmentação eficiente. Três conjuntos de dados (*Shakespeare* no formato JSON com 10.000 documentos, 20 grupos de notícias com 20.000 documentos e GOV2) e três algoritmos de seleção de fragmentos (*Rank-S*, *CORI* e *Redde*) foram utilizados nos testes. Conforme os autores, o desempenho do algoritmo *Rank-S* foi melhor em comparação com os algoritmos *CORI* e *Redde*, reduzindo o custo em 28%. Os resultados mostraram que a técnica de fragmentação fornece um mecanismo eficiente para trabalhar com dados em grande escala.

Kononenko et al. [34], cita que a percepção derivada de grandes conjuntos de dados permite resolver problemas difíceis para obter vantagem competitiva. Os autores utilizaram o *Elasticsearch* como *back-end* para construir o DASH, uma ferramenta que fornece informações agregadas sobre o processo de revisão de *patch* para desenvolvedores *Mozilla*. Os resultados mostraram que o *Elasticsearch* é um mecanismo de pesquisa de texto facilmente escalável, capaz de lidar com grandes quantidades de dados *online*. Assim o DASH, implementado com o *Elasticsearch*, mostrou um tempo de resposta notável, permitindo que os desenvolvedores *Mozilla* observassem ao vivo a imagem do processo de revisão do *patch*.

Bajer [35] utilizou a pilha ELK para integrar e processar dados reais de dispositivos IoT instalados no prédio da *ABB Corporate Research* em Cracóvia na Polônia. Os dados coletados pelos dispositivos IoT e indexados no *Elasticsearch* foram usados para analisar e visualizar *insights* significativos sobre as operações de construção. Os resultados mostraram que o *Elasticsearch* apresentou desempenho excelente, mesmo sem qualquer customização, fornecendo alto desempenho com dados disponíveis para visualização.

2.4.2 Kibana

Kibana é uma interface *open source* usada para pesquisar, visualizar e analisar dados armazenados nos índices do *Elasticsearch* [32]. A plataforma possui basicamente quatro componentes principais: *Discover*, *Visualize*, *Dashboard* e *Stack Management* (Figura 2.1). O *Discover* é utilizado para explorar e procurar percepções e relacionamentos ocultos. O componente *Visualize* fornece o *Lens*, uma interface para construção rápida de gráficos, tabelas, métricas e mapas. O *Dashboard* fornece um conjunto de visualizações com *insights* dos dados armazenados nos índices do *Elasticsearch*. O componente *Stack Management* faz o gerenciamento da pilha ELK (índices, *clusters*, licenças, padrões de índice, etc.).

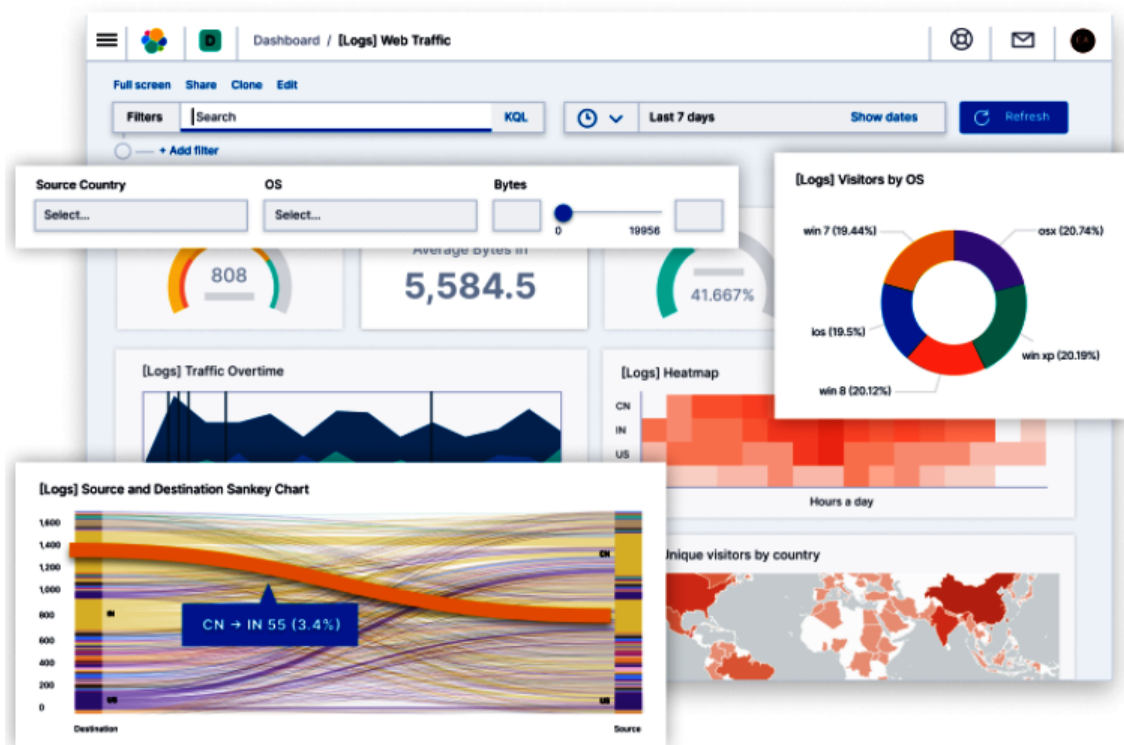


Figura 2.1 – *Dashboard* com conjunto de visualizações [32].

O *Kibana* oferece também a possibilidade de atender demandas específicas através de *plugins* que podem ser customizados conforme necessidade. Esses *plugins* são adicionados no componente *Visualize*.

2.5 VISUALIZAÇÃO DE DADOS

A visualização pode ser definida como a comunicação de informação usando representações gráficas [36]. Ela transforma dados, informações e conhecimento em uma forma na qual o sistema visual humano percebe as informações embutidas nela [37]. Segundo Ward

[36], a visualização é importante porque “somos seres visuais que usam a visão como um dos nossos principais sentidos para a compreensão da informação”. Portanto, o objetivo da visualização é auxiliar a compreensão dos dados, aproveitando a capacidade do sistema visual humano de reconhecer padrões, identificar tendências e identificar *outliers* [38]. Se as visualizações forem elaboradas, elas podem melhorar a compreensão dos dados e dar as pessoas uma impressão imediata e profunda. Em vez de submetermos as pessoas ao processo de leitura com história complexa, podemos ir direto ao ponto [39].

A representação visual avançou com novas formas de coleta, manipulação e interação de dados, misturando-se com outros campos e processos. Neste contexto, a visualização tem sido também útil para cientistas de dados. Segundo Gray et al. [40], na fase de elaboração de relatórios, as visualizações podem ajudar a identificar temas, questões, tendências e desvios incomuns, encontrar exemplos típicos e até mesmo sugerir lacunas e omissões nos relatórios. Além disso, as visualizações também desempenham vários papéis na publicação, pois ilustram um ponto feito em uma história de maneira mais convincente, removem informações técnicas desnecessárias e sugerem transparência aos leitores sobre o processo de geração de relatórios.

De maneira geral, conforme já mencionado nesta seção, a visualização de dados consiste na representação gráfica de informações e dados. Usando elementos visuais, como diagramas, gráficos e mapas, a visualização de dados é uma forma acessível de ver e entender exceções, tendências e padrões, que de outra forma, não seriam percebidos facilmente. As ferramentas de *big data* e as tecnologias de visualização de dados são essenciais para analisar em tempo real enormes quantidades de informações e tomar decisões impulsionadas por dados.

Vários produtos comerciais de rastreamento de mídia social, como o *Hootsuite*¹ e o *Sproutsocial*², concentram-se em oferecer estatísticas de interação com o cliente. Eles rastreiam métricas relacionadas ao número de respostas às mensagens de uma empresa, tentam definir a duração da resposta mais bem-sucedida e outros tipos de estatísticas agregadas de atividade diária. Embora essa informação seja valiosa de uma perspectiva de *marketing online*, isso não ajuda as empresas a entender as expressões de seus clientes no nível do local. Outros sistemas, como o *Brandwatch*³ e o *Zignal Labs*⁴, concentram-se na identificação de eventos em tempo real, por exemplo, identificando clientes insatisfeitos. Novamente, as informações no nível do local não são suportadas e os analistas precisam fornecer palavras-chave de interesse ao sistema.

Diversos sistemas de pesquisa foram desenvolvidos para mineração de dados do *Twitter*.

¹<https://hootsuite.com/>

²<https://sproutsocial.com/>

³<https://www.brandwatch.com/>

⁴<https://zignallabs.com>

Diakopoulos et al. [14] desenvolveu uma ferramenta (*Vox Civitas*) para mineração de eventos atuais, que visava apoiar jornalistas durante a extração de notícias dos dados agregados da mídia social *Twitter*. A interface do usuário foi projetada especificamente para permitir a investigação jornalística de respostas em tempo real a eventos de notícias. Da mesma forma, Marcus et al. [41] afirma que o *TwitInfo* permite que os usuários explorem eventos em tempo real que ocorrem no *Twitter*. Ambos os sistemas utilizam linhas de tempo para extrair elementos notáveis baseados em picos de volume de *tweets* e heurísticas baseadas em frequência de palavras. Esses dois sistemas contêm elementos potencialmente valiosos para um analista de negócios. No entanto, nenhum desses sistemas considera a origem geográfica das mensagens, perdendo, assim, um nível substancial de contexto que poderia ser usado para coletar inteligência de negócios.

A mídia social oferece oportunidades potenciais para as empresas extraírem inteligência de negócios. Sijtsma et al. [42] apresenta o *Tweetviz*, uma ferramenta interativa para ajudar empresas a extrair informações acionáveis de um grande conjunto de mensagens ruidosas do *Twitter*. O *Tweetviz* visualiza o sentimento do *tweet* dos locais da empresa, identifica outros locais de negócios que os usuários do *Twitter* visitam e estima alguns dados demográficos simples dos usuários do *Twitter* que frequentam uma empresa. Um estudo de caso para avaliar a capacidade do sistema indica que o *Tweetviz* pode fornecer uma visão geral dos problemas e negócios de uma empresa, bem como informações que ajudam os usuários a criar perfis de clientes. O objetivo dessa pesquisa [42] é alavancar as informações geográficas para fornecer informações específicas de localização acionáveis.

Oliveira Junior et al. [33] apresenta um ambiente denominado *HoneySELK* para pesquisa e visualização de ataques cibernéticos em tempo real. O *HoneySELK* utiliza a pilha ELK para realizar o armazenamento distribuído da estrutura completa dos dados do monitoramento em tempo real dos ataques, com dados de georreferenciamentos, estatísticas e grafos, indicando relacionamentos diversos, que auxiliam na identificação e *modus operandi* de atacantes.

Pimenta Rodrigues et al. [43], aplica técnicas de *Deep Packet Inspection* (DPI) para detectar anomalias e avaliar diferentes ataques no tráfego de rede destinado a uma *Honeynet* de Alta Interatividade. Com base nos dados coletados e através da Pilha ELK foi possível gerar estatísticas de usuários, serviços, senhas utilizadas e distribuição de endereços IP.

2.6 ANÁLISE DE SENTIMENTOS NA CLASSIFICAÇÃO DE TEXTOS

O crescimento do interesse na área de análise de sentimentos permite explorar as visões ou textos presentes em diferentes plataformas. A análise de sentimento pode ser aplicada em plataformas de mídias sociais, páginas de jornais ou revistas, *blogs*, páginas de influenciadores digitais, ou em páginas específicas sob assuntos de interesse. Conforme Biswas et al.

[44], o tipo de análise de sentimento depende do conjunto de dados e da abordagem de raciocínio adotada. O sentimento pode ser binário ou com várias classes envolvidas (positiva, negativa ou neutra).

Além disso, por meio de técnicas de aprendizado de máquina [45], abordagem léxica [46] ou abordagem híbrida [47], é possível estimar determinadas características do objeto de análise, gerando *insights* úteis para análise aprofundada [48].

Um sentimento pode ser representado de forma sutil ou complexa em um texto. A mistura de informações objetivas e subjetivas sobre um determinado tema pode gerar ruídos (palavras de parada, *emojis*, *emoticons*, ironias, etc.), os quais, são comumente encontrados na maioria dos conjuntos de dados disponíveis, tornando necessária a limpeza ou modificação destes com técnicas específicas [5]. Assim, a tarefa de reconhecimento automático de sentimentos nos textos se torna mais complexa.

Para Ravi et al. [49], os dados adquiridos precisam passar por várias etapas de pré-processamento (*tokenização*, remoção de palavras de parada, *stemming* e *parts of speech* (POS) tagger) antes de iniciar uma análise completa. Além disso, devido ao ruído extremo nos textos, a etapa de pré-processamento requer também um nível extremo de extração de recursos. Para Kumar et al. [50] o processo de transformação e filtragem dos dados deve ser aplicado para remoção dos ruídos, a fim de melhorar a precisão e acelerar o processo de classificação dos textos. É importante destacar também que a análise de sentimentos trata de um problema de classificação que pode ser utilizada para classificar textos de acordo com sua polaridade, independentemente da frase denotar algum sentimento [51].

2.6.1 Classificação com Abordagem Léxica

A abordagem léxica permite classificar textos considerando um dicionário. Este dicionário contém uma coleção de palavras com opiniões e valores de polaridades que podem ser classificadas em positivas ou negativas. Conforme Araque et al. [52], o sentimento léxico é utilizado para estimar a polaridade através da correspondência de palavras contidas em um texto associadas com suas polaridades de sentimento.

Madhu [53] propôs um método de análise de sentimento que utiliza os conceitos de *Bag Of Words* (BOW), *Part of Speech* (POS) e Processamento de Linguagem Natural (NLP) para analisar tendências suicidas em *blogs* e *tweets* com a biblioteca *TextBlob*. Neste trabalho, a autora utiliza, mas não cita, que a biblioteca *TextBlob* emprega como padrão o algoritmo *Pattern Analyzer* (abordagem léxica) para analisar e classificar os textos [10]. Além disso, não aparece no texto o *corpus* que foi utilizado pelo algoritmo para calcular os valores da polaridade. Esse processo deveria ficar mais evidente no trabalho, porque pode prejudicar o entendimento do método correto que foi aplicado para classificar os textos.

Farooqui e Ritika [54] constroem um *Framework* com cinco módulos (coleta, armazenamento, processamento, cálculo da polaridade e visualização) que utiliza a biblioteca *TextBlob* para atribuir e verificar a pontuação de sentimento de *tweets* na forma de *hashtags* que representam opiniões sobre partidos políticos. A Figura 2.2, mostra a estrutura do *Framework* utilizado para a análise de sentimentos com seus módulos: coleta, armazenamento, processamento, cálculo da polaridade e visualização. Durante os testes os autores observaram que a técnica de Processamento de Linguagem Natural (NLP) é um método melhor para análise de sentimento em comparação aos métodos tradicionais. Em nenhum momento, foi mencionado que a abordagem léxica foi utilizada neste trabalho. Mostraram apenas que a biblioteca *TextBlob* possui recursos de processamento para calcular a polaridade dos *tweets*.

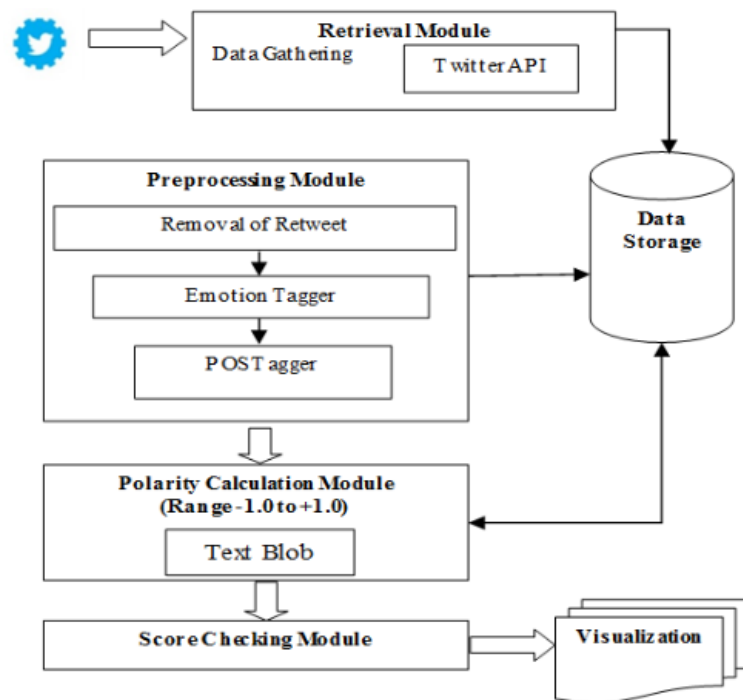


Figura 2.2 – *Framework* utilizado para realizar a análise de sentimento [54].

Patil et al. [55] utiliza a plataforma *Twitter* para extrair opiniões e sentimentos dos usuários sobre os seguintes tópicos: revogação do artigo 370 e seu impacto sobre o comércio no Paquistão, o terrorismo na Índia, na Caximira e no Paquistão. Os autores dividiram o trabalho em quatro partes: coleta dos dados, pré-processamento, análise de sentimento e representação dos dados. A Figura 2.3 representa o fluxo desse processo (inserção de uma palavra-chave, processamento e visualização dos dados). A biblioteca *TextBlob* foi utilizada para processar os *tweets* e realizar a análise de sentimentos. Os resultados mostraram que o Paquistão estava mais preocupado com o impacto no comércio, enquanto a Índia, por outro lado, estava mais preocupada sobre o aumento no terrorismo. Os autores não citaram no trabalho, qual o *corpus* e o algoritmo da biblioteca *TextBlob* foi utilizado para classificar o *tweets*. Citam apenas que o *TextBlob* retorna um valor de polaridade dentro do intervalo [-1.0,

+1.0]. Na leitura foi possível verificar que o método aplicado utilizou a abordagem léxica. Neste caso, foi empregado o algoritmo *default* (*Pattern Analyzer*) da biblioteca *TextBlob*.

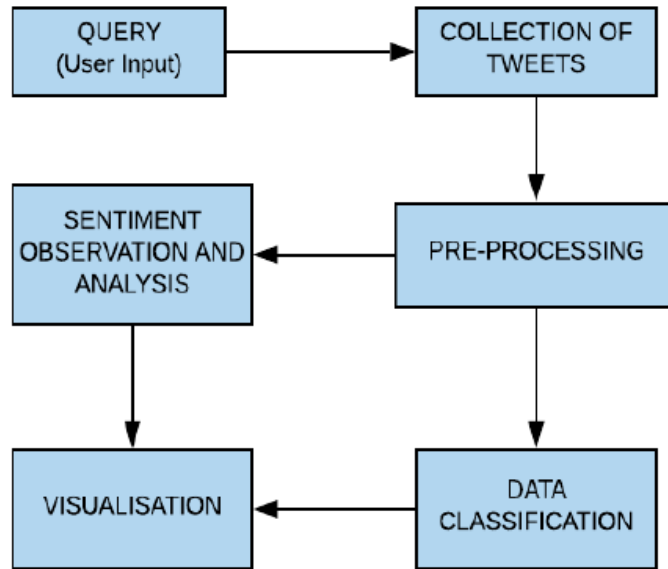


Figura 2.3 – Fluxograma com a metodologia proposta [55].

Ahmed et al. [56] investiga nos *tweets* a perspectiva de sentimento e emoção das pessoas sobre a reabertura nos EUA durante a pandemia do COVID-19. A biblioteca *TextBlob* foi utilizada pelos autores para encontrar o sentimento dos usuários nos *tweets* através da polaridade e subjetividade. Os testes mostraram que as pessoas estavam com emoção dominante de medo quando os estados dos EUA fizeram o bloqueio em março. Entretanto, depois da reabertura, as pessoas ficaram com menos medo, apresentando sentimento menos negativo, mesmo com o aumento dos casos positivos em comparação com a situação de bloqueio. Assim como em [55], não foi possível identificar o *corpus* e o algoritmo da biblioteca *TextBlob*. Os autores acrescentaram apenas que foi utilizada uma biblioteca *Python TextBlob* para realizar análise de sentimentos (pontuações de polaridade e subjetividade).

Biswas et al. [44] propõe um método para prever a reação dos investidores com relação às notícias e desenvolvimento da pandemia do coronavírus. A linguagem de programação *Python* e a biblioteca *TextBlob* foram utilizadas para prever as tendências de mercado após agregar *tweets* sobre o tema coronavírus. Conforme os autores, o modelo proposto (Figura 2.4) foi capaz de calcular as tendências do mercado de ações com previsões correspondentes aos movimentos reais do SENSEX (principal índice da bolsa de valores de Bombaim). O autor menciona apenas uma vez que as notícias são classificadas através do algoritmo *Naive Bayes*, mas isso não fica claro. O *Naive Bayes* não é o algoritmo *default* da biblioteca *TextBlob* [10]. Não aparece também no trabalho qual foi o *corpus* utilizado pela biblioteca *TextBlob* para classificar os textos. Nessa biblioteca, o *Naive Bayes* é treinado em um *corpus movie reviews* [57, 58] (2.000 resenhas de filmes rotuladas com relação a polaridade global do sentimento).

Os autores mostram apenas que a biblioteca *TextBlob* possui uma função de sentimento que calcula os valores da polaridade [+1, -1] e subjetividade [0, 1].

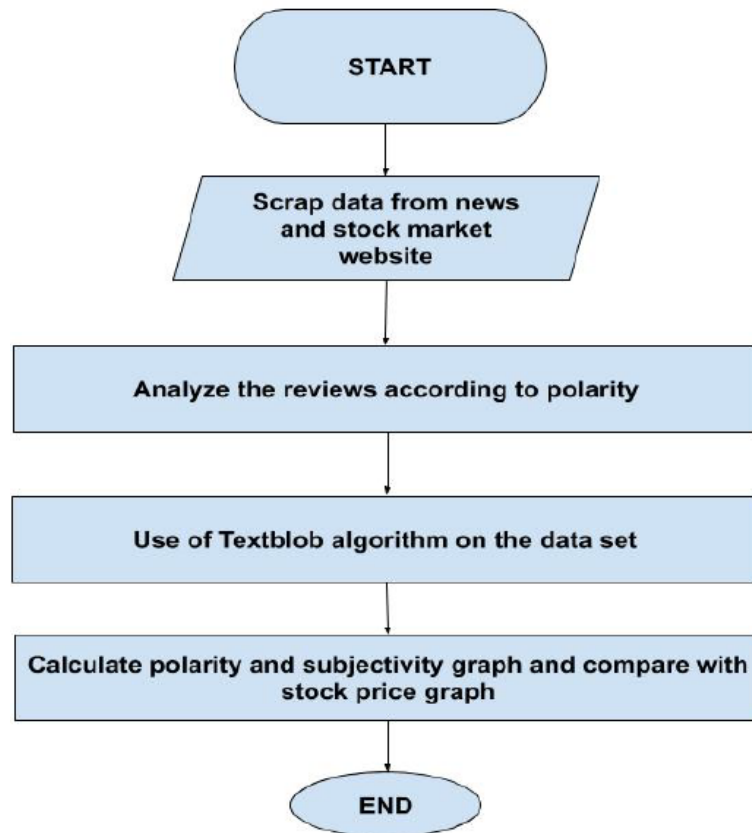


Figura 2.4 – Fluxograma com os componentes do algoritmo proposto [44].

Rajput et al. [59] apresenta dois estudos empíricos sobre as mensagens no *Twitter* relacionadas a pandemia coronavírus. Um estudo foi feito sobre a frequência de palavras e sentimentos dos *tweets*. A análise de sentimento foi aplicada nos *tweets* publicados pelo público em geral e pela OMS (Organização Mundial da Saúde) para entender as atitudes dos usuários sobre a pandemia. Os valores das polaridades dos *tweets* foram calculados com a biblioteca *TextBlob*. Os resultados mostraram que 60% das mensagens da OMS e 29% das mensagens do público em geral foram classificadas como positivas. Apenas 15% das mensagens tiveram sentimento negativo. Assim como [53, 54, 55, 56], não foi mencionado também pelos autores neste trabalho o *corpus*, o algoritmo e a abordagem utilizada para calcular a polaridade dos textos. Outro ponto a ser questionado tem a ver com o idioma (inglês) do *corpus* utilizado pela biblioteca *TextBlob*. Neste caso, deve-se aplicar métodos de tradução para corrigir os *tweets* postados com outros idiomas antes de serem submetidos para o algoritmo de classificação.

Kaur e Sharma [60], analisaram 2.058 *tweets* de diferentes países para entender o sentimento dos usuários em relação a doença COVID-19. Para coletar os dados, os autores utilizaram a biblioteca *Tweepy* e a API do *Twitter*. O pré-processamento e o trabalho de aná-

lise de sentimentos no conjunto de dados coletados (*tweets*) foram feitos pelas bibliotecas *Natural Language Toolkit* (NLTK) e *TextBlob*. Nos testes 24% dos *tweets* foram classificados como positivos e 32,1% como negativos. Conforme os autores, uma porcentagem muito alta dos *tweets* tiveram o sentimento neutro (43%). Neste trabalho, assim como [59], não foi implementado na fase de pré-processamento os métodos de correção e tradução dos *tweets* para o idioma inglês. Esse pode ter sido um problema da porcentagem muito alta de *tweets* de diferentes países serem classificados como neutros. Além disso, não foi identificado no texto o algoritmo utilizado para classificar os *tweets*.

Em outra perspectiva, de Oliveira Júnior et al. [61] propõem o *OctopusViz*, um *Framework* que monitora e coleta *tweets* em tempo real de forma anônima e *online* para processar, pesquisar, visualizar e classificar automaticamente os sentimentos das mensagens em três categorias distintas: positiva, negativa e neutra. O algoritmo *Pattern Analyzer* da biblioteca *TextBlob* foi implementado na camada de classificação para realizar o trabalho de análise de sentimento. Para validar a solução, foram coletados 730.850 *tweets* sobre a seleção brasileira durante a Copa do Mundo FIFA 2018. Os resultados mostraram que durante todo o período da coleta a quantidade de *retweets* publicados foi maior em todas as polaridades (positiva, negativa e neutra). A análise de sentimentos apontou através do algoritmo *Pattern Analyzer* que 36,05% (263.485) dos usuários foram favoráveis a seleção brasileira, 20,04% (146.445) contra e 43,91% (320.920) neutros. Neste trabalho, os autores deixaram claro qual foi o *corpus* (*en-sentiment.xml*) e a abordagem (léxica) utilizada para classificar os textos. Além disso, foi possível verificar também que os métodos de tradução e correção foram aplicados na camada de pré-processamento para melhorar o processo de classificação dos *tweets*. Um aspecto interessante neste trabalho [61] é que esses métodos utilizados são importantes porque podem evitar também possíveis estratégias de ataques adversariais. Este trabalho será detalhado no Capítulo 3.

2.6.2 Classificação com Técnicas de Aprendizado de Máquina

A abordagem com técnicas de aprendizado de máquina utiliza um conjunto de treinamento e um conjunto de teste para desenvolver um classificador de sentimentos [62]. O conjunto de treinamento é utilizado para desenvolver o modelo de classificação e o conjunto de teste para validar a precisão do desempenho desse modelo [63]. Várias técnicas de aprendizado de máquina como *Naive Bayes*, Máquinas de Vetores de Suporte (SVM), Regressão Logística e Árvores de Decisão são utilizadas para classificar textos.

Cerón-Guzmán e León-Guzmán [64], coletaram um conjunto de dados relacionado aos *tweets* da eleição presidencial colombiana de 2014 e uma técnica de aprendizagem supervisionada foi implementada em uma coleção rotulada de usuários para distinguir as contas de *spammer* das não-*spammer*. Eles desenvolveram e aplicaram um sistema de análise de

sentimentos com o objetivo de investigar o potencial das mídias sociais para inferência de intenção de voto. De acordo com os resultados experimentais, os métodos de inferência baseados em dados do *Twitter* não são consistentes, apesar do método de inferência proposto obter erro absoluto médio mais baixo e classificar corretamente os candidatos com maior número de votos no primeiro turno das eleições.

Para a eleição presidencial dos EUA em 2016, muitas pessoas expressaram seus gostos ou desgostos de um determinado candidato presidencial. O objetivo de Joyce e Deng [65] era calcular o sentimento expresso por esses *tweets* e, em seguida, comparar esse sentimento com os dados de pesquisa para ver a correlação que eles compartilhavam. Mesmo que o *Naive Bayes Machine Learning Algorithm* tenha parecido identificar o sentimento associado a determinadas *hashtags*, ele não superou a análise baseada em léxico⁵ como previsto quando comparado com os dados de pesquisa do Trump. No entanto, os *tweets* marcados automaticamente superaram os *tweets* marcados manualmente para ambos os candidatos e teve melhor precisão quando comparados com a análise do léxico de Hillary Clinton. Assim, o método automático economiza horas de trabalho, melhora a precisão e elimina qualquer possível viés que possa ocorrer quando os *tweets* são rotulados manualmente. O coeficiente de correlação muito alto com os *tweets* do Trump sugere que o *Twitter* está se tornando uma plataforma maior e mais diversificada que está começando a rivalizar com técnicas sofisticadas de pesquisa eleitoral ou de intenção de voto. Talvez no futuro, as pesquisas de mídia social se tornem mais incorporadas aos esquemas de votação.

Em outra perspectiva, Micu et al. [67], aplica técnicas de análise de sentimentos para analisar o comportamento *on-line* de clientes em termos de gosto, classificação e revisão de restaurantes. Neste trabalho, os autores utilizaram as bibliotecas NLTK e *TextBlob* para analisar o sentimento dos usuários e mostrar como essa técnica pode ajudar profissionais de *marketing* na interpretação do comportamento do cliente para destacar estratégias de pré-venda, vendas e pós-vendas. Os resultados mostraram que, diferente da localização dos clientes, o sexo não influenciava na classificação dos restaurantes. O estudo mostrou ainda que clientes que viviam em um estado diferente do restaurante atribuíam valores mais baixos na classificação do que os clientes que viviam no mesmo estado. Os autores não deixaram claro no trabalho, qual foi o algoritmo (*Pattern Analyzer* ou *Naive Bayes Analyzer*) da biblioteca *TextBlob* utilizado para classificar os textos. Eles citam apenas que *Naive Bayes*, NLTK e *TextBlob* foram as ferramentas utilizadas para realizar análise de sentimentos. Assim, não fica entendível qual foi a abordagem aplicada para classificar os textos. Além disso, nenhum método de pré-processamento foi aplicado para tratar dos dados.

Conforme Singh et al. [68] a análise de sentimentos é conhecida também como mineração de opiniões porque extrai essas características do *Twitter*. O autor utilizou o *Twython*,

⁵Método que incorpora parte da marcação da fala e utiliza um dicionário de palavras com pontuações semânticas atribuídas a elas para calcular a polaridade final de um *tweet* [66].

uma biblioteca *Python*, para coletar em 2017, dados das eleições da Assembleia Legislativa de Punjab, um Estado da Índia. O conjunto de dados coletados pela ferramenta foi utilizado para realizar a análise de sentimentos sobre os partidos políticos BJP, INC e AAP. Uma implementação do algoritmo *Naive Bayes* da biblioteca *TextBlob* foi utilizada para classificar os *tweets*. Entretanto, o autor não deixou claro como treinou o algoritmo *Naive Bayes*. Segundo [10], o classificador *Naive Bayes Analyzer* da biblioteca *TextBlob* é treinado em um *corpus movie reviews* [57, 58].

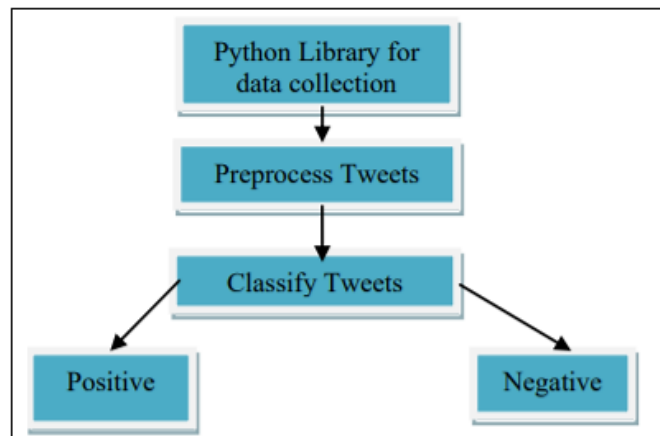


Figura 2.5 – Arquitetura para coletar, processar e classificar os *tweets* [68].

Pokharel [69] concentra estudos na plataforma *Twitter* para identificar o sentimento dos cidadãos nepalenses em relação ao surto da pandemia do coronavírus. A coleta foi feita entre 21 de maio de 2020 e 31 de maio de 2020 através das *hashtags* *#COVID-19* e *#coronavirus*. As bibliotecas *Tweepy* e *TextBlob* foram utilizadas para realizar a coleta e a análise de sentimentos com base nos dados que continham informações dos usuários que compartilhavam a localização do Nepal. A Figura 2.6 representa os métodos deste trabalho (coleta, filtro, limpeza, polaridade e classificação dos dados). Os resultados mostraram que embora a maioria da população do Nepal esteja adotando uma abordagem mais positiva e esperançosa, ainda há casos de medo, tristeza e nojo. Neste trabalho, não foi apresentado o *corpus* utilizado para treinar o algoritmo *Naive Bayes*. Fica claro apenas, que o autor utilizou o algoritmo *Naive Bayes* da biblioteca *TextBlob*. Além disso, o *dataset* coletado no *Twitter* teve um espaço de tempo muito pequeno para fazer os testes (apenas 615 *tweets* em 11 dias). No trabalho foi possível verificar também que foram analisados apenas os *tweets* com idioma inglês. Essa limitação poderia ser resolvida acrescentando um método de tradução na fase de pré-processamento para que os *tweets* escritos com o idioma nepalês fossem traduzidos para o idioma inglês.

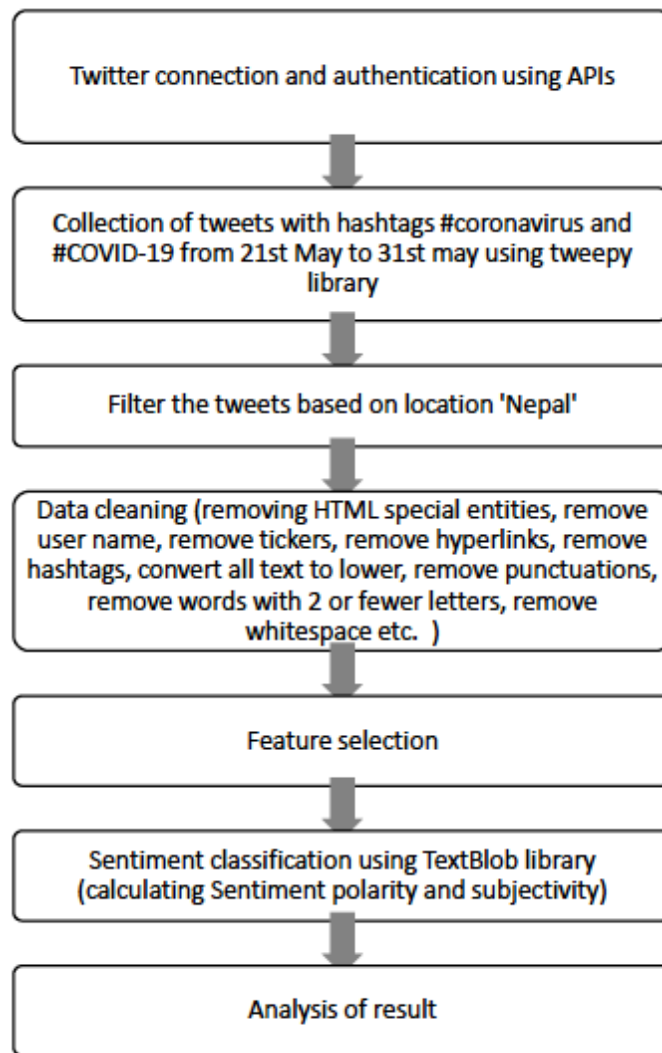


Figura 2.6 – Método aplicado para análise dos dados [69].

Manguri et al. [70], utiliza a biblioteca *Tweepy* para extrair aproximadamente 500.000 *tweets* relacionados a pandemia COVID-19. Os dados coletados foram submetidos para serem utilizados pela biblioteca *TextBlob* em tarefas de análise de sentimentos. A Figura 2.7 representa o procedimento que foi aplicado pelos autores. Os testes mostraram que 36% das pessoas estavam otimistas com relação a pandemia COVID-19. Apenas 14% dos *tweets* foram classificados como negativos. Além disso, grande parte dos *tweets* foram mais objetivos (64%) do que subjetivos (22%). Cabe observar neste trabalho, que o *corpus movie reviews* utilizado para treinar o algoritmo *Naive Bayes* da biblioteca *TextBlob* está escrito em inglês [57, 58]. O trabalho de análise de sentimento pode ter sido prejudicado, tendo em vista, os dados coletados (*hashtags #COVID-19 e #coronavirus*) não terem sido separados por idioma ou traduzidos para o idioma inglês na fase de pré-processamento.

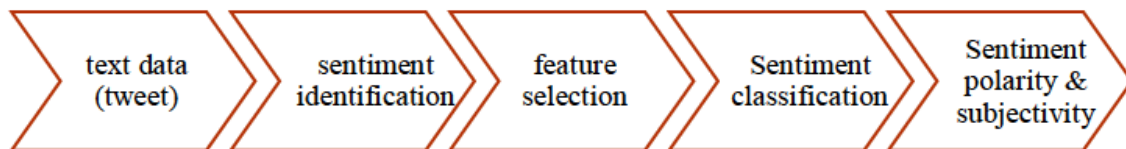


Figura 2.7 – Procedimento para realizar análise de sentimentos [70].

2.6.3 Classificação com Abordagem Híbrida

Apesar do uso de várias técnicas de aprendizado de máquina e ferramentas para análise de sentimentos, há uma necessidade de uma abordagem diferente para melhorar a eficácia da classificação. A abordagem híbrida utiliza como método a combinação de dois ou mais classificadores de sentimento. Conforme Gupta e Joshi [71], a abordagem híbrida explora métodos estatísticos e métodos baseados no conhecimento para detecção de polaridade. Ela aproveita a alta precisão dos algoritmos de aprendizado de máquina com a estabilidade da abordagem léxica [72].

Para lidar com esses desafios, Hasan et al. [73] busca contribuir com o campo incluindo a adoção de uma abordagem híbrida, envolvendo três analisadores de sentimentos diferentes: *SentiWordNet*, *Word Sense Disambiguation (WSD)* e *TextBlob* para calcular a polaridade e a subjetividade dos *tweets*; e dois classificadores de aprendizado de máquina: *Naive Bayes* e Máquinas de Vetores de Suporte (SVM) para classificar as opiniões dos usuários no *Twitter*. As Figuras 2.8(a) e 2.8(b) representam a estrutura responsável pelo processo de coleta, pré-processamento, cálculo da polaridade/subjetividade e classificação das opiniões dos usuários no *Twitter*. Conforme os autores, *TextBlob* e *WSD* foram melhores em comparação com o *SentiWordNet* quando utilizados com o classificador SVM para prever sentimentos eleitorais. O *WSD* teve uma maior taxa de precisão para prever sentimentos quando aplicado com o classificador *Naive Bayes*. Os autores poderiam implementar um método de correção (corrigir palavras com erros de ortografia) para melhorar o processo de rotulação e classificação dos *tweets*.

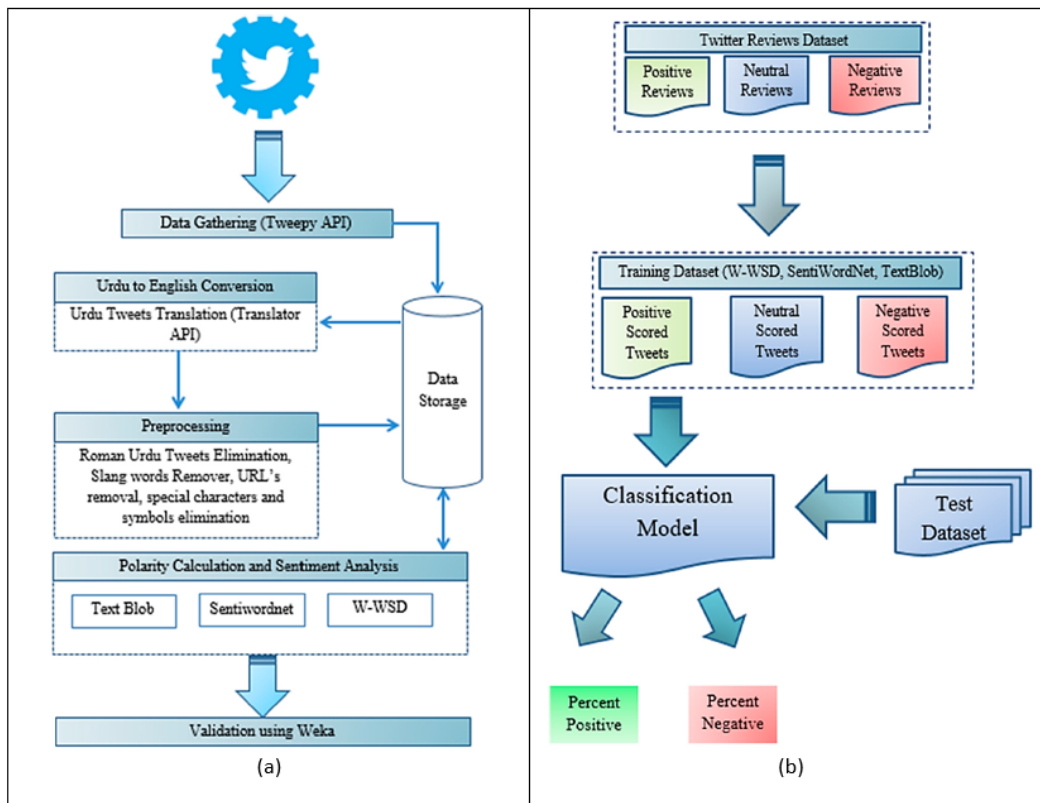


Figura 2.8 – Estrutura de análise de sentimento (adaptado de [73]).

Gomes et al. [51] aplica a mineração de textos em seu trabalho na busca de extrair conhecimento acerca de notícias (*feeds* RSS) da economia de Portugal. O autor utiliza a metodologia de Descoberta de Conhecimento em Texto (DCT) e propõe um modelo de análise de sentimentos (classificador híbrido através da combinação dos modelos BeR e *Naive Bayes*) que polariza as notícias em positivas, negativas ou neutras, além de fornecer um documento com procedimentos para que as organizações extraíam conhecimento de dados textuais. Assim, o autor visita sítios com informações sobre a economia de seu país para representar o sentimento expresso e analisar os textos publicados.

Saha et al. [74] aplica vários métodos para detectar e analisar o sarcasmo no *Twitter*. A finalidade do trabalho foi identificar a opinião pública sobre as tendências e eventos recentes. A biblioteca *TextBlob* foi aplicada no pré-processamento (*tokenização*, *POS tagger* e *stemming*) dos dados. A plataforma *RapidMiner* e a biblioteca *TextBlob* foram utilizadas para encontrar a polaridade e subjetividade dos dados. A ferramenta *Weka* foi usada para calcular a precisão dos *tweets* com base nos classificadores *Naive Bayes* e SVM. Os resultados mostraram que o classificador *Naive Bayes* (65,2%) teve uma acurácia melhor quando comparado com SVM (60,1%). Nas Figuras 2.9(a) e 2.9(b) é possível verificar o mecanismo de detecção de sarcasmo e as etapas envolvidas na abordagem proposta. Neste trabalho, os autores retiram na fase de pré-processamento, os *tweets* e *retweets* que foram publicados com o idioma diferente do inglês. Assim, apenas 2.250 *tweets* foram utilizados para o cálculo da

acurácia. Esses *tweets* removidos poderiam ser utilizados se fosse aplicado os métodos de tradução e correção.

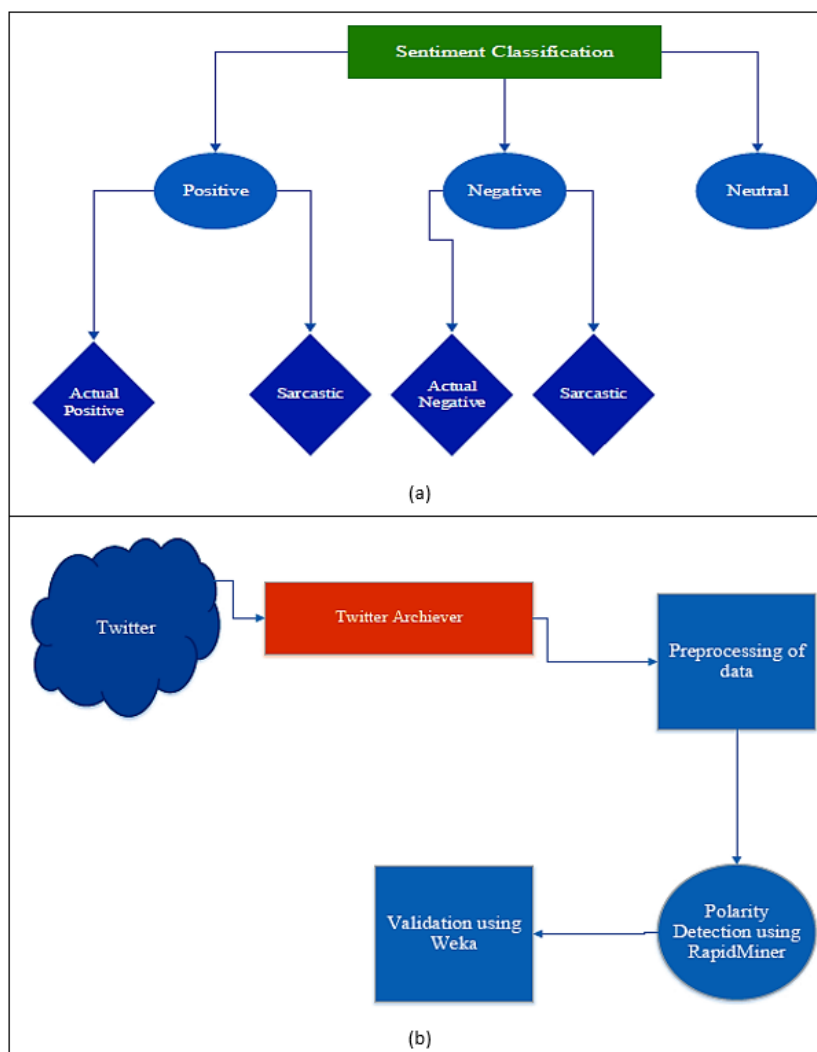


Figura 2.9 – Mecanismo de detecção de sarcasmo e etapas envolvidas na abordagem proposta (adaptado de [74]).

Kunal et al. [75] propõe o uso das bibliotecas *Python Tweepy* e *TextBlob* para acessar, calcular a polaridade, rotular e classificar os *tweets* usando o algoritmo *Naive Bayes*. Essa proposta destina-se a facilitar o processo de análise, sumarização e classificação de *tweets*. Apesar de não fornecer mecanismos de visualização, o trabalho fornece análise de sentimentos em tempo real de qualquer comunidade, governo, religião, celebridades ou políticos em todo o mundo a qualquer momento. Os autores usaram a plataforma *RapidMiner* para comparar o *Decision Tree* e o *Naive Bayes* com um conjunto de dados estáticos “Titanic” disponível no *RapidMiner*. O fluxograma com as técnicas e os módulos de trabalho para realizar a análise de sentimentos pode ser observado na Figura 2.10. O *Naive Bayes* obteve acurácia de 92,58% e o *Decision Tree* obteve apenas 79,04%. Com esses resultados comparativos, o *Naive Bayes* foi a melhor opção para classificação no caso apresentado. No

entanto, neste trabalho não foi citado pelos autores, que além de utilizar uma abordagem híbrida, antes do processo final de análise de sentimentos pelo algoritmo *Naive Bayes*, os *tweets* foram classificados e rotulados (polaridades positivas, negativas ou neutras) através de uma abordagem léxica pelo algoritmo *Pattern Analyzer* da biblioteca *TextBlob*.

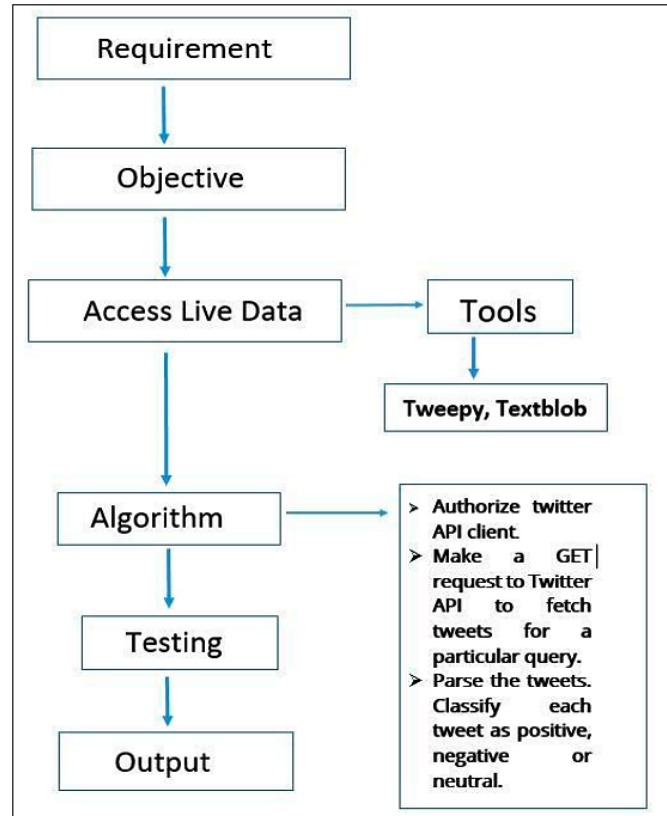


Figura 2.10 – Fluxograma de análise de sentimentos (adaptado de [75]).

Em Praciano et al. [76], é proposta uma estrutura para análise de tendências espaço-temporais das eleições presidenciais brasileiras com base na análise de sentimentos em dados do *Twitter*. A estrutura proposta foi dividida pelos autores nos seguintes blocos funcionais (Figuras 2.11(a) e 2.11(b)): (*Crawling* (extração dos dados do *Twitter*); Pré-Processamento dos dados (limpeza dos *tweets*); Classificação de sentimentos (rotulação das polaridades positiva, negativa ou neutra através das bibliotecas *TextBlob* e *OpLexicon* com *Sentilex*, e classificação dos *tweets* através dos algoritmos SVM, *Naive Bayes*, Regressão Logística e Árvore de Decisão); e Visualização dos dados. Resultados experimentais mostraram que o *framework* proposto apresentou boa eficácia na previsão de resultados eleitorais, bem como na disponibilização de *timestamp* de geolocalização e *tweet*, com uma precisão próxima a 90% quando o algoritmo *Support Vector Machine* (SVM) é aplicado para classificação de sentimento. Os autores citaram no trabalho que o processo de tradução pode produzir erros durante a fase de classificação com a biblioteca *TextBlob*. Esse problema poderia ser melhorado com a aplicação de um método de correção (idioma inglês) depois do método de tradução na fase de pré-processamento.

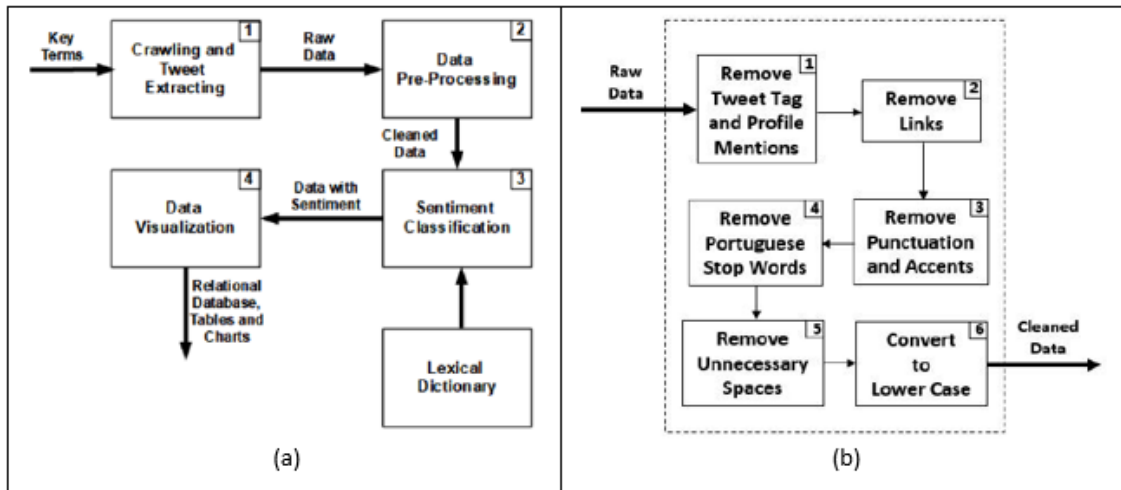


Figura 2.11 – Estrutura para análise de sentimentos (adaptado de [76]).

2.6.4 Comparação das Abordagens de Análise de Sentimentos

Esta seção compara as diferenças e eficiências entre a abordagem léxica e aprendizado de máquina. A abordagem com técnicas de aprendizado de máquina utilizou os classificadores *Naive Bayes*, SVM e Regressão Logística. Para a abordagem léxica foram utilizados os léxicos de sentimentos: *SentiLex-PT*, *TextBlob*, *SentiWordNet* e *Valence Aware Dictionary and sEntiment Reasoner* (VADER).

Tumitan e Becker [77] utilizam duas abordagens (léxica e aprendizado de máquina) para classificar sentimentos de *tweets* para análise de tendência política. Eles investigam a possibilidade de prever a variação na intenção de voto com base em séries temporais de sentimentos extraídos de um conjunto de dados com notícias de três eleições brasileiras. Para polarizar as mensagens em positiva, negativa ou neutra (1, -1 e 0), os autores utilizaram o léxico de sentimentos *SentiLex-PT*. Na abordagem de aprendizado de máquina, o algoritmo SVM foi treinado para classificar as mensagens em positivas ou negativas. A Figura 2.12 mostra a proposta utilizada. Conforme os autores, os resultados das duas abordagens não foram satisfatórios, tendo em vista a indisponibilidade de bons léxicos e de um processo de anotação para dados de treinamento em algoritmos de aprendizado de máquina.

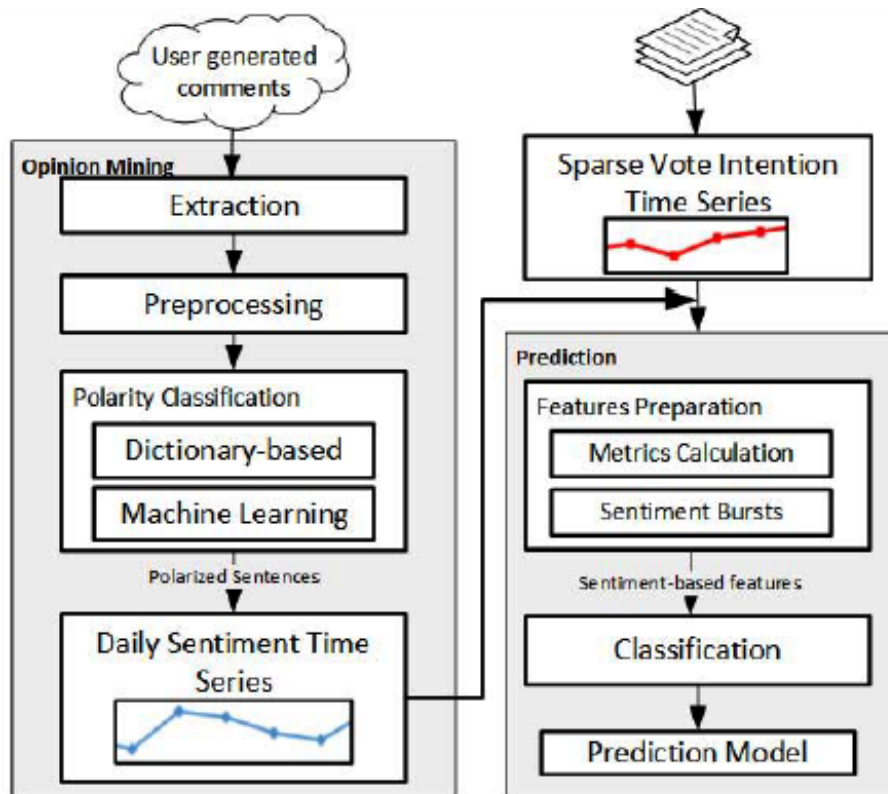


Figura 2.12 – Visão geral da proposta utilizada [77].

Sohangir et al. [78] utiliza a abordagem léxica como método para analisar o sentimento das mensagens postadas em 2015 e 2016 da rede social financeira *StockTwits*. Os autores procuram determinar se a classificação com abordagem léxica melhora a precisão da análise de sentimentos dos dados *StockTwits* em comparação com a abordagem de aprendizado de máquina. Assim, eles comparam os classificadores léxicos *TextBlob*, *SentiWordNet* e *VADER* com os classificadores que utilizam técnicas de aprendizado de máquina *Naive Bayes*, *SVM* e *Regressão Logística*. A Figura 2.13 representa uma área comparativa através da curva Característica de Operação do Receptor (ROC) entre a abordagem léxica e a abordagem de aprendizado de máquina. Os desempenhos dos três classificadores com abordagem de aprendizado de máquina foram bem próximos. Por outro lado, os léxicos de sentimentos superam os métodos de aprendizado de máquina. Os autores citam também que as técnicas de aprendizado de máquina têm como desvantagem o processo de treinamento, onde consomem muito tempo e é computacionalmente caro em termos de CPU e requisitos de memória. A abordagem léxica não precisa de dados de treinamento e é favorável principalmente em tarefas que envolvem grande quantidade de dados.

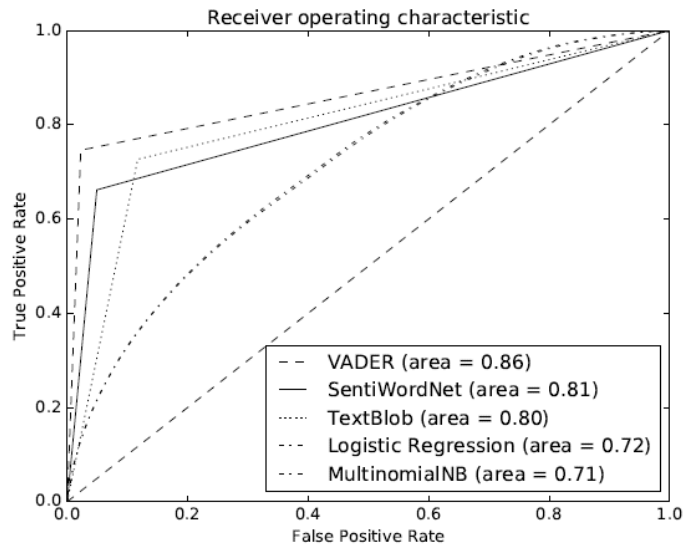


Figura 2.13 – Área comparativa através da curva ROC [78].

Shekhawat [79] compara o desempenho do algoritmo *Naive Bayes* com a abordagem de análise de sentimento da biblioteca *TextBlob*. O autor utiliza o conjunto de dados *Sentiment140* do *Twitter* [80] para treinar o algoritmo *Naive Bayes* e comparar os resultados com a biblioteca *TextBlob* na classificação de opinião pública de diferentes países sobre a reunião do BREXIT (saída do Reino Unido da União Europeia). A Figura 2.14 mostra o fluxo de trabalho utilizado com o classificador *Naive Bayes* e a biblioteca *TextBlob*. Os resultados dos testes mostraram que o algoritmo *Naive Bayes* (74%) alcançou uma maior precisão em relação ao *TextBlob* (61%).

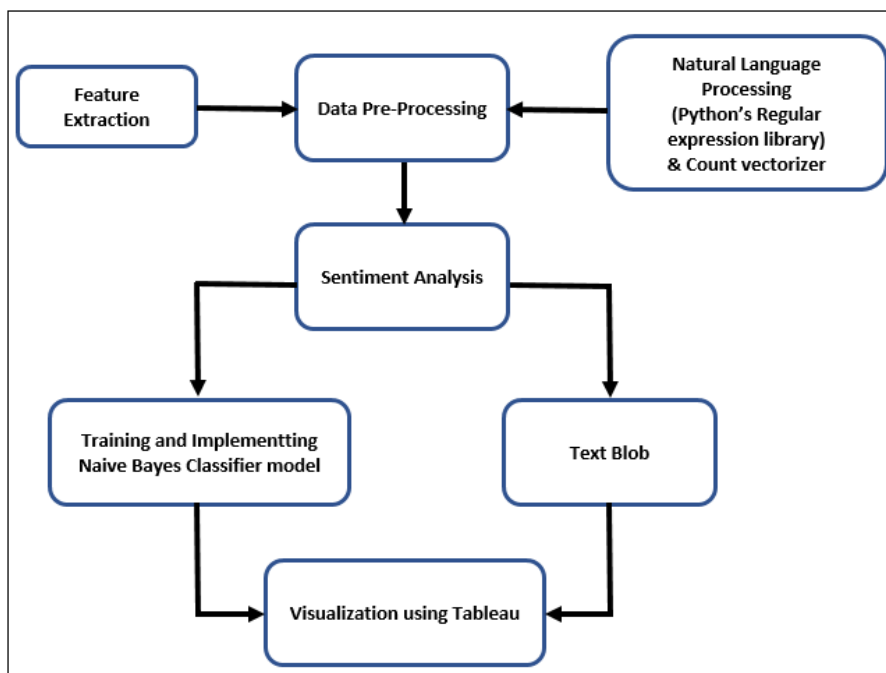


Figura 2.14 – Fluxo de trabalho com o classificador *Naive Bayes* e a biblioteca *TextBlob* (adaptado de [79]).

Laksono et al. [81] tenta classificar a satisfação dos clientes do restaurante *Surabaya* na plataforma de mídia *on-line TripAdvisor* para identificar sentimentos positivos e negativos. Conforme os autores, as opiniões dos clientes em plataformas de mídia *on-line* podem aumentar a popularidade do produto ou serviço vendido por agências. Para realizar o pré-processamento dos dados, eles utilizaram *Python* e *WebHarvy*. A classificação dos dados foi feita através do classificador *Naive Bayes* e do analisador de sentimento *TextBlob* (Figura 2.15). O objetivo também era identificar o melhor método para analisar dados de clientes de restaurantes, uma vez que, esses dois métodos têm diferenças fundamentais em termos de cálculos. Os resultados mostraram que os dois métodos foram suficientes para analisar as respostas dos clientes, sendo o método *Naive Bayes* mais preciso em 2,9% do que o analisador de sentimento *TextBlob*.

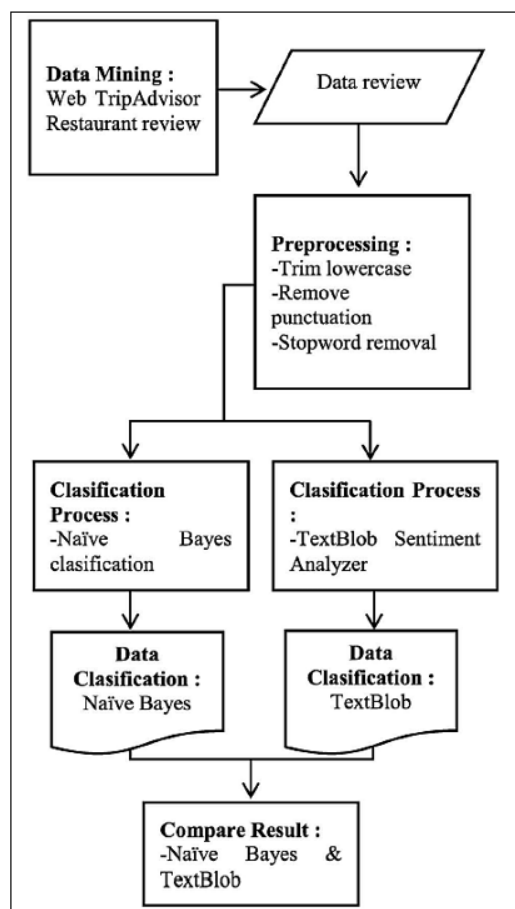


Figura 2.15 – Processo de análise dos dados (adaptado de [81]).

2.7 ATAQUES ADVERSARIAIS EM ALGORITMOS DE CLASSIFICAÇÃO

Aplicações com técnicas de Inteligência Artificial têm sido amplamente utilizadas em Processamento de Linguagem Natural (NLP), entre outros campos. No entanto, pesquisas sobre esse tema, tem se tornado um ponto crítico com novos métodos de defesa e ataques

adversários. No domínio da linguagem natural, pequenas perturbações são claramente perceptíveis, e a substituição de uma única palavra ou pequenas perturbações na forma de erros ortográficos podem alterar drasticamente o resultado da saída de um classificador de sentimento [82], alterando a percepção dos analistas e influenciando usuários da rede.

Conforme Hossein et al. [83], essas técnicas são vulneráveis a presença de adversários inteligentes e adaptativos. O autor propôs um ataque com base em exemplos adversários em um sistema desenvolvido pela *Google* e *Jigsaw*, chamado de *Perspective*, que utiliza aprendizado de máquina para detectar automaticamente linguagem tóxica (insultos *on-line*, assédios e discursos abusivos) [84]. Ele mostrou que uma pequena modificação (pontuações entre letras ou palavras incorretas) em uma frase altamente tóxica pode reduzir consistentemente os índices de toxicidade ao nível das frases não tóxicas. A Tabela 2.1 mostra algumas frases utilizadas pelo autor para realizar os ataques no sistema *Perspective* (frases originais e frases modificadas com os *scores* de toxicidade). É possível verificar nas duas últimas frases que o sistema *Perspective* atribui de forma errada, altos índices de toxicidade nas frases aparentemente benignas, sendo neste caso, considerada um falso positivo.

Tabela 2.1 – Demonstração dos ataques no sistema *Perspective* (adaptado de [83]).

Frase Original (Pontuação de Toxicidade)	Frase Modificada (Pontuação de Toxicidade)
<i>Climate change is happening and it's not changing in our favor. If you think differently you're an idiot.</i> (84%)	<i>Climate change is happening and it's not changing in our favor. If you think differently you're an idiot.</i> (20%)
<i>They're stupid, it's getting warmer, we should enjoy it while it lasts</i> (86%)	<i>They're stupid, it's getting warmer, we should enjoy it while it lasts</i> (2%)
<i>They are liberal idiots who are uneducated.</i> (90%)	<i>They are not liberal idiots who are uneducated.</i> (83%)
<i>They are stupid and ignorant with no class</i> (91%)	<i>They are not stupid and ignorant with no class</i> (84%)

Para Wong [85] existe relativamente poucos trabalhos que exploram a geração de exemplos adversários para classificadores de texto. O autor cita também que muitos algoritmos que geram esses exemplos precisam de acesso total aos parâmetros do modelo de destino. Para lidar com esse problema Wong [85] apresenta o DANCin SEQ2SEQ, um algoritmo inspirado em *Generative Adversarial Network* (GAN) que visa gerar exemplos de textos adversários para atacar classificadores de textos *black-box*. Ele reformulou a geração de textos adversários com uma tarefa de aprendizado por reforço e introduziu um algoritmo capaz de substituir diversas palavras semanticamente semelhantes para aumentar a probabilidade de classificação incorreta do alvo, preservando a igualdade semântica humana. O DANCin SEQ2SEQ foi testado em um classificador de texto real treinado com o conjunto de dados *Enron-Spam*. Os resultados mostraram que o gerador do algoritmo aprende a identificar e

substituir *tokens* adversários que aumentam a classificação incorreta, preservando a semelhança semântica humana.

Li et al. [86], cita que vulnerabilidades de segurança em Compreensão de Texto com Base em *Deep Learning* (DLTU) ainda são amplamente desconhecidas. O autor mostra que essa tecnologia é vulnerável a ataques de textos adversários. Seu trabalho [86] apresenta o *TEXTBUGGER*, uma estrutura de ataque que gera textos adversários. Os autores utilizaram os conjuntos de dados IMDB e MR para estudar os ataques adversariais (perturbações de caracteres e palavras). A eficiência do *TEXTBUGGER* foi testada em um conjunto de sistemas e serviços DLTU (*Google Cloud NLP*, *DLTU Waston*, *Microsoft Azure*, *Amazon AWS*, *Facebook fastText*, *ParallelDots*, *TheySay*, *Aylien Sentiment*, *TextProcessing*, *Mashape Sentiment*) utilizado para análise de sentimentos e detecção de conteúdo tóxico. Conforme os autores, sistemas DLTU utilizam um dicionário para representar um conjunto finito de palavras. Assim, palavras importantes podem ser convertidas em palavras desconhecidas. Os principais resultados dos ataques *black-box* mostraram que *TEXTBUGGER* perturba apenas algumas palavras para atingir 100% de sucesso no conjunto de dados do IMDB direcionado para as plataformas *Microsoft Azure* e *Amazon AWS*. Para o conjunto de dados MR direcionado para a plataforma *Microsoft Azure*, *TEXTBUGGER* perturbou apenas 7% das palavras para atingir 96,8%. Os autores citam também como método de defesa a possibilidade de verificação ortográfica para mitigar o ataque. A taxa de sucesso da aplicação do *TEXTBUGGER* na API *Perspective* após a verificação ortográfica foi de 35,6%.

Samanta et al. [87], mostra um novo método para elaborar amostras de textos adversários, alterando as amostras originais através da exclusão ou substituição de palavras importantes ou adicionando novas palavras na amostra do texto. Resultados experimentais no *dataset* de resenhas de filmes do IMDB para análise de sentimentos e no *dataset* do *Twitter* para detecção de gênero mostraram a eficiência do método proposto.

Alzantot et al. [82] propõe a geração de exemplos adversários através da utilização de um algoritmo genético de base populacional para substituir palavras por sinônimos, a fim de gerar semanticamente e sintaticamente exemplos adversários semelhantes que enganam a análise de sentimentos bem treinada no primeiro experimento e os modelos de vinculação textual, no segundo experimento. Uma comparação entre a taxa de sucesso do ataque e a porcentagem média de modificações exigidas pela genética do ataque mostra a eficiência do método proposto em ambos os experimentos. Uma validação humana mostrou que os exemplos gerados eram considerados contraditórios e perceptivelmente bastante semelhantes. Na Tabela 2.2 é possível observar um exemplo de ataque adversarial com os resultados. As palavras modificadas estão destacadas na cor verde (texto original) e vermelho (texto adversarial).

Tabela 2.2 – Exemplo com os resultados de um ataque adversarial (adaptado de [82]).

Texto Original = Negativo (Confiança = 78%)
<i>This movie had terrible acting, terrible plot, and terrible choice of actors. (Leslie Nielsen ...come on!!!) the one part I considered slightly funny was the battling FBI/CIA agents, but because the audience was mainly kids they didn't understand that theme.</i>
Texto Adversário = Positivo (Confiança = 59.8%)
<i>This movie had horrific acting, horrific plot, and horrifying choice of actors. (Leslie Nielsen ...come on!!!) the one part I regarded slightly funny was the battling FBI/CIA agents, but because the audience was mainly youngsters they didn't understand that theme.</i>

Gao et al. [88] apresenta o *DeepWordBug*, um algoritmo que gera pequenas perturbações de texto para forçar um classificador de aprendizado profundo *black-box* classificar incorretamente uma entrada de texto. Estratégias de pontuações foram desenvolvidas pelos autores para encontrar as palavras mais importantes a serem modificadas com pequenas perturbações a nível de caractere, forçando o classificador fazer uma previsão errada. A Figura 2.16 mostra um exemplo de texto adversário gerado pelo *DeepWordBug*. É possível observar duas partes: uma com amostra de texto original e outra com o texto adversário gerado a partir da amostra de texto original. Percebe-se também que apenas alguns caracteres são modificados para enganar o classificador. O *DeepWordBug* foi avaliado em dois conjuntos de dados: *Enron-Spam* e resenhas de filmes do IMDB. Os resultados mostraram que o algoritmo pode reduzir a precisão da classificação de 99% para 40% no *Enron-Spam* e de 87% para 26% no IMDB.

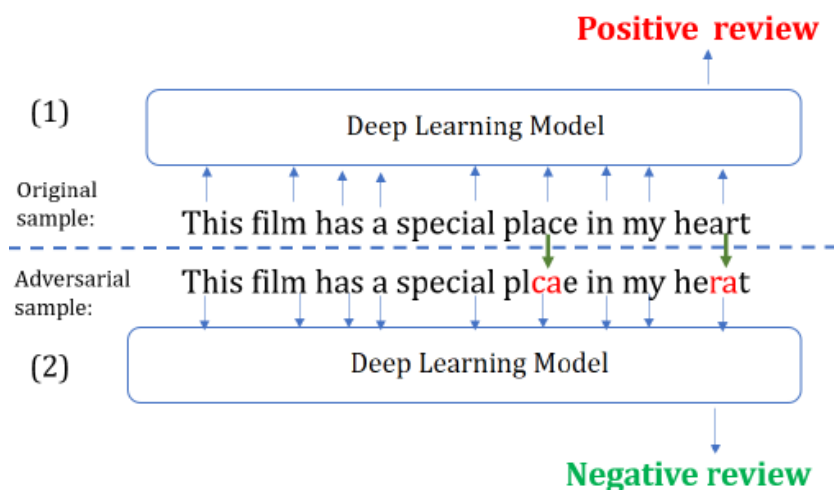


Figura 2.16 – Exemplo de texto adversário gerado pelo *DeepWordBug* (adaptado de [88]).

Tsai et al. [89] apresenta um método chamado "*Global Search*", que consiste em um algoritmo de ataque *white-box*. Este método é comparado com um simples erro de ortografia e com outra abordagem de ataque *white-box* chamado "*Greedy Search*". Um classificador de

sentimentos da Rede Neural Convolutacional (CNN) é treinado no conjunto de dados de revisão de filmes do IMDB. Em seguida, os métodos de ataque são avaliados. Além da precisão da classificação, os autores analisaram também através de 15 voluntários, a semelhança entre os textos originais e os textos adversários. Como resultado das experiências, o método "*Global Search*" proposto gera exemplos adversários com maior precisão e semelhança (menos deformações ou menos alterações) em relação ao texto original (Tabela 2.3). Assim, embora o classificador de sentimentos classifique de forma errada os exemplos adversários do *Global Search*, um humano classificará de forma correta o sentimento do mesmo texto adversário. Ainda conforme os autores, os exemplos adversários do método *Greedy Search* são difíceis de ler e não podem informar o sentimento do texto.

Tabela 2.3 – Exemplos de textos adversários - métodos *Global Search* e *Greedy Search* (adaptado de [89]).

Texto Original	<i>as long as you go into this movie knowing that it 's terrible : bad acting , bad " effects , " bad story , bad ... everything , then you 'll love it . this is one of my favorite " goof on " movies ; watch it as a comedy and have a dozen good laughs !</i>
<i>Global Search</i>	<i>as long as you go into this movie knowing that it 's terrible : worse acting , bad " effects , " bad story , bad ... everything , then you 'll love it . this is one of my favorite " goof on " movies ; watch it as a comedy and have a dozen good laughs yes</i>
<i>Greedy Search</i>	<i>as long as you leave into this blockbuster telling whether it 's horrendous : bad acting , bad " effects , " bad story , bad ... everything , then you 'll love it . this is one of my favorite " goof on " movies ; watch it as ...</i>

Vijayaraghavan et al. [90], mostra uma abordagem baseada em aprendizado por reforço para gerar exemplos adversariais em ambientes *black-box*. Conforme os autores, o método proposto foi capaz de enganar com taxas de sucesso consideravelmente altas, preservando a semântica do texto original, modelos bem treinados para classificação de sentimentos do IMDB e categorização de notícias do corpus da AG (conjunto de dados com mais de um milhão de artigos que foram coletados em mais de 2.000 fontes de notícias)..

Jin et al. [91] apresenta o *TextFooler*, um modelo para ataques de linguagem natural em ambientes *black-box*. Conforme os autores, *TextFooler* identifica no texto palavras importantes e substitui por palavras semanticamente semelhantes e gramaticalmente corretas até que a previsão do alvo seja alterada. Resultados mostraram que o algoritmo foi capaz de reduzir a precisão de quase todos os modelos de destino, inclusive o *Bidirectional Encoder Representations from Transformers* (BERT) em todas as tarefas. Além disso, os textos adversários produzidos pelo algoritmo *TextFooler* foram classificados corretamente por avaliadores humanos. A Figura 2.17 apresenta um ataque com texto gerado pelo modelo

TextFooler. Percebe-se que o texto adversário é semanticamente semelhante ao texto original e gramaticalmente aceitável por humanos.

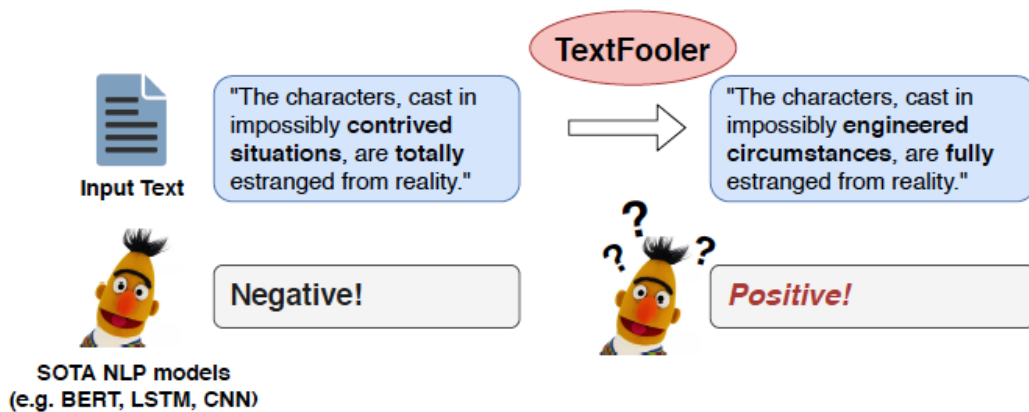


Figura 2.17 – Exemplo de um ataque com texto gerado pelo modelo *TextFooler* [91].

2.8 CONTRIBUIÇÕES PRINCIPAIS DESTE TRABALHO

Este trabalho apresenta como contribuição principal a aplicação de estratégias de ataques adversariais em um classificador léxico de linguagem natural. Do ponto de vista da arquitetura de coleta, para fundamentar o desenvolvimento do ambiente proposto à frente, foram discutidos os aspectos relativos a representação visual, análise de dados de mídia social, *bot* social, sistemas distribuídos, sistemas de anonimato, visualização de dados e análise de sentimentos na classificação de textos. Do lado da aplicação dos ataques adversariais, o *corpus* e o código do classificador de sentimento utilizado na camada de classificação da arquitetura de coleta foram estudados e utilizados para identificar vulnerabilidades com a finalidade de desenvolver estratégias de ataques adversariais e procedimentos de defesa que poderão ser utilizados para mitigar esses ataques em aplicações que empregam essas técnicas para classificar textos.

2.8.1 Estratégias de Ataques Adversariais

Na Seção 2.6.1 foi discutida a literatura existente sobre classificadores de sentimentos que utilizam a abordagem léxica para projetar modelos de previsão com base em textos extraídos de mídias sociais. Estes trabalhos foram argumentados visando fundamentar a elaboração da proposta das estratégias dos ataques adversariais. Todos os autores [44, 53, 54, 55, 56, 59, 60, 61] utilizaram a biblioteca *TextBlob* para classificar os textos. Entretanto, os trabalhos [44, 53, 54, 55, 56, 59, 60] não citaram o *corpus* (*en-sentiment.xml*) e o algoritmo (*Pattern Analyzer*) empregado por essa biblioteca no processo de análise de sentimentos. O *corpus en-sentiment.xml* e o algoritmo *Pattern Analyzer* possuem vulnerabi-

lidades que podem ser exploradas por usuários mal-intencionados para realizar ataques de substituição (Seção 5.2). Além disso, nenhum método de pré-processamento foi feito nos trabalhos [53, 59] para melhorar a precisão e acelerar o processo de classificação dos *tweets*. Não ficou evidente também, para os trabalhos que aplicaram a fase de pré-processamento [54, 55, 56, 44, 60] algum método de tradução (qualquer idioma para o idioma inglês que está no *corpus*) e correção (palavras escritas com erros de ortografia) para os textos analisados. Métodos de correção devem ser aplicados para mitigar os ataques de inserção de caracteres no classificador *Pattern Analyzer* da biblioteca *TextBlob* (Seção 5.1).

A Seção 2.6.2 revisou os conceitos e apresentou alguns trabalhos que utilizaram técnicas de aprendizado de máquina para classificar textos. Alguns pontos devem ser considerados, por exemplo, Manguri et al. [70] não deixa claro qual o algoritmo da biblioteca *TextBlob* foi utilizado para fazer a classificação dos textos. Outros trabalhos [68, 69, 70] não apresentaram nenhuma informação sobre o *corpus* utilizado para treinar e validar o algoritmo de classificação. Além disso, alguns métodos não foram aplicados na fase de pré-processamento para melhorar a classificação dos dados e mitigar os ataques de inserção [67, 69, 70].

O algoritmo da biblioteca *TextBlob* (abordagem léxica) aplicado nos trabalhos da Seção 2.6.3, apresenta os mesmos problemas de vulnerabilidades (seções 5.1 e 5.2) que foram apontados nos trabalhos anteriores [44, 53, 54, 55, 56, 59, 60]. Assim, a eficácia da classificação da abordagem híbrida pode ser prejudicada se os métodos de correção [73, 74, 75, 76] e tradução [75, 76] não forem implementados na fase de pré-processamento. Ambientes de coleta sem esses métodos podem ser explorados por usuários mal-intencionados para enganar o classificador de sentimento.

A Tabela 2.4 destaca os trabalhos relacionados que são vulneráveis para os ataques de inserção (ATK1 - caracteres em frases com idioma inglês e ATK2 - caracteres em frases com outros idiomas) e substituição (ATK3 - frases negativas ou positivas e ATK4 - frases com palavras de negação). É possível observar que o trabalho [61] foi o único que mitigou através dos métodos de tradução e correção, aplicados na fase de pré-processamento, o ataque de inserção de caracteres em frases com idioma inglês. Todos os outros trabalhos foram vulneráveis para todos os ataques apresentados nesta seção.

Tabela 2.4 – Trabalhos vulneráveis para os ataques de inserção e substituição.

	[44]	[53]	[54]	[55]	[56]	[59]	[60]	[61]	[73]	[74]	[75]	[76]
ATK1	x	x	x	x	x	x	x	-	x	x	x	x
ATK2	x	x	x	x	x	x	x	x	x	x	x	x
ATK3	x	x	x	x	x	x	x	x	x	x	x	x
ATK4	x	x	x	x	x	x	x	x	x	x	x	x

Os trabalhos mencionados na Seção 2.7 [82, 83, 85, 86, 87, 88, 89, 90, 91] tem caracte-

rísticas em comum, especialmente em termos de classificadores de sentimentos que utilizam a abordagem com aprendizado de máquina. Nesse contexto, esses ataques possuem modelos em comuns, como a implementação de algoritmos para gerar textos adversários, embora cada um use estratégias de ataques diferentes para serem aplicados em classificadores. Entretanto, a maioria das estratégias de ataques aqui estudadas provam de certa forma a sua efetividade, mas não consideram os ataques adversariais em classificadores de sentimento que utilizam a abordagem léxica. Outro ponto a ser considerado para o sucesso dos ataques neste trabalho foram os procedimentos de engenharia reversa aplicados no *corpus* e no código do algoritmo para identificar vulnerabilidades.

Assim, o trabalho apresentado nesta tese difere das abordagens descritas, uma vez que as estratégias de ataques adversariais e os procedimentos de defesa deste trabalho foram projetados para serem realizados em um algoritmo de classificação que atualmente está sendo muito utilizado na abordagem léxica [44, 53, 54, 55, 56, 59, 60, 61] e na abordagem híbrida [51, 73, 74, 75, 76]. Outros trabalhos também foram publicados com o mesmo algoritmo para comparar a abordagem léxica e aprendizado de máquina [77, 78, 79, 81]. Os atributos (AT1 - AT10) deste trabalho em relação as técnicas de engenharia reversa, estratégias de ataques adversariais e contramedidas para mitigar esses ataques são os seguintes:

- AT1: Engenharia reversa no *corpus* para identificar vulnerabilidades;
- AT2: Engenharia reversa no código do algoritmo para identificar vulnerabilidades;
- AT3: Ataques de inserção em frases com idioma inglês;
- AT4: Ataques de inserção em frases com outros idiomas;
- AT5: Ataques de substituição;
- AT6: Classificador de sentimento léxico;
- AT7: Contramedidas para mitigar os ataques de inserção em frases com idioma inglês;
- AT8: Contramedidas para mitigar os ataques de inserção em frases com outros idiomas;
- AT9: Contramedidas para mitigar os ataques de substituição no *corpus*;
- AT10: Contramedidas para mitigar os ataques de substituição no código do algoritmo.

A Tabela 2.5 destaca a diferença deste trabalho (T) em relação aos trabalhos relacionados sobre as estratégias de ataques adversariais. Os atributos foram abreviados para facilitar a visualização do que é comum a cada trabalho referenciado.

Tabela 2.5 – Diferença entre este trabalho (T) e os trabalhos relacionados.

	AT1	AT2	AT3	AT4	AT5	AT6	AT7	AT8	AT9	AT10
T	x	x	x	x	x	x	x	x	x	x
[82]	-	-	-	-	x	-	-	-	-	-
[83]	-	-	x	-	-	-	-	-	-	-
[85]	-	-	-	-	x	-	-	-	-	-
[86]	-	-	x	-	-	-	x	-	-	-
[87]	-	-	x	-	-	-	-	-	-	-
[88]	-	-	x	-	-	-	-	-	-	-
[89]	-	-	-	-	x	-	-	-	-	-
[90]	-	-	x	-	-	-	-	-	-	-
[91]	-	-	-	-	x	-	-	-	-	-

2.8.2 Arquitetura de Coleta

A arquitetura de coleta denominada de *OctopusViz*, foi construída para estudar, desenvolver e realizar as estratégias de ataques adversariais e respectivos procedimentos de defesa. O classificador léxico de linguagem natural *Pattern Analyzer* foi implementado na subcamada de classificação da arquitetura. Esse classificador é utilizado para calcular o sentimento dos dados postados e coletados por usuários em diversas aplicações de mídia social.

Tendo em vista a grande vantagem de utilizar o *Twitter* como rede social para mineração, este trabalho busca detectar as opiniões dos usuários no *Twitter*. São extraídos e analisados os *tweets* que indiquem os sentimentos dos autores das postagens, estatísticas lógicas de usuários, *hashtags*, *retweets*, menções, quantidade de *likes* e mapeamento de usuários através de grafos de acordo com o interesse do analista. O ambiente oferece também a construção de uma base de dados real e atualizada para pesquisa e análise.

Neste trabalho, um processo de mineração de texto semelhante a Gomes et al. [51] é usado, no entanto, a aplicação será na rede social *Twitter*. A escolha de usar essa rede social é pelo alcance global da mesma, que possui milhões de usuários cadastrados. Os textos a serem minerados compõem um *tweet*, que é uma sequência de caracteres publicada pelos usuários, podendo conter outros tipos de dados anexados.

Este trabalho assemelha-se aos trabalhos [92, 64, 73, 76, 77] para determinar o sentimento expresso pelos usuários, correlacionando os resultados aos fatos ocorridos em um determinado período voltado a um contexto de interesse. Entretanto, difere-se dos trabalhos citados anteriormente por sua operação ser em tempo real, apresentando resultados dentro de limites de tempo previamente definidos (dados em prazos compatíveis com a ocorrência de eventos externos), além de considerar outras palavras do *tweet* que possam expressar um sentimento, mesmo que estas não estejam marcadas como uma *hashtag*.

Diferente dos trabalhos de [14, 41, 75] a classificação será feita por meio de um algoritmo de classificação que utiliza abordagem léxica, que classificará em tempo real as opiniões dos usuários do *Twitter* utilizando a mesma categorização: positivas, negativas e neutras. Isso será melhor explicado na Seção 3.2.3.

Quanto ao quesito de visualização, este trabalho assemelha-se aos trabalhos de Sijtsma et al. [42], de Oliveira Júnior et al. [33] e Pimenta Rodrigues et al. [43] pela riqueza das opções de visualização. A Seção 3.3.1.10 apresenta também um método rápido e prático de identificação de *bots* sociais através da análise de *Outliers*.

Diferente de todos os trabalhos apresentado acima, este prioriza o anonimato dos pesquisadores utilizando uma VPN para a coleta dos dados. A Tabela 2.6 apresenta a diferença entre os trabalhos relacionados e o ambiente *OctopusViz*.

Tabela 2.6 – Diferença entre o *OctopusViz* e os trabalhos apresentados.

	Anonimização	Análise de Sentimento	Operação em Tempo Real	Armazenamento Distribuído	Visualização
<i>OctopusViz</i>	x	x	x	x	x
[14]	–	x	x	–	–
[41]	–	x	x	x	x
[42]	–	x	–	–	x
[33]	–	–	x	x	x
[43]	–	–	–	x	x
[92]	–	x	–	–	–
[64]	–	x	–	–	–
[73]	–	x	–	–	–
[75]	–	x	x	–	–
[76]	–	x	–	–	–
[77]	–	x	–	–	–

2.9 RESUMO DO CAPÍTULO 2

Neste capítulo, foram apresentados os conceitos sobre análise de dados de mídia social, *bot* social, sistemas de anonimato, sistemas distribuídos e visualização de dados, necessários para a elaboração da arquitetura de coleta. Na sequência, foram também apresentados a fundamentação teórica sobre análise de sentimentos na classificação de textos e estratégias de ataques adversariais. Estas foram discutidas nas principais contribuições do trabalho visando fundamentar a elaboração dos ataques adversariais.

3

DESCRIÇÃO DO PROBLEMA

Devido ao crescimento exponencial das mídias sociais em todo o mundo, cada vez mais órgãos de Governo e empresas estão dependentes de informações para suas estratégias de negócio. Segundo Gomes et al. [51], com a popularização da Internet, as pessoas geram um imenso volume de dados a cada segundo. O desafio/problema é saber como manipular essa grande quantidade de informação gerada e investigar como as organizações podem se beneficiar desses dados, considerando que grande parte desses conhecimentos estão contidos em textos, além de poder realizar análises de dados em tempo real.

Assim, torna-se fundamental desenvolver técnicas para acompanhar e observar a evolução de um determinado tema, gerando dados que colaborem no processo de tomada de decisão. Na análise de mídias sociais, considera-se que um analista deve levar em conta e usar todas as medidas legais a seu alcance, no sentido de identificar possíveis influenciadores e o sentimento de usuários em relação a um determinado assunto.

Entretanto, os classificadores de sentimentos léxicos utilizados nesses ambientes de coleta [44, 51, 53, 54, 55, 56, 59, 60, 73, 74, 75, 76, 78, 79, 81] devem ser levados em consideração, tendo em vista os aspectos relacionados às estratégias de ataques adversariais que podem ser fabricados por usuários mal-intencionados para atacar esses ambientes (Seção 2.8.1).

A título de ilustração, cabe citar como exemplo, a aplicação LUX *Election 2020* [93], um *software* de análise que mostra em tempo real o sentimento dos eleitores sobre candidatos políticos. Conforme notícia veiculada pela revista *Forbes* [94], essa aplicação mostrou que Joe Biden estava acumulando sentimentos positivos, mesmo entre os republicanos, enquanto seu oponente Donald Trump, apresentava um sentimento mais negativo entre os eleitores.

Considerando esses detalhes observados, suponha um atacante com informações mínimas sobre o *corpus* e o algoritmo de classificação utilizados pelo *software* LUX *Election 2020* para realizar análise de sentimentos. Esse atacante poderia gerar conteúdo nas mídias sociais que o *software* usa como fonte de dados e, como consequência, inundar o sistema com ataques adversariais para alterar a percepção do classificador de sentimento. Se o *software* de análise LUX *Election 2020* não possuir contramedidas, é plausível afirmar que essa plataforma poderia gerar informações erradas devido aos ataques capturados como fonte de dados e induzir os tomadores de decisão com informações equivocadas. Por exemplo, Donald Trump acumulando um sentimento mais positivo entre os eleitores do que o democrata Joe Biden.

Cabe observar que essas informações erradas, publicadas pelo *software* LUX *Election*

2020 e acessada por milhões de usuários, poderiam ser utilizadas também como desinformação (*fake news*) para dar uma falsa imagem da realidade e manipular a opinião pública dos eleitores durante a eleição dos Estados Unidos, podendo gerar inclusive risco à democracia.

Conforme revisado na Seção 2.7, existem vários trabalhos que tratam apenas de ataques adversariais em classificadores de sentimento que utilizam algoritmos de aprendizado de máquina. Como ficou evidente, os ataques adversariais não são realizados em algoritmos de classificação que utilizam a abordagem léxica para classificar textos.

Como já foi apresentado na Seção 1.1, o objetivo principal deste trabalho é identificar vulnerabilidades para desenvolver estratégias de ataques adversariais em um classificador de sentimento léxico. Além disso, apresentar também as contramedidas que podem ser utilizadas para mitigar esses ataques. Para esses processos em si, os seguintes objetivos específicos são requisitos: construir e validar uma arquitetura de coleta que utiliza técnicas de análise de sentimento com abordagem léxica; utilizar esse ambiente para coletar e monitorar dados em tempo real; e por fim, estudar e verificar o comportamento do classificador utilizado na camada de classificação da arquitetura de coleta.

Inicialmente serão apresentados os aspectos relacionados ao desenvolvimento e eficiência do ambiente proposto, cuja arquitetura (Figura 3.1) tem como objetivo monitorar e realizar busca anônima de *tweets* em tempo real para gerar informações que possam avaliar a análise de sentimentos em relação a um determinado assunto. Os aspectos sobre as técnicas de engenharia reversa, estratégias de ataques adversariais e a representação visual destes ataques no ambiente *OctopusViz* serão abordados, respectivamente, nos Capítulos 4, 5 e 6.

3.1 PROPOSTA DA ARQUITETURA DE COLETA

Com a intenção de estudar, identificar vulnerabilidades e testar a efetividade dos ataques em um classificador de sentimento léxico, propomos uma solução denominada de *OctopusViz*, um *framework* que compreende um conjunto de aplicativos que monitora e coleta uma grande quantidade de *tweets* em tempo real de forma anônima e *online* para processar, pesquisar, visualizar e classificar automaticamente os sentimentos das mensagens em três categorias distintas: positiva, negativa e neutra. Para isso, o *OctopusViz* captura os *tweets* de acordo com o interesse do analista, classifica os seus sentimentos usando um algoritmo de classificação que utiliza abordagem léxica e, por fim, exibe os resultados em gráfico comparando diferentes consultas.

Usando o *framework*, é possível acompanhar e analisar dados e sentimentos sobre diversos temas, tais como: greves, eleições, empresas, *marketing*, protestos, ataques cibernéticos, operações militares e pesquisas de mercado. Este tipo de capacidade permite antecipar cenários possíveis para assessoramento no processo de tomada de decisão. Também permite

identificar facilmente anomalias conhecidas como *outliers*, que podem ser *bots* sociais tentando influenciar determinado assunto.

Como requisito fundamental e por questões de segurança e privacidade, o ambiente deve garantir que a fonte da coleta seja anônima, de maneira a não gerar oportunidade de tendências de alteração de dados e nem possibilidade de influência na origem dos dados. Para tanto, uma das maneiras de se conseguir tal requisito é por meio da utilização de uma VPN, para que eventuais fontes coletoras não sejam facilmente identificadas, evitando risco de contaminação de dados por parte da origem.

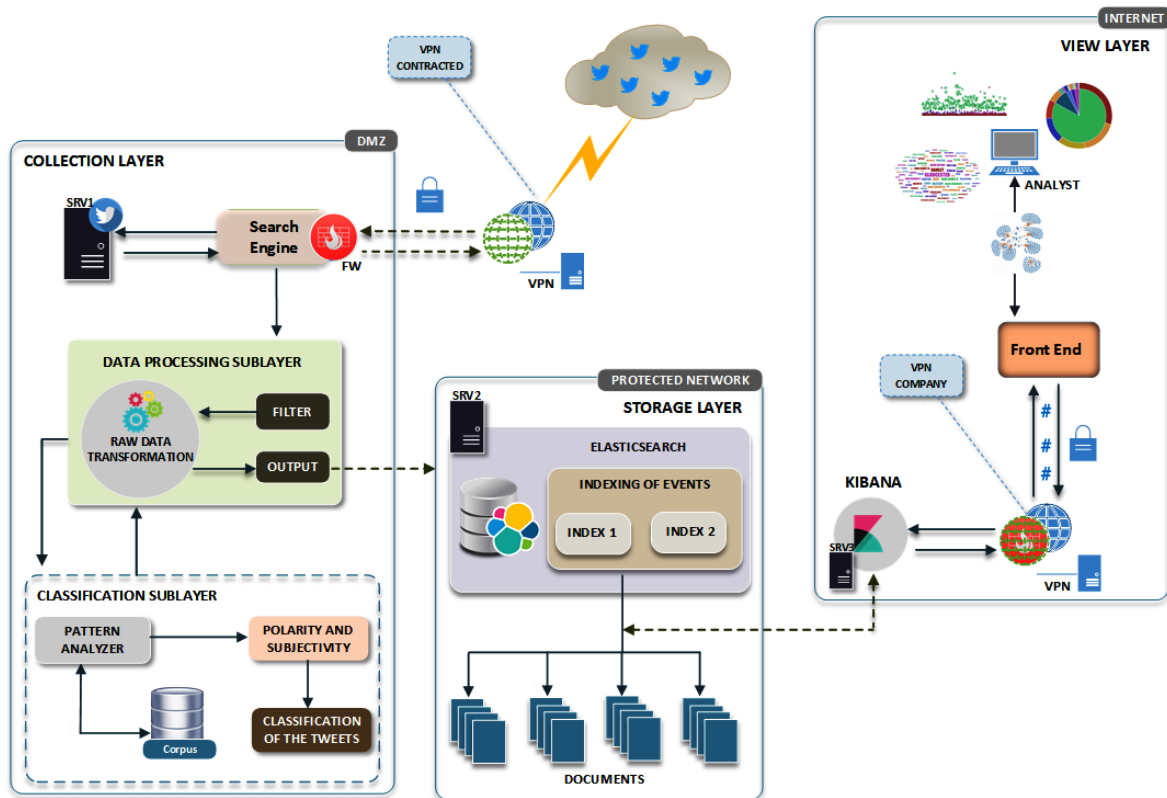


Figura 3.1 – Arquitetura *OctopusViz*.

A arquitetura física do *OctopusViz* (Figura 3.1) emprega um hospedeiro (*host*) ligado em uma Zona Desmilitarizada (DMZ) e a uma rede protegida do Laboratório de Pesquisa da Universidade de Brasília (UnB). Nesse *host* existem três máquinas virtuais.

O *Hypervisor XenServer* foi configurado para criar a infraestrutura lógica e de roteamento do ambiente [95]. Toda a estrutura do projeto foi feita em apenas um único *host* (*Dell PowerEdge R730*). A configuração e gerência dos sistemas convidados no *XenServer* é feita através do *XenCenter*. A Tabela 3.1 apresenta as características do *host* e *Hypervisor* utilizado para o desenvolvimento da arquitetura proposta. A Tabela 3.2 mostra a configuração dos sistemas convidados.

Tabela 3.1 – Características do *host*.

Servidor	Configuração
<i>Dell PowerEdge R730</i>	Processador <i>Intel Xeon E5-2690 v3 @ X5560</i> 2.6 GHz, 48 núcleos com tecnologia <i>Intel VT</i> , 128 GB de memória RAM, 6 discos de 1 TB configurados com RAID 5 e 6 placas de rede 10/100/1000.
<i>Hypervisor</i>	<i>XenServer 7.4</i> , DBV 2018.0223.

Tabela 3.2 – Sistemas convidados e suas configurações.

Sistemas Convidados	Configuração
Fw (Firewall)	Processador com 2 núcleos, 4 GB de memória RAM, um disco virtual de 50 GB e três interfaces virtuais de rede. Versão <i>pfSense-2.4.3-RELEASE</i> baseado no Sistema Operacional <i>FreeBSD</i> .
Srv1 (Coleta)	Processador com 12 núcleos, 16 GB de memória RAM, um disco virtual de 50 GB e uma interface virtual de rede. Sistema Operacional <i>Linux Debian Stretch 9.0</i> com a linguagem de programação <i>Python 3</i> e as bibliotecas <i>tweepy</i> , <i>json</i> , <i>time</i> , <i>elasticsearch</i> , <i>datatime</i> , <i>os</i> , <i>re</i> e <i>textblob</i> .
Srv2 (Armazenamento)	Processador com 12 núcleos, 32 GB de memória RAM, um disco virtual de 400 GB e uma interface virtual de rede. Sistema Operacional <i>Linux Debian Stretch 9.0</i> com o serviço <i>elasticsearch-6.2.4</i> .
Srv3 (Visualização)	Processador com 8 núcleos, 8 GB de memória RAM, um disco virtual de 50 GB e uma interface virtual de rede. Sistema Operacional <i>Linux Debian Stretch 9.0</i> com o serviço <i>kibana-6.2.4</i> .

A arquitetura do *OctopusViz* foi projetada para ter três camadas distintas (Figura 3.1): i) a camada de coleta faz a captura dos *streamings* de *tweets* em tempo real de acordo com palavras-chave inseridas na aplicação. Esta camada possui ainda i.i) a subcamada de processamento, que faz a transformação dos dados brutos em informações de interesse de acordo com os filtros e i.ii) a subcamada de classificação, que realiza análise de sentimentos e utiliza recursos computacionais para identificar a opinião pública dos usuários; ii) a camada de armazenamento distribuído faz a indexação e busca dos *tweets* recebidos da camada de captura, por fim iii) a camada de visualização é responsável pela apresentação dos dados para facilitar a interpretação dos analistas.

As camadas de coleta e visualização e as subcamadas de processamento de dados e classificação estão configuradas na rede DMZ do laboratório. A camada de armazenamento distribuído na rede protegida, com acesso restrito e controlado.

Para garantir o anonimato, a camada de coleta se autentica em um servidor VPN contratado durante a pesquisa deste projeto. Todos os *tweets* coletados pelo motor de busca são enviados para a camada de armazenamento distribuído que usa o *Elasticsearch* para oferecer suporte e indexar grandes volumes de dados, como os trabalhos [32, 33, 43]. A apresentação dos dados com métricas, estatísticas, grafos, indicando relacionamento diversos é feita através do *Kibana*, [32, 96, 33, 43].

3.2 DESCRIÇÃO DAS FASES DE IMPLEMENTAÇÃO

O desenvolvimento da arquitetura proposta ocorreu em cinco fases, sendo que a fase 1 trata da camada de coleta de dados; a fase 2, da subcamada de processamento de dados; a fase 3 da subcamada de classificação; a fase 4 da camada de armazenamento distribuído e a fase 5 dos aspectos da visualização dos *tweets* em tempo real. Os detalhes de cada fase são explicados a seguir.

3.2.1 Fase 1: Camada de Coleta dos Dados

Esta camada tem como finalidade coletar dados de interesse na plataforma *Twitter* em tempo real. A autenticação e coleta dos dados é feita com a biblioteca *Tweepy* [97] configurada em um *script Python* [98]. Esta API faz a autenticação do usuário cliente *Python* através de um aplicativo (chaves e *tokens*) criado no *Twitter*. Na arquitetura, o motor de busca é configurado para se autenticar na VPN contratada pelo projeto, com finalidade de garantir a privacidade e a confidencialidade [26].

3.2.2 Fase 2: Subcamada de Processamento de Dados

A subcamada de processamento faz a transformação dos dados brutos em dados de interesse. Como não existe um padrão de escrita para ser usado nas redes sociais, foi necessário executar alguns procedimentos (tradução, correção, *stopwords* e *tokenização*) para alcançar melhores resultados no refinamento das informações. Este processo é feito através das bibliotecas *TextBlob* e *NLTK* [99] em um *script python* (transformação e centralização dos dados). A API *TextBlob* trabalha com Processamento de Linguagem Natural (NLP), análise de sentimentos, classificação (algoritmos *Naive Bayes* e *Árvore de Decisão*), *tokenização*, tradução e correção ortográfica [10].

3.2.2.1 Tradução e Correção dos Dados Textuais

Antes de qualquer processamento, o idioma de cada *tweet* é detectado, traduzido automaticamente e corrigido para o idioma inglês. Essa tradução é feita de forma dinâmica pela API do *Google Translate* através dos métodos *get_languages()*, *detect_language()* e *translate()* [100]. A correção é feita pelo método *correct()* da biblioteca *TextBlob* [10]. Este procedimento permite que a proposta desenvolvida seja utilizada com mais de 100 idiomas e milhares de pares de idiomas. A Tabela 3.3 apresenta uma função com os métodos de tradução e correção utilizados no ambiente.

Tabela 3.3 – Função para tradução e correção dos *tweets*.

Entrada dos Dados em Português	O Brasil jogou muito bem contra a Costa Rica
Pré-processamento dos Dados (Tradução e Correção)	<pre>tweet = TextBlob("O Brasil jogou muito bem contra a Costa Rica") if tweet.detect_language() != 'en': translate_to_english = TextBlob(str(tweet.translate(to='en'))) correct_tweet = translate_to_english.correct() print(correct_tweet) else: tweet.correct() print(tweet.correct())</pre>
Saída dos Dados	<i>Brazil played very well against Costa Rich</i>

3.2.2.2 Stopwords e Caracteres Especiais

Esta técnica é aplicada durante a atividade de pré-processamento. Tem como finalidade remover as palavras que não possuem valor à análise. Geralmente correspondem a artigos, preposições, pontuações, conjunções e pronomes. O *corpus stopwords* e os métodos *stopwords.words()* e *string.punctuation* da biblioteca NLTK são utilizados para esta função [99]. Além disso, removemos também as URLs dos *tweets*, tendo em vista, que estas URLs direcionam para informações que não apresentam dados com requisitos para a análise de sentimentos em nosso trabalho. Outra vantagem desta técnica é a diminuição dos *tweets* e do tempo de processamento destes dados. A Tabela 3.4 mostra os caracteres especiais, pontuações e alguns *stopwords* que são removidos durante o pré-processamento. A Tabela 3.5 apresenta um exemplo da função utilizada para remover *stopwords* e caracteres especiais.

Tabela 3.4 – Palavras do *corpus stopwords* e caracteres especiais.

Métodos	Descrição dos Métodos	Saída dos Dados
<code>stopWords = set(stopwords.words('english'))</code> <code>print(stopWords)</code>	Palavras do <i>corpus</i>	<code>['i', 'me', 'my', 'we', 'our', 'ours', 'his', 'y', 'your', 'it']</code>
<code>string.punctuation</code>	Pontuações e caracteres especiais	<code>"!\"#\$%&'()*+,-./:;<=>?@[\\^_`{ } ’</code>

Tabela 3.5 – Função para limpeza dos *tweets*.

Exemplo de Entrada dos Dados	<code>Brazil is an excellent soccer team :) !!!</code>
Pré-processamento dos Dados (Stopwords e Caracteres Especiais)	<pre> tweet = TextBlob("Brazil is an excellent soccer team :) !!!") translation_correction(tweet) stopwords_english = stopwords.words('english') words = tweet.words words_clean = [] for word in words: if word not in stopwords_english: if word not in string.punctuation: words_clean.append(word) print (words_clean) </pre>
Saída dos Dados	<code>['Brazil', 'excellent', 'soccer', 'team']</code>

3.2.2.3 Tokenização

A identificação de *tokens* (palavras) é uma importante etapa do pré-processamento que divide textos em palavras, frases ou símbolos. Neste trabalho, utilizamos o método `textblob.tokenizers.WordTokenizer()` da biblioteca `TextBlob` para dividir os *tweets* em palavras individuais. As palavras geradas ajudam na análise e execução de outras tarefas da Subcamada de Classificação. A Tabela 3.6 mostra um exemplo da função utilizada para *tokenização*.

Tabela 3.6 – Função para tokenização dos *tweets*.

Exemplo de Entrada dos Dados	<code>Brazil played very well against Costa Rica</code>
Pré-processamento dos Dados (Tokenização)	<pre> tweet = TextBlob("Brazil played very well against Costa Rica") translation_correction(tweet) tweet_clean_stopwords(tweet) print (tweet.words) </pre>
Saída dos Dados	<code>['Brazil', 'played', 'very', 'well', 'against', 'Costa', 'Rica']</code>

3.2.3 Fase 3: Subcamada de Classificação

O principal objetivo dessa camada é realizar a análise de sentimentos dos *tweets* para identificar comportamentos que possam vir a mensurar a opinião pública dos usuários no *Twitter*. A biblioteca *TextBlob* [10] é configurada para realizar o processamento dos dados textuais.

3.2.3.1 Análise de Sentimentos

Neste trabalho, utilizamos a implementação do algoritmo *Pattern Analyzer* (baseado na biblioteca *Patterns* [101] e implementado na biblioteca *TextBlob* [10]) e um *corpus* léxico para classificar os *tweets*. Depois da transformação, os dados são enviados para o analisador de sentimentos. Nesta fase, o algoritmo *Pattern Analyzer* consulta o *corpus* léxico e faz a classificação dos *tweets* através da polaridade, subjetividade e intensidade. A pontuação da polaridade é atribuída dentro do intervalo $[(-1,0), (1,0)]$, onde: $[(0,0.1), (1,0)] =$ positivo, $[(-0.01), (-1,0)] =$ negativo e $[(0,0)] =$ neutro. A subjetividade trabalha com intervalo de $[(0,0), (1,0)]$, sendo 0,0 muito objetivo e 1,0 muito subjetivo [10]. A Tabela 3.7 apresenta a função utilizada para classificar os *tweets* conforme os valores da polaridade e subjetividade.

Tabela 3.7 – Função para classificação dos *tweets*.

Exemplo de Entrada dos Dados	<i>Brazil is an excellent soccer team :) !!!</i>
Classificação dos Dados (Polaridade e Subjetividade)	<pre>tweet = TextBlob("Brazil is an excellent soccerteam :) !!!") translation_correction(tweet) tweet_clean_stopwords(tweet) tokenization(tweet) if tweet.sentiment.polarity > 0: print (tweet.sentiment) print (Polarity: Positive) elif tweet.sentiment.polarity == 0: print (tweet.sentiment) print (Polarity: Neutral) else: print (tweet.sentiment) print (Polarity: Negative)</pre>
Saída dos Dados	<i>Sentiment[(polarity=0,98828125), (subjectivity=1,0)]</i> <i>Polarity: Positive</i>

3.2.4 Fase 4: Camada de Armazenamento Distribuído

Esta camada faz a indexação e busca de grades volumes de dados. Este processo é feito por um *guest* separado na rede interna através da ferramenta *Elasticsearch*. Nesta fase,

o *Elasticsearch* realiza o armazenamento distribuído da estrutura completa dos dados do monitoramento em tempo real dos *tweets*, com dados que auxiliam no entendimento e na interpretação de comportamentos coletados do *Twitter* [32, 33, 43].

3.2.5 Fase 5: Camada de Visualização

A visualização dos *tweets* e *retweets* no ambiente tem como finalidade facilitar a interpretação dos analistas para que seja possível antecipar informações e propor medidas eficientes do ponto de vista da interpretação dos dados. Além disso, o monitoramento permite observar em tempo real *tweets*, *retweets*, menções, *hashtags*, relacionamentos entre entidades através de grafos, quantidade de *likes*, georreferenciamentos e sentimentos dos usuários sobre um tema.

Este processo é feito pela ferramenta *Kibana* [32]. Esta ferramenta fornece uma interface rica para permitir consultas analíticas avançadas, visualização e interação com os dados armazenados nos índices do *Elasticsearch* [32].

Aplicamos também nesta camada o conceito de grafo [102, 103] através do *plugin* *Kbn Network* [96] para facilitar a visualização dos dados através de relacionamentos entre as entidades de um índice. Este *plugin* fornece uma alternativa para extrair e resumir informações dos documentos e termos de um índice no *Elasticsearch*. Estes relacionamentos entre as entidades indexadas podem ser explorados para verificar as ligações mais significativas. No *Kbn Network* as entidades são chamadas de nós. A relação entre dois nós é uma conexão (aresta) na qual se resume aos documentos que contenham entidades de ambos os nós.

3.3 EFICIÊNCIA DA ARQUITETURA DE COLETA

Atualmente o *Twitter* oferece novas oportunidades para órgãos de Governo e empresas extraírem informações de grande relevância para suas estratégias de interesse, isso porque compartilhar livremente ideias e opiniões em larga escala nessa rede social tornou-se uma atividade comum.

Em razão desse tipo de comportamento, esse trabalho pode ser utilizado por analistas para coletar, pesquisar, analisar e visualizar dados do *Twitter* em tempo real, indiferente de sua atividade. No caso desta proposta, diferentemente de outras plataformas abertas estudadas, existe a aplicação de técnicas de análise de sentimento, anonimização da coleta e isolamento de camadas, transformando o conceito em uma solução modular, adicionando ou removendo componentes conforme a necessidade.

O ambiente passou por diversos testes de implementação antes de ser colocado em pro-

dução, a fim de minimizar erros e falsos positivos. A solução foi instalada no laboratório de pesquisa da Universidade de Brasília. Após os testes, dois estudos de casos foram utilizados na arquitetura de coleta para avaliar a eficácia e eficiência do classificador de sentimento léxico implementado na subcamada de classificação do *OctopusViz*: Copa do Mundo FIFA 2018 e COVID-19.

É importante mencionar que o *framework* desenvolvido e os métodos aplicados para a análise dos estudos de casos não estão restritos a esses assuntos em particular. Decidimos usar os estudos de casos Copa do Mundo FIFA 2018 e COVID-19, porque eles trariam uma visão neutra sobre política, empresas, religião ou discussões sobre cores.

3.3.1 Estudo de Caso I: Copa do Mundo FIFA 2018

Este estudo de caso tem como objetivo observar através de métricas, estatísticas e sentimento a repercussão e a opinião pública dos usuários no *Twitter* sobre o tema "**seleção brasileira**" durante os jogos da Copa do Mundo FIFA 2018.

A Copa do Mundo FIFA 2018 aconteceu na Rússia entre os dias 14 de junho de 2018 e 15 de julho de 2018 [104]. Foi a vigésima primeira edição deste evento esportivo. O torneio foi realizado com a participação de 32 países, incluindo o Brasil, com jogos em 11 cidades.

3.3.1.1 Coleta dos Dados

A coleta dos dados ocorreu entre os dias 15 junho de 2018 e 31 de julho de 2018. O ambiente seguiu a lógica *booleana*, utilizada pela função de busca do *Twitter*. As palavras-chave utilizadas para a coleta foram: "seleção brasileira" ou "seleção do brasil".

3.3.1.2 Apresentação do Resumo Geral da Coleta

A Figura 3.2 representa através da camada de visualização o conjunto total de dados coletados pelo ambiente. Na Figura 3.2(a) pode-se verificar picos de *tweets* e *retweets* que foram publicados nos dias dos jogos do Brasil (17/junho (31.957), 22/junho (33.949), 27/junho (35.204), 02/julho (34.968) e 06/julho (25.506), sendo detalhada na Seção 3.3.1.9 a *hashtag* mais comentada sobre o tema nas quartas de final, na Seção 3.3.1.10 a identificação de *outliers* (usuários com atividades discrepantes por um período de tempo) e na Seção 3.3.1.11 a análise de uma rede de *bots* usada para propagar *tweets* e *retweets*. Observa-se que o maior pico de *tweets* e *retweets* publicados aconteceu no dia 07/julho (42.733), um dia depois das quartas de final (06/julho), jogo em que a Seleção do Brasil foi eliminada da Copa do Mundo da FIFA 2018. Nota-se também por outro lado uma queda depois do dia 10/julho indicando que o tema "**seleção brasileira**" não estava mais tão presente no *Twitter*. É possível identifi-

car também que 122.975 usuários publicaram 730.850 mensagens e 4.240 *hashtags* (Figuras 3.2(b) e 3.2(c)).

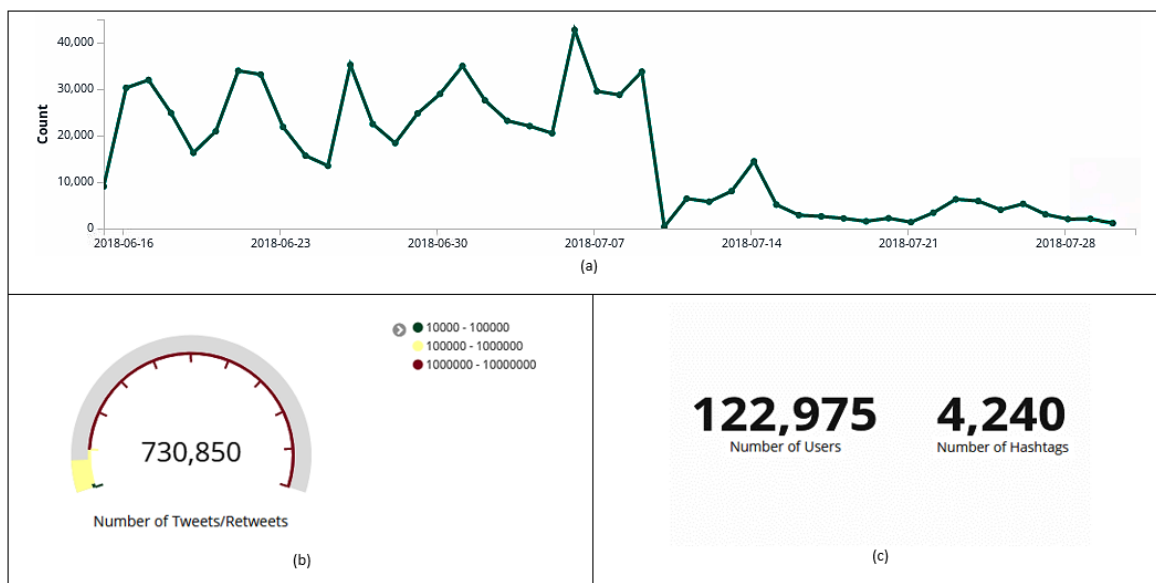


Figura 3.2 – Dados coletados no *Twitter* entre os dias 15 junho de 2018 e 31 de julho de 2018. *Histogram* com a quantidade de *tweets* e *retweets* coletados por dia (a). Quantidade total de *tweets* e *retweets* (b). Quantidade total de usuários e *hashtags* (c).

3.3.1.3 Análise de Tweets e Retweets

Classificar as mensagens em *tweets* e *retweets* é relevante para que os analistas tenham um entendimento das mensagens que podem ser estudadas de acordo com o interesse. Além disso, existe a consideração da influência sobre um determinado assunto, ou seja, um usuário pode manipular um determinado tema, apenas enviando uma grande quantidade de *retweets*.

Na Figura 3.3(a) é possível observar que das 730.850 mensagens, 168.509 foram classificadas como *tweets* e 562.341 como *retweets*. A Figura 3.3(b) apresenta a quantidade de *tweets* e *retweets* publicados por dia. Verifica-se que durante todo o período da coleta, em todos os dias, a quantidade de *retweets* superou a de *tweets*, destacando-se os dias 22/junho (rodada 2 de 3 do grupo E: Brasil 2 x Costa Rica 0) com 27.971 *retweets*, 02/julho (oitavas de final: Brasil 2 x México 0) com 26.928 *retweets* e 07/julho (um dia após as quartas de final: Brasil 1 x Bélgica 2) com 35.709 *retweets*. Nota-se também uma diferença (29.105 *retweets* - 4.647 *tweets*) no dia da semifinal (10/julho), onde a seleção que eliminou o Brasil (Bélgica) nas quartas de final perdeu para a França por 1 x 0.

Como observação, cabe ressaltar, que o carácter especial # é retirado das *hashtags* antes da indexação dos dados na camada de armazenamento distribuído (Seção 3.2.4). Este processo é feito na subcamada de processamento de dados (Seção 3.2.2) para que os *tweets* e *retweets* sejam classificados pela subcamada de classificação (Seção 3.2.3).

Tabela 3.8 – As cinco *hashtags* mais referenciadas entre os dias 15 junho de 2018 e 31 de julho de 2018.

Hashtags	Tweets	Retweets	Total
#Copa2018	1.356	3.193	4.549
#BRA	575	2.701	3.276
#VaiBrasil	100	1.062	1.162
#BrasilGanha	95	960	1.055
#BRAMEX	123	821	944

3.3.1.5 Aplicação de Filtros

Através da interação fornecida pela ferramenta foi possível aplicar filtros para criar e separar nuvens de *hashtags* por *tweets* e *retweets*. Observamos que as *hashtags* mais referenciadas dependem também do interesse do analista em *tweets* (Figura 3.5(a)) ou *retweets* (Figura 3.5(b)). Verifica-se nas duas primeiras linhas dos dados representados na Tabela 3.9 que as *hashtags* #Copa2018 e #BRA foram as mais comentadas tanto nos *tweets* quanto nos *retweets*. Já na terceira linha, observa-se que as *hashtags* mudam conforme esse interesse.

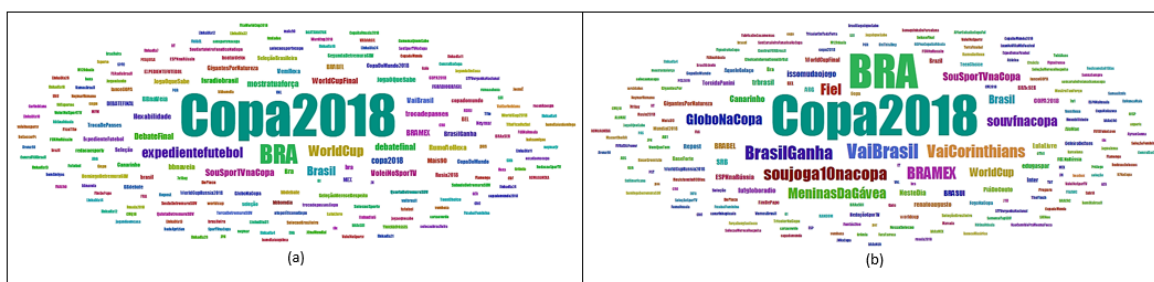


Figura 3.5 – Nuvem com as *hashtags* mais referenciadas. (a) indica *tweets* e (b) *retweets*.

Tabela 3.9 – *Hashtags* mais referenciadas por *tweets* ou *retweets*.

Hashtags	Tweets	Hashtags	Retweets
#Copa2018	1.356	#Copa2018	3.193
#BRA	575	#BRA	2.701
#WorldCup	277	#VaiBrasil	1.062
#expedientefutebol	273	#BrasilGanha	960
#Brasil	238	#soujoga10nacopa	923

3.3.1.6 Análise de Usuários

Analisar usuários é importante na perspectiva de um analista, tendo em vista que no *Twitter*, existem muitas contas falsas, sendo algumas *bots* e outras com finalidade escusa. Em geral, um usuário comum é incapaz de postar muitas mensagens em um curto espaço de tempo. Além disso, é prática comum no *Twitter* encontrar contas falsas, *bots*, ou estratégias para difusão de informações falsas (*Fake News*) ou até mesmo utilizada com o intuito de acompanhar atividades de usuários sem que os mesmos percebam.

A Figura 3.6 mostra os treze usuários que mais enviaram *tweets* e *retweets* entre os dias 15 junho de 2018 e 31 de julho de 2018. Nota-se na Tabela 3.10, que neste período, o usuário que mais publicou *retweets* no conjunto de dados foi *InfosFutebol*. Apesar de publicar apenas 34 *tweets*, esta conta acumulou mais de 35 mil *retweets* (4,857% do total de dados coletados). É possível observar também na Tabela 3.10 que os usuários *Allec_Matheus* e *rosedixdelrey* não publicaram *tweets*, mas eles retuítaram muito.

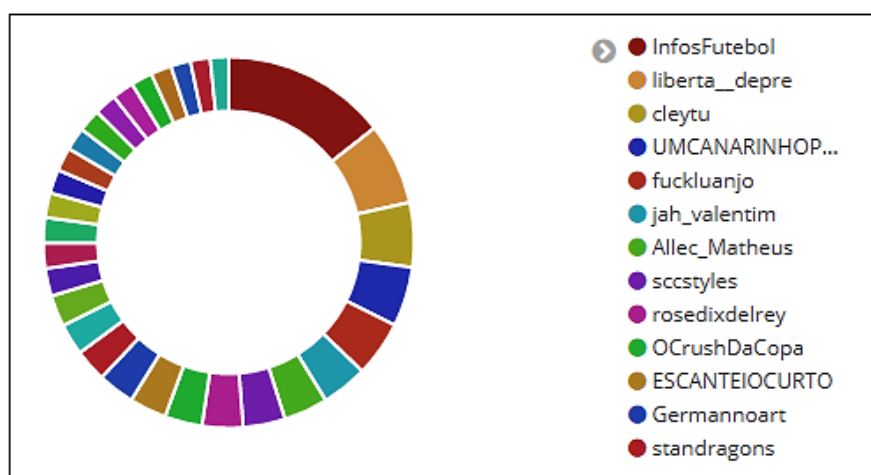


Figura 3.6 – Usuários que mais publicaram *tweets* e *retweets*.

Tabela 3.10 – Classificação dos *tweets* e *retweets* por usuário.

Usuários	Tweets	Retweets	Total
<i>InfosFutebol</i>	34	35.504	35.538
<i>liberta_depre</i>	16	17.575	17.591
<i>cleytu</i>	2	13.856	13.858
<i>UMCANARINHOPUTO</i>	7	12.799	12.806
<i>fuckluanjo</i>	1	11.997	11.998
<i>jah_valentim</i>	2	9.899	9.901
<i>Allec_Matheus</i>	0	9.529	9.529
<i>sccstyles</i>	4	9.046	9.050
<i>rosedixdelrey</i>	0	8.776	8.776
<i>OCrushDaCopa</i>	1	8.073	8.074
<i>ESCANTEIOCUTO</i>	3	7.986	7.989
<i>Germannart</i>	1	7.901	7.902
<i>standragons</i>	14	6.971	6.985

O ambiente permite identificar também as menções, a quantidade de *likes* e as *hashtags* citadas em cada *tweet* e *retweet*. Observa-se na Figura 3.7, quatro *retweets* que foram publicados pelo usuário *InfosFutebol*. Interessante notar a quantidade de *likes* (definidas pela coluna *favorite_count* - 14.975, 10.462, 12.679 e 9.999) que cada *retweet* recebeu.





image	username	text	mentions	favorite_count	is_retweet	hashtags
	InfosFutebol	O MELHOR JOGADOR DA SELEÇÃO BRASILEIRA SE CHAMA PHILIPPE COUTINHO!!!! https://t.co/DAQShcWhpo	InfosFutebol	14,975	true	-
	InfosFutebol	Tetracampeão italiano, tricampeão brasileiro, artilheiro do mundo em 2005, 29 gols e 2 títulos em 50 jogos pela sel... https://t.co/VcWbXmg6ro	InfosFutebol	10,462	true	-
	InfosFutebol	RT se você acha que essa dupla deve ser titular da seleção brasileira! https://t.co/QDpwmrTAKM	InfosFutebol	12,679	true	-
	InfosFutebol	Atualmente, Philippe Coutinho é o melhor jogador da seleção brasileira.	InfosFutebol	9,999	true	-

Figura 3.7 – *Retweets* publicado pelo usuário *InfosFutebol*.

3.3.1.7 Análise de Sentimentos

A análise de sentimentos tem como finalidade classificar a opinião pública dos usuários nos *tweets* e *retweets*, a fim de identificar o tipo de discurso, que permita tomar decisões específicas. Esta classificação é feita na subcamada de classificação pelo algoritmo *Pattern Analyzer* (Seção 3.2.3) e indexada na camada de armazenamento distribuído (Seção 3.2.4).

As Figuras 3.8, 3.9 e 3.10 indicam o sentimento dos usuários sobre o tema "**seleção brasileira**" nos *tweets* e *retweets* entre os dias 15 junho de 2018 e 31 de julho de 2018.

Conforme a Figura 3.8(a), o maior pico de *tweets* e *retweets* classificados como positivo (21.146) aconteceu no dia 07/julho, um dia depois em que a Seleção da Bélgica ganhou da Seleção do Brasil por 2x1 nas quartas de final. Nota-se que mesmo perdendo o jogo, os usuários foram favoráveis a seleção brasileira. Verifica-se também na Figura 3.8(a) que o maior pico com a polaridade negativa aconteceu no dia 22/junho (13.007) durante a rodada 2 de 3 do grupo E: Brasil 2 x Costa Rica 0. Provavelmente essa repercussão negativa, deve-se ao fato do Brasil ter empatado o jogo no dia 17/junho durante a rodada 1 de 3 do grupo E: Brasil 1 x Suíça 1.

Na Figura 3.8(b) é possível identificar também neste período, picos de *tweets* e *retweets* por hora. Observa-se que o maior pico com a polaridade positiva (1.128) aconteceu antes do primeiro jogo da Seleção do Brasil (18:00hs do dia 16/junho).

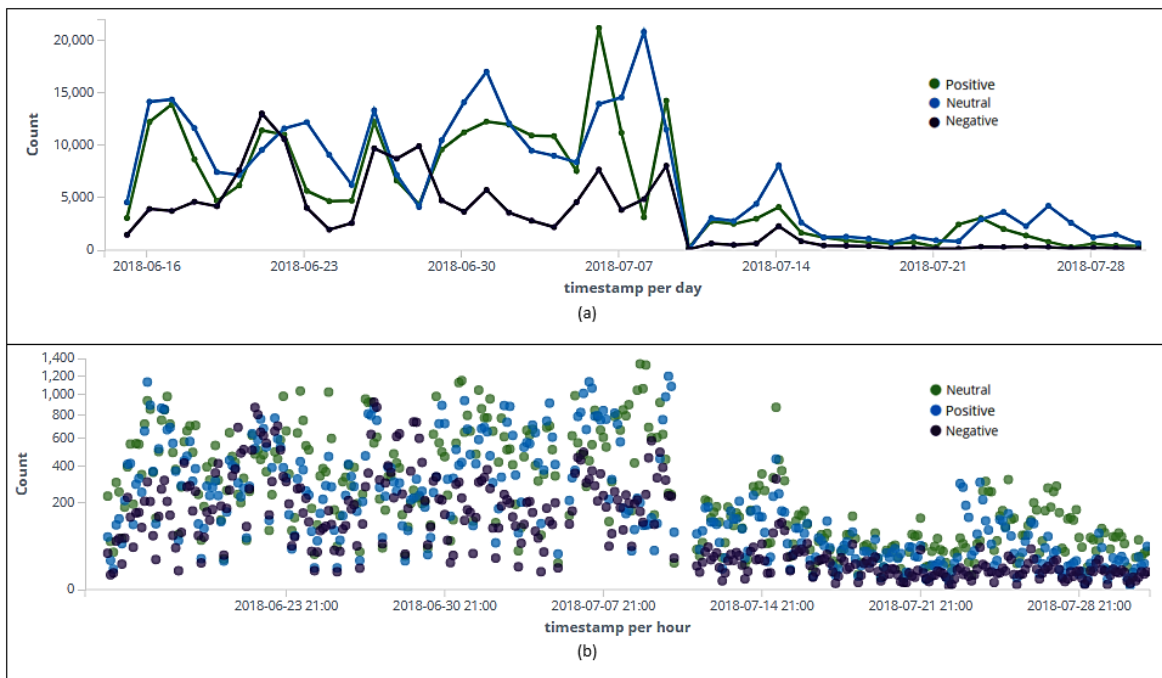


Figura 3.8 – Classificação (positivo, negativo e neutro) dos *tweets* e *retweets* por dia (a). Classificação (positivo, negativo e neutro) dos *tweets* e *retweets* por hora (b).

Separamos também os *tweets* dos *retweets* para identificar a classificação conforme as polaridades. Interessante notar que durante todo o período da coleta a quantidade de *retweets* publicados foi maior em todas as polaridades (neutra Figura 3.9(a), positiva Figura 3.9(b) e negativa Figura 3.9(c)). A Tabela 3.11 mostra os dias (22/junho, 07/julho e 09/julho) que foram publicados mais *tweets* e *retweets* conforme o sentimento dos usuários.

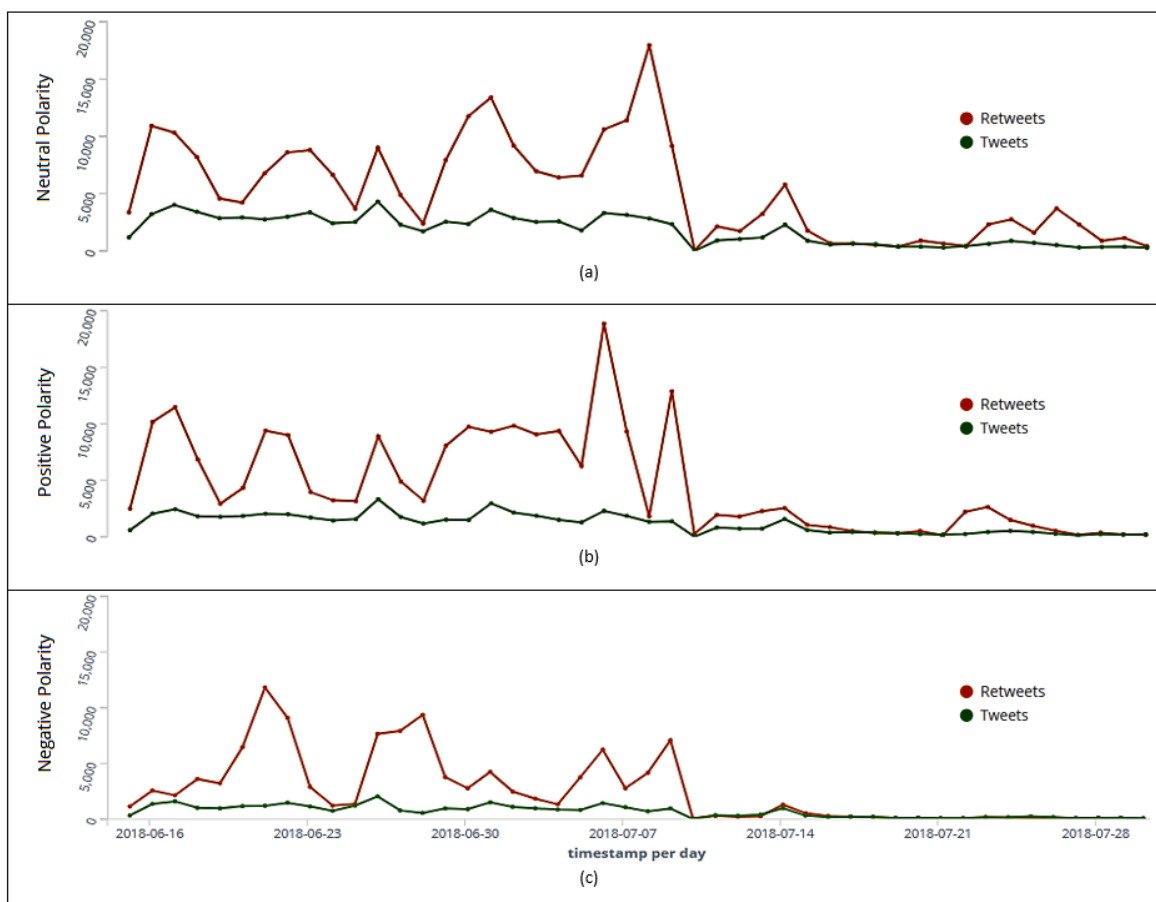


Figura 3.9 – Classificação dos *tweets* e *retweets* por dia com as polaridades neutra (a), positiva (b) e negativa (c).

Tabela 3.11 – Picos das polaridades dos *tweets* e *retweets* por dia.

Polaridade	Dia	Tweets	Retweets	Total
Neutro	09/julho	2.824	17.946	20.770
Positivo	07/julho	2.291	18.855	21.146
Negativo	22/junho	1.196	11.811	13.007

Considerando a quantidade total de *tweets* e *retweets* (730.850), a análise de sentimentos apontou através do algoritmo *Pattern Analyzer* (Figura 3.10) que 36,05% (263.485) dos usuários são favoráveis a seleção brasileira, 20,04% (146.445) parecem ser contra e que 43,91% (320.920) demonstram serem neutros (Tabela 3.12).

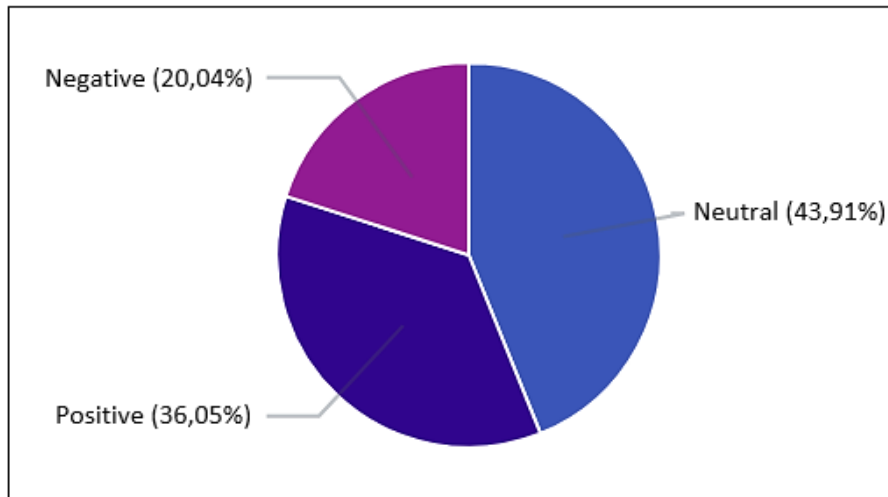


Figura 3.10 – Classificação geral dos *tweets* e *retweets* conforme o algoritmo *Pattern Analyzer*.

Tabela 3.12 – Polaridade dos *tweets* e *retweets*.

Polaridade	Tweets	Retweets	Total
Positivo	53.993	209.492	263.485
Negativo	31.230	115.215	146.445
Neutro	83.286	237.634	320.920

3.3.1.8 Análise de Vínculos

A análise de vínculos tem como objetivo integrar informações de múltiplas entidades (usuários, *hashtags*, *tweets*, *retweets*, menções, polaridade dos sentimentos e imagens) do *Twitter* para depurar, organizar e interpretar dados brutos, que permite ao analista detectar padrões e relacionamentos existentes.

A Figura 3.11(a) apresenta um grafo entre as entidades polaridade (positiva, negativa e neutra) e usuário. Para esta análise selecionamos de forma aleatória três mil usuários indexados no ambiente. Percebe-se no grafo da Figura 3.11(a) que mais usuários estão relacionados com as polaridades neutra e positiva. Verifica-se também que vários usuários têm relação com duas ou três polaridades diferentes (positiva, negativa e neutra). Neste caso, alguns *tweets* e *retweets* destes usuários foram classificados como positivos e outros como negativos e neutros.

A Figura 3.11(b) mostra a entidade polaridade positiva no centro do grafo. Conforme o algoritmo *Pattern Analyzer*, os usuários *InfosFutebol*, *fuckluanjo*, *rosedixdelrey*, *dobresdelena* e *bbru_no* (mais próximos do centro) publicaram mais *tweets* e *retweets* favoráveis a seleção brasileira.

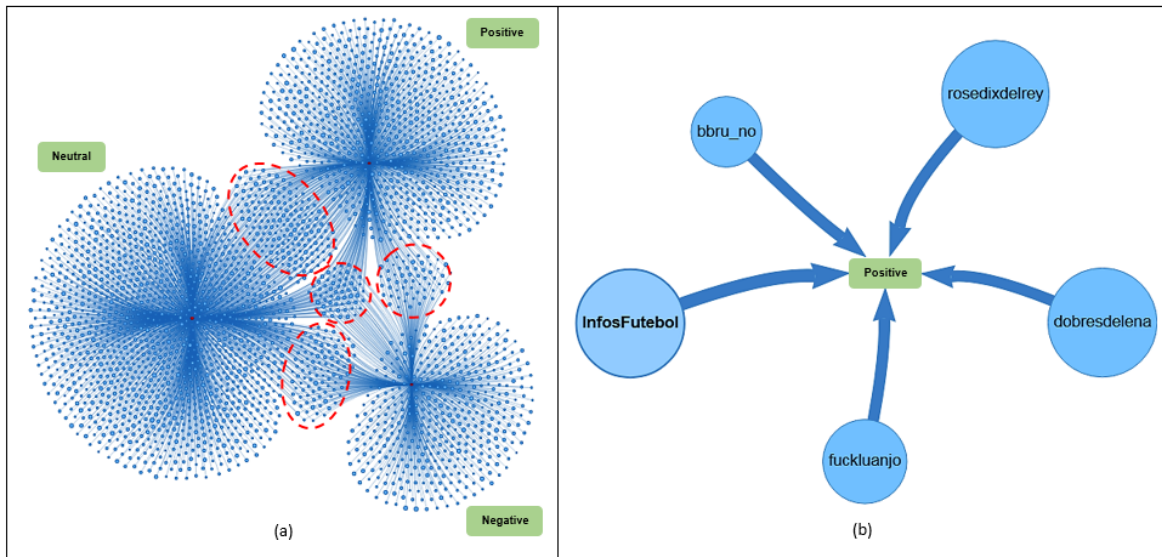


Figura 3.11 – Relacionamento entre as entidades polaridade (positiva, negativa e neutra) e usuário (a). Usuários que mais publicaram *tweets* e *retweets* com a polaridade positiva (b).

Outra análise de vínculo foi feita entre as entidades *hashtag* e usuário. Nesta análise, utilizamos todos os dados indexados no ambiente (15 junho de 2018 à 31 de julho de 2018). Nota-se através da Figura 3.12(a) uma grande dificuldade de identificar essas entidades devido a quantidade de relacionamentos existentes. A ferramenta contorna este problema através da aplicação de filtros de aproximação. Assim, foi possível observar que a *hashtag* #*copa2018* aparece no centro do grafo como a mais referenciada nos *tweets* e *retweets* (Figura 3.12(b)).

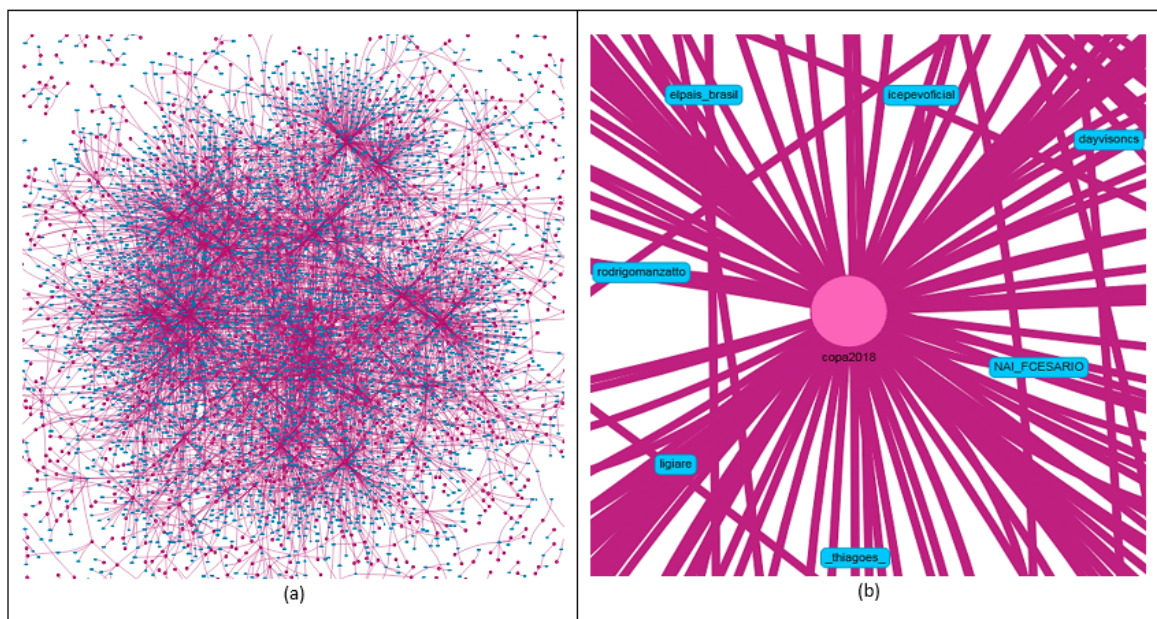


Figura 3.12 – Relacionamento entre as entidades *hashtag* e usuário (a). *Hashtag* mais referenciada nos *tweets* e *retweets* (b).

3.3.1.9 Análise da Hashtag mais Comentada nas Quartas de Final

No dia 06 de julho, durante o quinto jogo da seleção brasileira, nas quartas de final, verificou-se que a *hashtag* #Copa2018 estava em destaque no *Dashboard*. Para se obter informações mais detalhadas, aplicou-se um filtro nesta *hashtag*. O resultado do filtro e as respectivas informações decorrentes podem ser observadas conforme se segue: observa-se nas Figuras 3.13(a) e 3.13(b) que 1.022 usuários publicaram 4.098 *tweets* e *retweets* com esta *hashtag*; outras 368 *hashtags* (Figura 3.13(a)) estão relacionadas com #Copa2018; nota-se na Figura 3.13(d) que 2.869 mensagens com essa *hashtag* foram classificadas como *retweets* e 1.229 como *tweets*; o usuário *torcidasfotos* (Figura 3.13(e)) foi quem mais enviou mensagens com essa *hashtag*; verifica-se na Figura 3.14(a) que 31,92% dos usuários são a favor, 13,88% contra e 54,2% neutros; a *hashtag* #Copa2018 (Figura 3.14(b)) foi classificada em três polaridades diferentes (positiva, negativa e neutra); na Figura 3.14(c) é possível identificar que o maior pico de *tweets* e *retweets* com essa *hashtag* aconteceu no dia 27/junho (773).

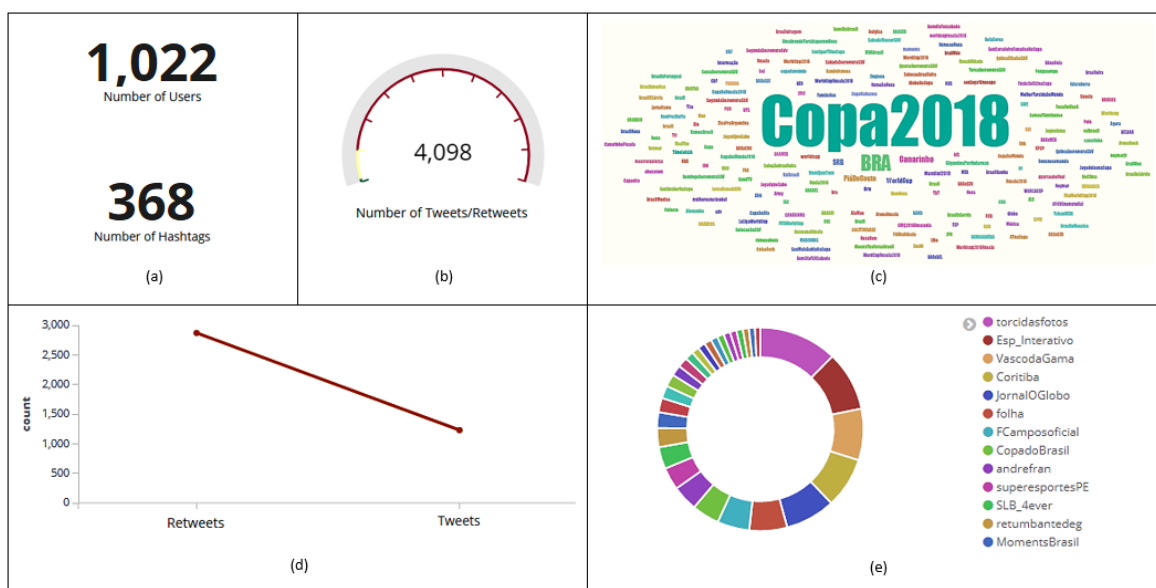


Figura 3.13 – Quantidade de usuários, *hashtags* (a), *tweets* e *retweets* (b). *Hashtag* mais comentada (c). Classificação das mensagens que foram incluídas com essa *hashtag* em *tweets* e *retweets* (d). Usuários que mais enviaram mensagens com essa *hashtag* (e).

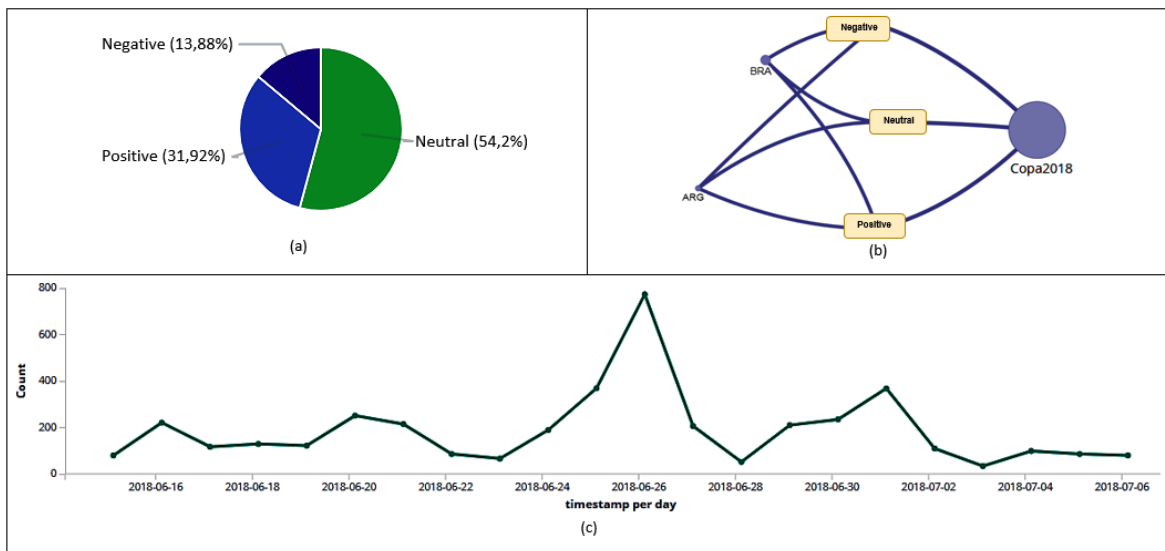


Figura 3.14 – Polaridade dos *tweets* e *retweets* (a). Grafo com o relacionamento entre as entidades *hashtag* e polaridade (b). *Histogram* com a quantidade de *tweets* e *retweets* coletados por dia com essa *hashtag* (c).

3.3.1.10 Análise de Outliers

Para este trabalho, consideramos *outliers* os elementos que não seguem um padrão do conjunto de usuários ao qual eles foram agrupados segundo os critérios de interesse da análise. São usuários com atividades discrepantes por um período de tempo que requerem atenção especial, pois normalmente produzem valores com efeitos não confiáveis.

No dia 21/junho, antes do segundo jogo da Seleção do Brasil na Copa do Mundo FIFA 2018, verificou-se que a conta *dobresdelena* apresentava um grande afastamento dos outros usuários, sendo considerada pela análise um *outlier* (Figura 3.15). Conforme a Tabela 3.13, os usuários (*dobresdelena*, *lorenzopaag*, *whindersson*, *cleytu*, *adrianowilkson* e *laxarruda*) não publicaram *tweets*. Nota-se também que o usuário *dobresdelena* publicou 6.723 *retweets*. Uma análise detalhada foi conduzida para tentar identificar as características deste usuário.

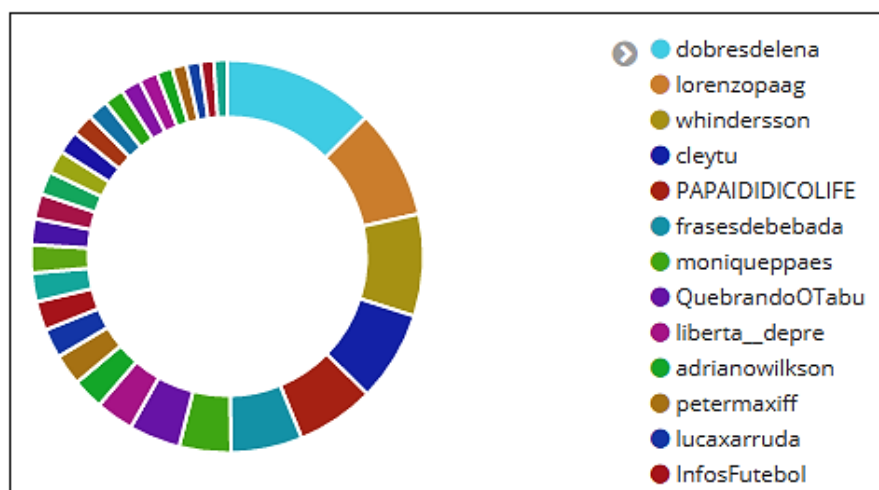


Figura 3.15 – Dados coletados no *Twitter* entre os dias 15 junho de 2018 e 21 de junho de 2018. Usuários discrepantes.

Tabela 3.13 – Quantidade de *tweets* e *retweets* por usuário.

Usuários	Tweets	Retweets	Total
<i>dobresdelena</i>	0	6.723	6.723
<i>lorenzopaag</i>	0	4.877	4.877
<i>whindersson</i>	0	4.526	4.526
<i>cleytu</i>	0	4.003	4.003
<i>PAPAIDIDICOLIFE</i>	2	3.481	3.483
<i>frasesdebebada</i>	1	3.170	3.171
<i>moniqueppaes</i>	1	2.299	2.300
<i>QuebrandoOTabu</i>	1	2.279	2.280
<i>liberta_depre</i>	5	1.686	1.691
<i>adrianowilkson</i>	0	1.388	1.388
<i>petermaxiff</i>	1	1.385	1.386
<i>lacaxarruda</i>	0	1.296	1.296
<i>InfosFutebol</i>	5	1.287	1.292

As Figuras 3.16 e 3.17, apresentam informações sobre usuário, *tweets*, *retweets*, *hashtags* e sentimento (polaridade das mensagens). Interessante verificar que todos os *retweets* foram classificados pelo algoritmo *Patterns Analyzer* como positivo, influenciando a análise sobre o sentimento dos usuários em relação ao tema "**seleção brasileira**". Detalhes das informações podem ser observadas conforme se segue: na Figura 3.16(a) observa-se que o usuário *dobresdelena* não publicou nenhuma *hashtag*; verifica-se na Figura 3.16(d) que todas as mensagens publicadas foram classificadas como *retweets*; nota-se que o maior pico de *retweets* (Figura 3.16(e)) aconteceu no dia 18/junho (3.917); 100% dos *retweets* foram classificados como positivos (Figuras 3.17(a) e 3.17(b)); o maior pico de *retweets* (374) classificados como positivo aconteceu no dia 17/junho às 22:00hs (Figura 3.17(c)).

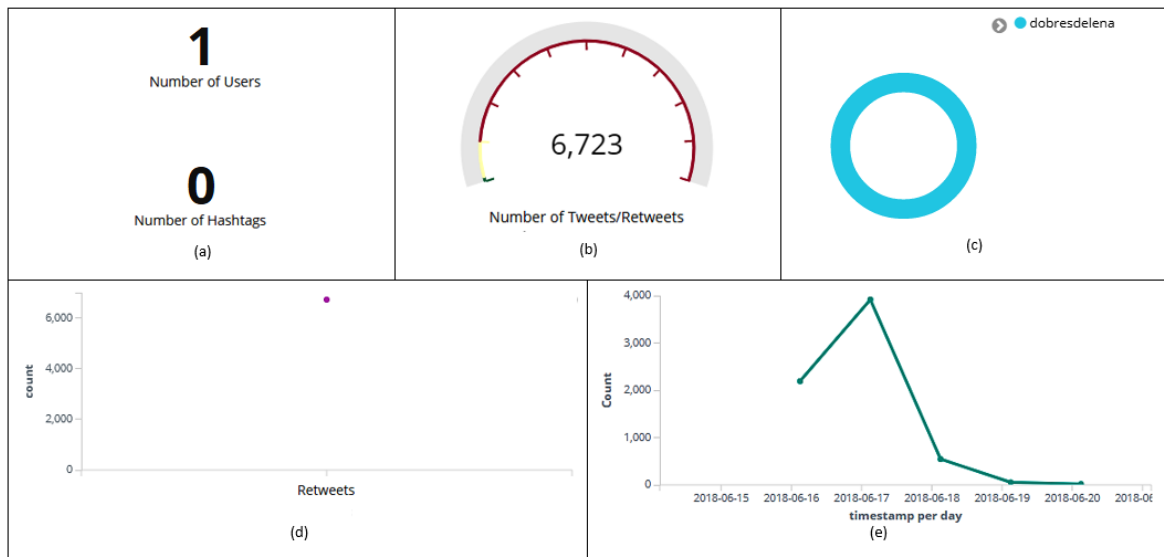


Figura 3.16 – *Dashboard* com informações do usuário *dobresdelena*. Quantidade de *hashtags* (a), *tweets* e *retweets* (b). Usuário discrepante (c). Quantidade de mensagens classificadas como *retweets* (d). *Histogram* com a quantidade de *retweets* coletados por dia (e).

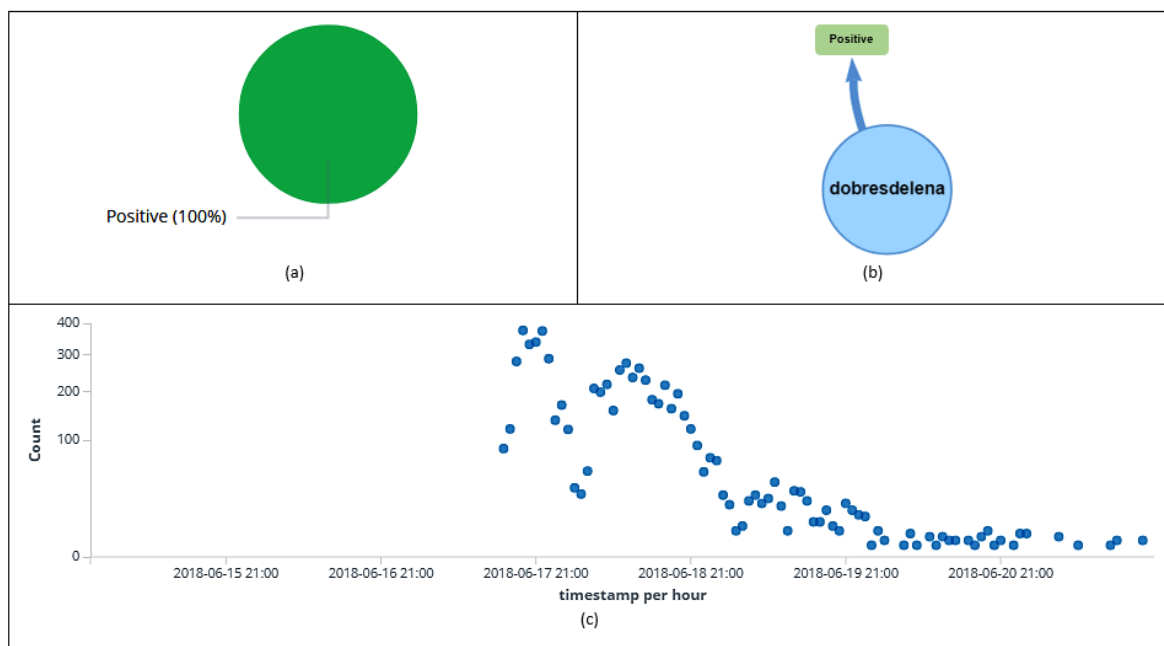


Figura 3.17 – Informações sobre a polaridade dos *retweets* do usuário *dobresdelena* (a). Relação entre as entidades polaridade (positiva) e o usuário (b). Classificação (positiva) de *retweets* por hora (c).

3.3.1.11 Análise de Botnet

Botnets são contas controladas por algoritmos para realizar funções repetitivas (*retweetar* conteúdo, responder e enviar mensagens diretas a novos seguidores) ou executar tarefas complexas (conversas *online*) nas mídias sociais. A capacidade de controlar remotamente

grandes quantidades de agentes autônomos no *Twitter*, mostrou ser uma poderosa ferramenta para execução de atividades, como a produção de *spam*, seguidores falsos, manipulação de debates e opinião pública [17].

Neste cenário, analisamos a conta no *Twitter* do usuário *dobresdelena* para conseguir detalhes pertinentes ao interesse da análise. A Figura 3.18 mostra que até o momento desta análise o último *retweet* publicado por este usuário estava com 36.334 *likes*. Nota-se que as publicações destes *retweets* são feitas sempre no mesmo horário com variações de segundos (17/junho 23:56:36, 18/junho 23:56:36 e 19/junho 23:56:25). É possível identificar também que nos *retweets* existe um *link* (<https://t.co/IIYLzqYHXF>) que apresenta três fotos do goleiro da seleção brasileira que foi utilizado como estratégia da conta para ganhar *likes*.





▶	June 21st 2018, 19:38:45.548		dobresdelena	socorro o goleiro da seleção brasileira é muito lindo tô até com falta d ar https://t.co/IIYLzqYHXF	36,334	positivo
▶	June 19th 2018, 23:56:25.981		dobresdelena	socorro o goleiro da seleção brasileira é muito lindo tô até com falta d ar https://t.co/IIYLzqYHXF	36,177	positivo
▶	June 18th 2018, 23:56:36.056		dobresdelena	socorro o goleiro da seleção brasileira é muito lindo tô até com falta d ar https://t.co/IIYLzqYHXF	34,042	positivo
▶	June 17th 2018, 23:56:36.547		dobresdelena	socorro o goleiro da seleção brasileira é muito lindo tô até com falta d ar https://t.co/IIYLzqYHXF	15,950	positivo

Figura 3.18 – *Retweets* publicados pelo usuário *dobresdelena*.

É comum *bots* utilizar fotografias de outras pessoas como avatares. De posse da imagem deste usuário, esta foi submetida para análise pelas ferramentas *Google Images* [100] e *TinEye* [105] com a finalidade de fazer uma pesquisa reversa e encontrar imagens semelhantes em outro lugar *online*.

Conforme a Figura 3.19(a), o perfil *dobresdelena* não apresenta um *ticket* azul para confirmar a veracidade da conta no *Twitter*. Além disso, as ferramentas *TinEye* e *Google Images* identificaram vários resultados de *sites* que utilizam a imagem com as mesmas características (Figuras 3.19(a) e 3.19(b)).

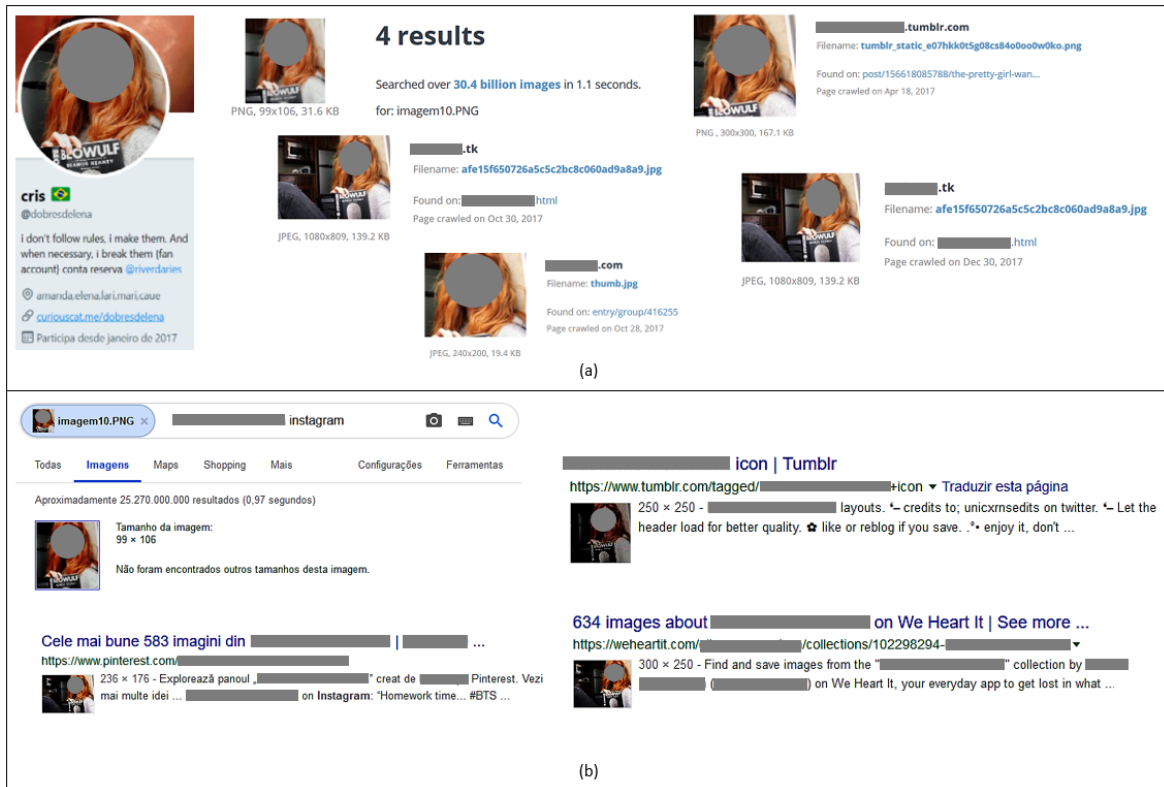


Figura 3.19 – Análise reversa da imagem nas ferramentas *TinEye* (a) e *Google Images* (b).

A ferramenta *TweetBotOrNot* [106] foi utilizada também para analisar a conta *dobresdelena*. Esta ferramenta utiliza aprendizado de máquina para analisar metadados e classificar o comportamento das contas no *Twitter*, informando se o usuário é uma *bot* [106]. A Figura 3.20 mostra que *TweetBotOrNot* classificou a conta *dobresdelena* como um possível *bot* (0,813).

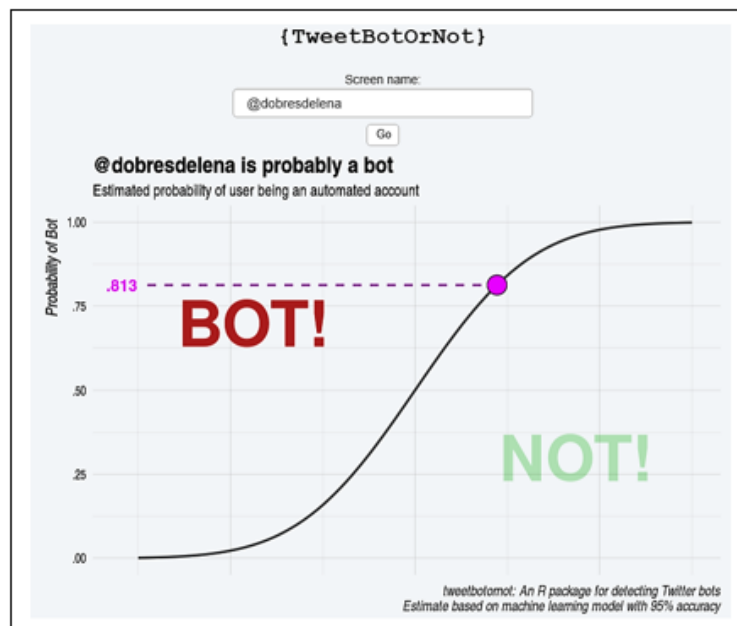


Figura 3.20 – Análise através da ferramenta *TweetBotOrNot*.

3.3.2 Estudo de Caso II: COVID-19

Este estudo de caso considerou a aplicação de técnicas de análise de sentimento para entender o comportamento dos usuários com relação ao tema "**coronavírus (COVID-19)**". O coronavírus ou COVID-19 é uma doença nova que se originou na cidade de Wuhan, província de Hubei, na China, em dezembro de 2019 [107]. A Organização Mundial da Saúde (OMS) caracterizou a doença como uma pandemia no dia 30 de janeiro de 2020 [108]. Atualmente a doença está afetando a economia, a saúde física e as condições de trabalho de milhares de pessoas em todo o mundo.

3.3.2.1 Coleta dos Dados

A coleta dos dados ocorreu durante o período da pandemia, entre os dias 01 abril de 2020 e 06 de setembro de 2020. A finalidade foi testar a eficiência e eficácia do classificador de sentimento implementado na subcamada de classificação do *OctopusViz* (Seção 3.2.3). As palavras-chave utilizadas durante a coleta foram: "covid-19", "covid19", "coronavírus", "coronavirus", "corona vírus" ou "corona virus".

3.3.2.2 Apresentação do Resumo Geral da Coleta

A camada de visualização do ambiente *OctopusViz* apresenta o conjunto total de dados. Na Figura 3.21(a) pode-se verificar picos de *tweets* e *retweets* que foram publicados por semana. Percebe-se na Figura 3.21(b) que os idiomas inglês (*en*), espanhol (*es*) e português

(pt) foram os mais utilizados para publicar os *tweets* e *retweets* sobre o tema. Conforme as Figuras 3.21(c) e 3.21(d), 1.739.816 usuários publicaram 8.397.140 mensagens e 321.303 *hashtags* sobre o tema coronavírus. Nota-se na Figura 3.21(e) que a *hashtag* #COVID19 foi a mais mencionada. Observa-se também que das 8.397.140 mensagens, 2.100.329 foram classificadas como *tweets* e 6.296.811 como *retweets* (Figura 3.21(f)). O usuário *kylegriffin1* (Figura 3.21(g)) foi quem mais publicou *tweets* e *retweets*.

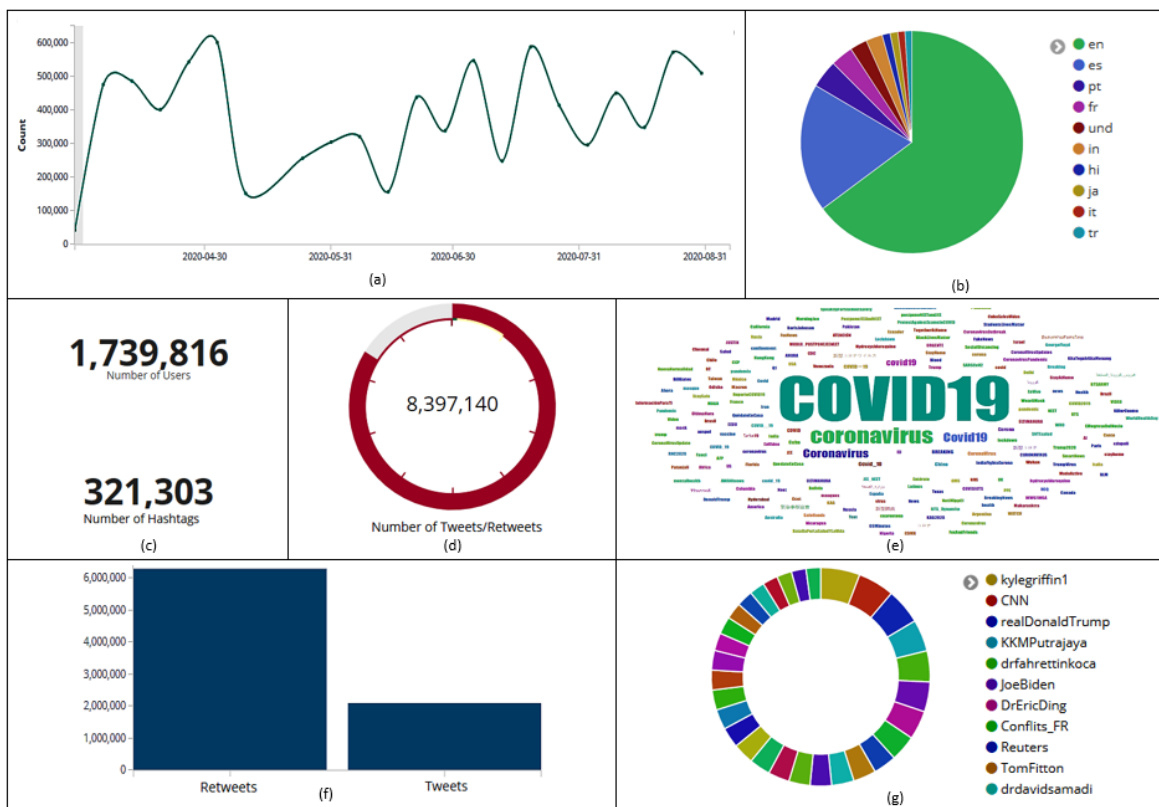


Figura 3.21 – Dados coletados no *Twitter* entre os dias 01 abril de 2020 e 06 de setembro de 2020. *Histogram* com a quantidade de *tweets* e *retweets* coletados por dia (a). Idiomas mais utilizados para publicar os *tweets* e *retweets* (b). Quantidade de usuários, *hashtags* (c), *tweets* e *retweets* (d). *Hashtag* mais comentada (e). Classificação das mensagens em *tweets* e *retweets* (f). Usuários que mais enviaram mensagens sobre o tema (g).

3.3.2.3 Análise de Sentimentos

Esta seção fornece uma discussão detalhada sobre os resultados obtidos da análise de sentimento. A Figura 3.22 indica o comportamento dos usuários por semana durante o período da pandemia (01 abril de 2020 e 06 de setembro de 2020). Observa-se ao longo da coleta que a quantidade de *tweets* e *retweets* publicados com a polaridade positiva foi maior do que a polaridade negativa.

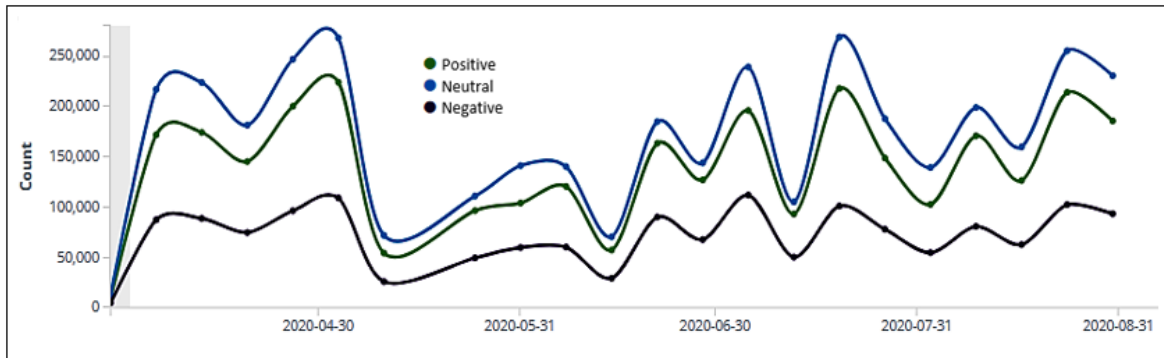


Figura 3.22 – Classificação dos *tweets* e *retweets* por semana.

A Figura 3.23 apresenta a classificação geral (polaridades positiva, negativa e neutra) dos *tweets* e *retweets* (8.797.140). Interessante notar que a quantidade de *retweets* publicados foi maior em todas as polaridades (positiva - 2.368.305, negativa - 1.207.152 e neutra - 2.709.425). Conforme o algoritmo *Pattern Analyzer*, a maioria dos *tweets* e *retweets* possui um sentimento positivo e neutro. A Tabela 3.14 mostrou que 36,65% (3.071.632) dos *tweets* e *retweets* foram classificados como positivo, 18,51% (1.551.562) como negativo e 44,84% (3.758.214) como neutros. Isso mostra que apesar dos problemas os usuários estão mais otimistas com relação a doença COVID-19.

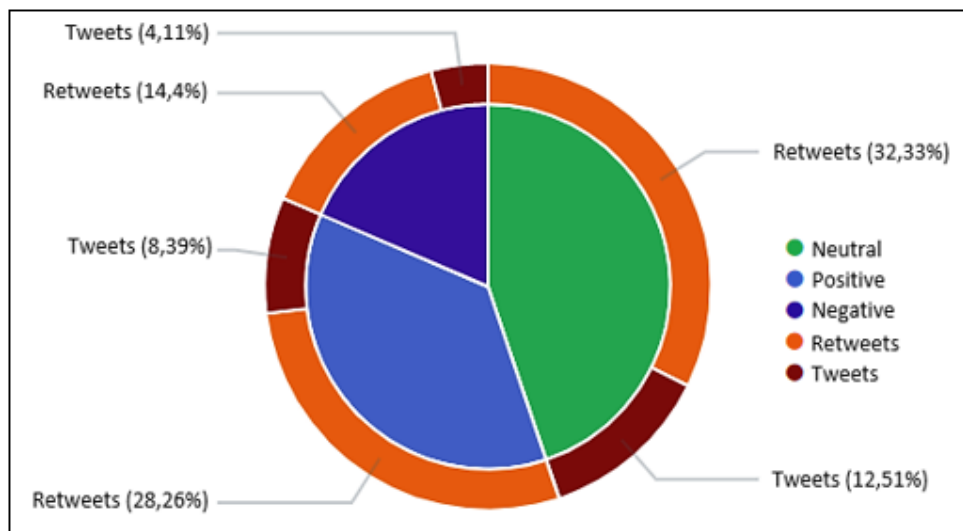


Figura 3.23 – Classificação geral dos *tweets* e *retweets* conforme o algoritmo *Pattern Analyzer*.

Tabela 3.14 – Polaridade dos *tweets* e *retweets*.

Polaridade	Tweets	Retweets	Total
Positivo	703.327	2.368.305	3.071.632
Negativo	344.410	1.207.152	1.551.562
Neutro	1.048.789	2.709.425	3.758.214

3.3.2.4 Análise de Sentimentos por Idioma

Conforme a Seção 3.2.2.1, o *OctopusViz* é capaz de traduzir e analisar mais de 100 pares de idiomas. Esta seção separou quatro idiomas (inglês - *en*, espanhol - *es*, português - *pt* e chinês - *zh*) que foram utilizados pelos usuários para publicar os *tweets* e *retweets*. O objetivo é extrair e analisar o sentimento desses usuários com relação ao coronavírus. O idioma chinês foi adicionado nesta análise devido a origem da doença.

Como exemplo, a Figura 3.24 mostra quatro *tweets* e suas respectivas polaridades atribuídas pelo classificador de sentimento que foram publicados com os idiomas inglês, espanhol, português e chinês.

▶ September 6th 2020, 12:42:43.006	en	CNBC		New for subscribers: Goldman gave clients a list of stocks that may surge if a coronavirus vaccine is approved.. https://t.co/Fs5Q5sK8V5	positivo
▶ September 6th 2020, 12:41:29.456	es	VTVcana18		Diosdado Cabello: la experiencia de ser positivo a COVID-19 es terrible, no me gustó sentir de cerca la muerte.. https://t.co/8uACP9ddrA	negativo
▶ August 28th 2020, 13:06:10.392	pt	SoCiencia		Não bastassem os vendilhões de falsas curas, agora grupo de pesquisadores e ativistas defende vacina "caseira" para.. https://t.co/D7h1Rdlezd	negativo
▶ August 28th 2020, 14:17:05.979	zh	CityofSanJose		出门进行必要活动时, 戴上脸罩以保护我们社区的弱势成员。结合实际的社交疏远和洗手, 面罩可以帮助减缓COVID-19的扩散。了解更多: https://t.co/LuXBW2HTom https://t.co/1Vi5R1RkmC	positivo

Figura 3.24 – *Tweets* publicados com os idiomas inglês, português e chinês.

A Figura 3.25 representa a quantidade de *tweets* e *retweets* publicados com cada idioma (inglês - *en*, espanhol - *es*, português - *pt* e chinês - *zh*). Na Tabela 3.15 é possível verificar uma grande quantidade de publicações de *tweets* e *retweets* com o idioma inglês (5.211.404) em comparação com os idiomas espanhol (1.802.811), português (325.645) e principalmente o chinês (6.265). Provavelmente esse fato deve estar relacionado com as restrições de acesso às redes sociais impostas pela China aos cidadãos chineses. Percebe-se também nos quatro idiomas que a quantidade de *retweets* publicados foi maior do que a quantidade de *tweets*.

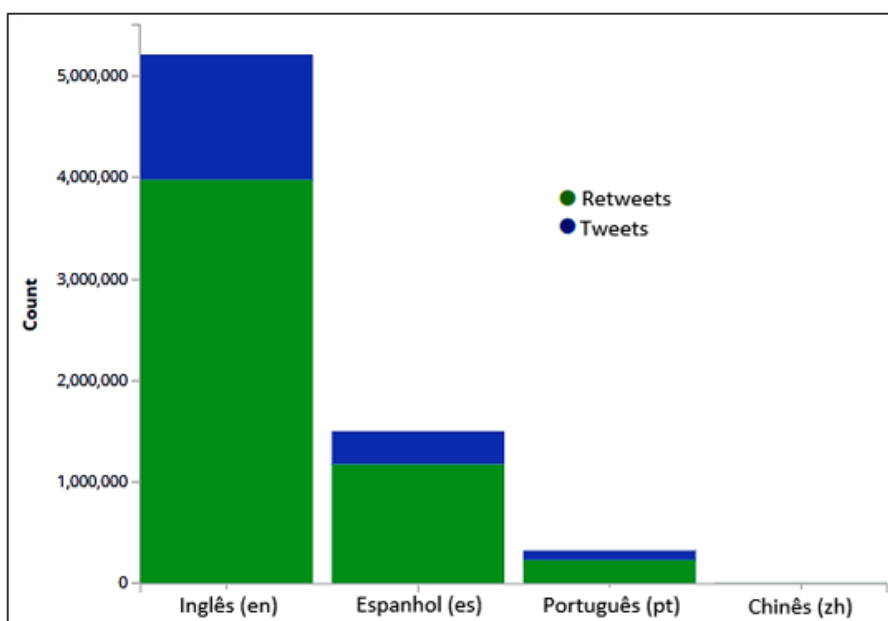


Figura 3.25 – Quantidade de *tweets* e *retweets* publicados pelos idiomas inglês, espanhol, português e chinês.

Tabela 3.15 – *Tweets* e *retweets* publicados com os idiomas *en*, *es*, *pt* e *zh*.

Idiomas	Tweets	Retweets	Total
Inglês - <i>en</i>	1.231.708	3.979.696	5.211.404
Espanhol - <i>es</i>	324.679	1.178.132	1.502.811
Português - <i>pt</i>	91.539	234.106	325.645
Chinês - <i>zh</i>	1.892	4.373	6.265

As Figuras 3.26(a), 3.26(b), 3.26(c) e 3.26(d) indicam o sentimento dos usuários que publicaram *tweets* e *retweets* com os quatro idiomas sobre o tema coronavírus. Nota-se em todos os idiomas um sentimento mais positivo do que negativo.

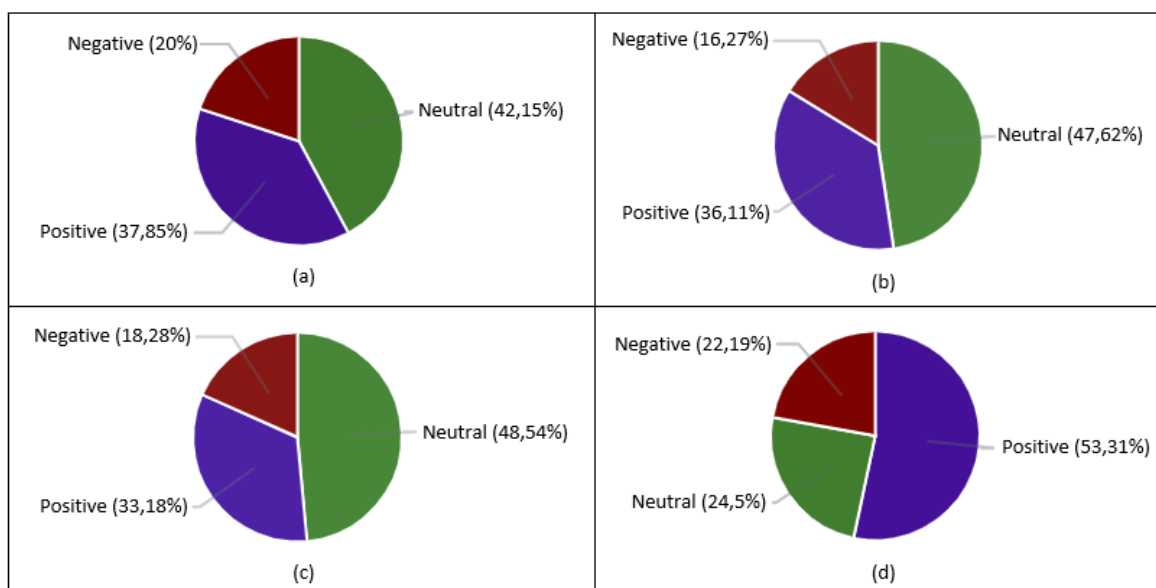


Figura 3.26 – Sentimento dos usuários que publicaram *tweets* e *retweets* com o idioma inglês (a). Sentimento dos usuários que publicaram *tweets* e *retweets* com o idioma espanhol (b). Sentimento dos usuários que publicaram *tweets* e *retweets* com o idioma português (c). Sentimento dos usuários que publicaram *tweets* e *retweets* com o idioma chinês (d).

O idioma inglês destaca-se pela quantidade de *tweets* e *retweets* classificados com a polaridade positiva. Conforme a Tabela 3.16, foram quase 50% a mais de *tweets* e *retweets* classificados com essa polaridade (37,85% - 1.975.669) em comparação com a polaridade negativa (20% - 1.043.711). Isso pode mostrar o que os usuários desse idioma estão realmente pensando sobre a doença. Provavelmente acreditam mais nas medidas adotadas pelos líderes mundiais para combater a pandemia coronavírus.

Tabela 3.16 – Polaridade dos *tweets* e *retweets* publicados com o idioma inglês.

Polaridade	Tweets	Retweets	Total
Positivo	445.973	1.529.696	1.975.669
Negativo	227.165	816.536	1.043.701
Neutro	560.462	1.637.837	2.198.299

Nas Tabelas 3.17 e 3.18 é possível verificar que os usuários dos idiomas espanhol e português também seguem o mesmo padrão dos usuários que utilizaram o idioma inglês. O sentimento desses usuários foram quase 50% mais positivo (espanhol - 36,11% - 545.989, português - 33,18% - 111.378) do que negativo (espanhol - 16,27% - 245.928, português - 18,28% - 60.916).

Tabela 3.17 – Polaridade dos *tweets* e *retweets* publicados com o idioma espanhol.

Polaridade	Tweets	Retweets	Total
Positivo	114.613	431.376	545.989
Negativo	51.404	194.524	245.928
Neutro	160.554	556.605	717.159

Tabela 3.18 – Polaridade dos *tweets* e *retweets* publicados com o idioma português.

Polaridade	Tweets	Retweets	Total
Positivo	29.495	81.883	111.378
Negativo	17.082	43.834	60.916
Neutro	46.854	112.762	159.616

Os *tweets* e *retweets* publicados com o idioma chinês foram 53,31% (3.340) mais positivos do que negativos (22,19% - 1.390). Apesar da COVID-19 ter sido originada na China, os usuários que utilizaram esse idioma também foram mais otimistas com relação a doença (Tabela 3.19).

Tabela 3.19 – Polaridade dos *tweets* e *retweets* publicados com o idioma chinês.

Polaridade	Tweets	Retweets	Total
Positivo	943	2.397	3.340
Negativo	385	1.005	1.390
Neutro	564	971	1.535

3.4 RESUMO DO CAPÍTULO 3

Este capítulo mostrou a necessidade de estudar e observar classificadores de sentimento antes de serem utilizados em sistemas de apoio à decisão. O desenvolvimento do *OctopusViz* foi dividido em cinco fases, buscando uma otimização dos aspectos envolvidos na arquitetura proposta. Na primeira fase, configurações no motor de busca foram feitas para coletar, anonimamente por meio de uma VPN, os dados no *Twitter*. Na segunda fase, foi implementado um *script python* para realizar a transformação e centralização dos dados. Na terceira fase, utilizamos o algoritmo *Pattern Analyzer* para realizar a análise de sentimentos dos *tweets* e identificar comportamentos que possam estimar a opinião pública dos usuários. Conforme Sohngir et al. [78], algoritmos que utilizam a abordagem léxica são mais favoráveis em tarefas que envolvem grandes quantidades de dados porque não precisam passar pelo processo de treinamento. Na quarta fase, automatizamos o armazenamento distribuído dos dados tex-

tuais para auxiliar o entendimento e interpretação dos dados coletados no *Twitter*. Na quinta fase, implementou-se uma solução para facilitar a interpretação dos analistas.

Na sequência, foram utilizados dois estudos de casos para testar a eficiência do classificador de sentimento léxico: Copa do Mundo FIFA 2018 e COVID-19. No primeiro estudo de caso foi possível observar e identificar o sentimento dos usuários com relação a seleção brasileira. O segundo estudo de caso mostrou que os usuários estavam mais otimistas com relação a doença COVID-19.

O algoritmo *Pattern Analyzer* foi implementado na subcamada de classificação da arquitetura *OctopusViz* para estudar, identificar vulnerabilidades e testar as estratégias de ataques adversariais. Conforme os Capítulos 4 e 5, o classificador de sentimento léxico *Pattern Analyzer* possui vulnerabilidades que podem ser exploradas por usuários mal-intencionados para alterar a percepção da aplicação que está utilizando esse classificador. Por exemplo, ataques como esse, poderiam alterar os resultados apresentados pelo *OctopusViz* no segundo estudo de caso (COVID-19), induzindo os tomadores de decisão com informações erradas – usuários com sentimento menos otimista com relação a pandemia coronavírus.

4

ENGENHARIA REVERSA NO ALGORITMO PATTERN ANALYZER DA BIBLIOTECA TEXTBLOB

Este capítulo apresenta com detalhes o funcionamento do classificador de sentimento léxico (*Pattern Analyzer*) da biblioteca *TextBlob*. Para tanto, foram feitos dois trabalhos de engenharia reversa para identificar vulnerabilidades. O primeiro foi no código do algoritmo *Pattern Analyzer* para entender como os valores da polaridade, subjetividade, intensidade e confiança eram atribuídos. O segundo foi no conjunto de dados utilizado como léxico pelo algoritmo *Pattern Analyzer*.

4.1 BIBLIOTECA TEXTBLOB

TextBlob [10] é uma biblioteca escrita em *Python* que utiliza as bibliotecas NLTK [99] e *Pattern* [101] para trabalhar com Processamento de Linguagem Natural (NLP), *Part-Of-Speech* (POS) *tagger*, extração de frases substantivas, análise de sentimentos, classificação (utilizando algoritmos *Naive Bayes* e *Árvore de Decisão*), *tokenização*, tradução e correção ortográfica de textos.

O módulo *textblob.sentiments()* contém duas implementações de algoritmos de análise de sentimentos: i) *Pattern Analyzer* baseado na biblioteca *Patterns* [101] e ii) *Naive Bayes Analyzer* (um classificador NLTK [99] treinado em um *corpus* de resenhas de filmes [109]). A biblioteca *Textblob* utiliza como padrão o algoritmo *Pattern Analyzer* para calcular a polaridade e subjetividade dos textos (Figura 4.1).

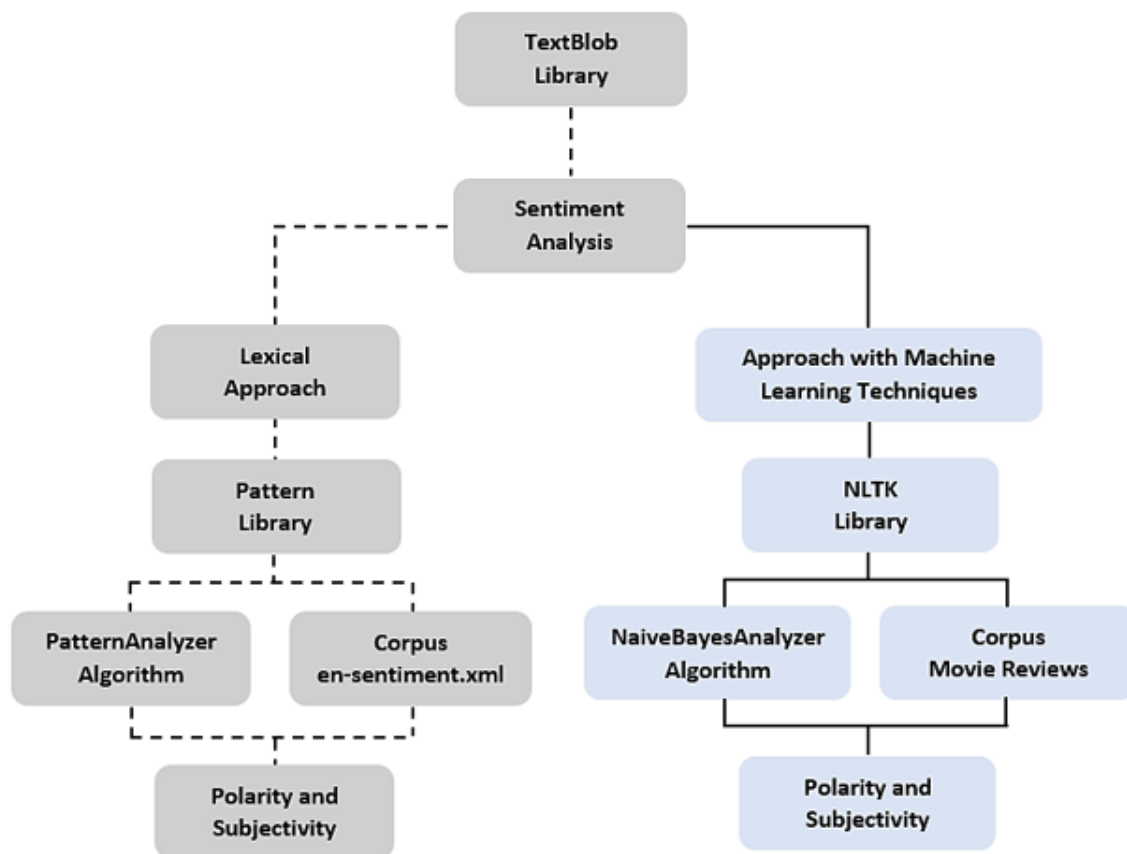


Figura 4.1 – Estrutura da biblioteca *TextBlob*.

4.1.1 Algoritmo Pattern Analyzer

O algoritmo *Pattern Analyzer* é baseado na biblioteca *Pattern*. Conforme [101], *Pattern* é uma biblioteca escrita em *Python*, dividida em vários módulos para mineração de dados (*Google*, *Twitter* e API do *Wikipedia*), processamento de linguagem natural (*POS taggers*, pesquisa por *n-grama*, análise de sentimentos e *WordNet*), aprendizado de máquina (*SVM*), análise de rede e visualização.

Os módulos *pattern.en* (inglês) e *pattern.nl* (holandês) contém um rápido *POS tagger* (capaz de identificar substantivos, adjetivos, verbos, etc., em uma frase), análise de sentimento, ferramentas para conjugação de verbos, singularização e pluralização de substantivos.

4.1.1.1 Conjunto de Dados Léxico

O arquivo *en-sentiment.xml* contém o *corpus* que é utilizado como léxico para atribuir pontuações de polaridade, subjetividade, intensidade e confiança. Cada palavra do conjunto de dados léxico tem uma identificação e pontuação. Esse *corpus* possui também um *Part-Of-Speech (POS) tagger* para determinar a classe gramatical (substantivos, verbos, adjetivos,

advérbios, conjunção, etc.) de cada palavra dentro de cada frase. As características do conjunto de dados léxico são as seguintes:

- Documento XML que inclui quatro entradas: polaridade, subjetividade, intensidade e confiança;
- Os adjetivos têm polaridade (negativa ou positiva, -1,0 a +1,0) e subjetividade (objetiva ou subjetiva, +0,0 a +1,0);
- A pontuação de cada palavra é definida de acordo com o sentido da frase, por exemplo: ridículo (lamentável) = negativa e ridículo (humorístico) = positiva;
- A identificação *cornetto_synset_id* refere-se ao banco de dados léxico *Cornetto* em holandês;
- A identificação *wordnet_id* refere-se ao banco de dados léxico *WordNet3* em inglês;
- Utiliza o conjunto de marcação de *Penn Treebank* [110] para determinar a classe gramatical (POS *tagger*) das palavras (Tabela 4.1).

Tabela 4.1 – Classe gramatical e POS *Tagger* padrão utilizado no conjunto de dados léxico.

Classe Gramatical	POS Tagger
Substantivo	"NN", "NNS", "NNP", "NNPS", "NP"
Verbo	"MD", "VB", "VBD", "VBG", "VBN", "VBP", "VBZ"
Adjetivo	"JJ", "JJR", "JJS"
Advérbio	"RB", "RBR", "RBS", "WRB"
Conjunção	"CC"
Preposição	"IN"
Interjeição	"UH"

A Tabela 4.2 apresenta três palavras do conjunto de dados léxico: a palavra ('great', 'JJ') é um adjetivo com polaridade positiva (1,0); a palavra ('very', 'RB') é um advérbio de intensidade com polaridade positiva (0,2); e a palavra ('bad', 'JJ') é um adjetivo com polaridade negativa (-0,7). É possível observar também que a palavra "very" pode ser utilizada para intensificar o valor da polaridade de outras palavras.

Tabela 4.2 – Palavras do conjunto de dados léxico.

Palavras	Conjunto de Dados Léxico
great	<word form="great" wordnet_id="a-01278818" pos="JJ" sense="of major significance or importance" polarity="1,0" subjectivity="1,0" intensity="1,0" confidence="0,9" />
very	<word form="very" wordnet_id="r-00031899" pos="RB" sense="used as intensifier" polarity="0,2" subjectivity="0,3" intensity="1,3" confidence="0,9" />
bad	<word form="bad" cornetto_synset_id="c_174" wordnet_id="a-01125429" pos="JJ" sense="having undesirable or negative qualities" polarity="-0,7" subjectivity="0,9" intensity="1,0" confidence="0,8" />

4.1.1.2 Palavras Desconhecidas

Palavras desconhecidas serão marcadas com POS *tagger* NN. As palavras desconhecidas que começarem com uma letra maiúscula serão marcadas com NNP. Na Tabela 4.3 é possível verificar que as palavras "World", "world", "orange", "apple", "ppxx" e "Ppxx" foram classificadas como desconhecidas (NN ou NNP) porque não estão no conjunto de dados léxico. Interessante notar também que o sinal de pontuação "\\\" em testes realizados neste trabalho foi classificado pelo algoritmo *Pattern Analyzer* com o mesmo POS *tagger* de uma palavra que começa com a letra maiúscula.

Tabela 4.3 – Relação de algumas palavras que não estão no conjunto de dados léxico.

Exemplo de Entrada dos Dados	World, world, orange, \\, apple, ppxx, Ppxx
Algoritmo Pattern Analyzer	<pre>words = TextBlob("World, world, orange, \\, apple, ppxx, Ppxx") print (words.tags)</pre>
Saída dos Dados	[('World', 'NNP'), ('world', 'NN'), ('orange', 'NN'), ('\\', 'NNP'), ('apple', 'NN'), ('ppxx', 'NN'), ('Ppxx', 'NNP')]

4.1.1.3 Cálculo da Polaridade e Subjetividade

A classe *_text.py* do algoritmo *Pattern Analyzer* é responsável pelo cálculo do sentimento. A propriedade *Sentiments* implementada no módulo *textblob.en.sentiments()* retorna uma *tuple* na forma de sentimento (polaridade, subjetividade). A pontuação da polaridade é um *float* dentro do intervalo $[-1,0), (1,0]$. A subjetividade é um *float* dentro do intervalo $[(0,0), (1,0)]$, onde 0,0 é muito objetivo e 1,0 é muito subjetivo. O algoritmo *Pattern Analyzer* inclui três classes de polaridade: positiva $[(0,01), (1,0)]$, negativa $[(-0,01), (-1,0)]$ e neutra (0,0).

Para calcular o valor da polaridade (\bar{P}) e da subjetividade (\bar{S}) de uma única palavra, o

algoritmo *Pattern Analyzer* aplica o conceito de média aritmética simples, onde a soma dos valores da polaridade ou subjetividade da palavra ($\sum_{i=1}^n X_i$) é dividida pela quantidade de vezes (n) que esta palavra aparece dentro do conjunto de dados léxico. As fórmulas para calcular a polaridade e a subjetividade de uma palavra do conjunto de dados léxico podem ser observadas nas Equações 4.1 e 4.2.

$$\bar{P}_{(x)} = \frac{1}{n} \sum_{i=1}^n X_i \quad (4.1)$$

$$\bar{S}_{(x)} = \frac{1}{n} \sum_{i=1}^n X_i \quad (4.2)$$

A Tabela 4.4 mostra que a palavra "great" aparece quatro vezes no conjunto de dados léxico com polaridades [(1,0), (1,0), (0,4) e (0,8)] e subjetividades [(1,0), (1,0), (0,2) e (0,8)] diferentes, mas com intensidades iguais a 1,0. Observa-se também que "great" possui o mesmo POS *tagger* (pos=JJ) nas quatro entradas.

Tabela 4.4 – Entradas da palavra "great" no conjunto de dados léxico.

Palavra	Conjunto de Dados Léxico
great	<word form="great" cornetto_synset_id="n_a-525317" wordnet_id="a-01123879" pos="JJ" sense="very good" polarity="1,0" subjectivity="1,0" intensity="1,0" confidence="0,9" />
	<word form="great" wordnet_id="a-01278818" pos="JJ" sense="of major significance or importance" polarity="1,0" subjectivity="1,0" intensity="1,0" confidence="0,9" />
	<word form="great" wordnet_id="a-01386883" pos="JJ" sense="relatively large in size or number or extent" polarity="0,4" subjectivity="0,2" intensity="1,0" confidence="0,9" />
	<word form="great" wordnet_id="a-01677433" pos="JJ" sense="remarkable or out of the ordinary in degree or magnitude or effect" polarity="0,8" subjectivity="0,8" intensity="1,0" confidence="0,9" />

Por exemplo, os valores da polaridade e subjetividade da palavra "great" são respectivamente 0,8 e 0,75 (Equações 4.3 e 4.4). O mesmo cálculo pode ser observado na Tabela 4.5 pelo algoritmo *Pattern Analyzer*.

$$\bar{P}_{(x)} = \frac{1,0 + 1,0 + 0,4 + 0,8}{4} = 0,8 \quad (4.3)$$

$$\bar{S}_{(x)} = \frac{1,0 + 1,0 + 0,2 + 0,2}{4} = 0,75 \quad (4.4)$$

Tabela 4.5 – Cálculo da polaridade e subjetividade da palavra "great" pelo algoritmo *Pattern Analyzer*.

Entrada dos Dados	<i>great</i>
Algoritmo Pattern Analyzer	<i>word = TextBlob("great")</i> <i>print (word.sentiment)</i>
Saída dos Dados	<i>Sentiment[(polarity=0,8), (subjectivity=0,75)]</i>

4.1.1.4 Polaridade e Subjetividade em Frases

Para calcular a polaridade (\overline{PF}) e a subjetividade (\overline{SF}) em frases, o algoritmo *Pattern Analyzer* faz a média aritmética simples com o conjunto de valores da polaridade (\overline{P}) ou subjetividade (\overline{S}) das palavras na frase. Neste caso, o valor final do sentimento da frase é calculado dividindo-se a soma dos valores das palavras que possuem sentimento na frase pela quantidade de tais palavras. As Equações 4.5 e 4.6 representam as duas fórmulas para calcular a polaridade e a subjetividade de uma frase que contém n palavras no conjunto de dados léxico. Palavras que não estão no conjunto de dados léxico, por exemplo, *stopwords* e caracteres especiais não entrarão no cálculo feito pelo algoritmo.

$$\overline{PF}_{(P)} = \frac{1}{n} \sum_{i=1}^n \overline{P}_i \quad (4.5)$$

$$\overline{SF}_{(S)} = \frac{1}{n} \sum_{i=1}^n \overline{S}_i \quad (4.6)$$

Por exemplo, na frase "*The world is beautiful and great*" existem palavras que não estão no conjunto de dados léxico. A palavra "*beautiful*" ($x1$) possui duas entradas e a palavra "*great*" ($x2$) quatro entradas (Tabela 4.6).

Tabela 4.6 – Polaridade e subjetividade das palavras da frase "*The world is beautiful and great*" no conjunto de dados léxico.

Palavras	POS Tagger	Polaridade	Subjetividade	Intensidade
<i>The</i>	DT	-	-	-
<i>world</i>	NN	-	-	-
<i>is</i>	VBZ	-	-	-
<i>beautiful</i>	JJ	[(0,7), (1,0)]	[(1,0), (1,0)]	1,0
<i>and</i>	CC	-	-	-
<i>great</i>	JJ	[(1,0), (1,0), (0,4), (0,8)]	[(1,0), (1,0), (0,2), (0,8)]	1,0

O cálculo da polaridade e da subjetividade das palavras "beautiful" (x_1) e "great" (x_2) podem ser observados nas Equações 4.7, 4.8, 4.9 e 4.10.

$$\bar{P}_{(x_1)} = \frac{0,7 + 1,0}{2} = 0,85 \quad (4.7)$$

$$\bar{P}_{(x_2)} = \frac{1,0 + 1,0 + 0,4 + 0,8}{4} = 0,8 \quad (4.8)$$

$$\bar{S}_{(x_1)} = \frac{1,0 + 1,0}{2} = 1,0 \quad (4.9)$$

$$\bar{S}_{(x_2)} = \frac{1,0 + 1,0 + 0,2 + 0,8}{4} = 0,75 \quad (4.10)$$

As Equações 4.11 e 4.12 mostram o cálculo da polaridade e da subjetividade da frase "The world is beautiful and great".

$$\overline{PF}_{(P)} = \frac{0,85 + 0,8}{2} = 0,825 \quad (4.11)$$

$$\overline{SF}_{(S)} = \frac{1,0 + 0,75}{2} = 0,875 \quad (4.12)$$

Na Tabela 4.7 é possível verificar também o mesmo cálculo feito pelo algoritmo *Pattern Analyzer*.

Tabela 4.7 – Cálculo da polaridade e subjetividade da frase "The world is beautiful and great" pelo algoritmo *Pattern Analyzer*.

Entrada dos Dados	<i>The world is beautiful and great</i>
Algoritmo Pattern Analyzer	<i>phrase = TextBlob("The world is beautiful and great") print (phrase.sentiment)</i>
Saída dos Dados	<i>Sentiment[(polarity=0,825), (subjectivity=0,875)]</i>

4.1.1.5 Polaridade e Subjetividade em Frases com Palavras de Negação

As palavras de negação "no", "not" e "never" invertem a polaridade da palavra seguinte. Para calcular a polaridade em uma frase negativa (*PFN*) o algoritmo *Pattern Analyzer* multiplica o valor (-0,5) da palavra de negação (n_1) pelo valor da polaridade (\bar{P}) da próxima palavra. As Equações 4.13 e 4.14 representam a fórmula e cálculo da polaridade da frase

"Not great". Observa-se que a polaridade da palavra "great" é invertida de 0,8 para -0,4.

$$PFN_{(n1,x1)} = n1 * \bar{P}_{(x1)} \quad (4.13)$$

$$PFN_{(n1,x1)} = (-0,5) * 0,8 = -0,4 \quad (4.14)$$

Na Tabela 4.8 é possível verificar a polaridade e a subjetividade da frase "Not great". Verifica-se que a palavra de negação "Not" não altera o valor da subjetividade da palavra "great".

Tabela 4.8 – Cálculo da polaridade e subjetividade da frase negativa "Not great" pelo algoritmo *Pattern Analyzer*.

Entrada dos Dados	<i>Not great</i>
Algoritmo Pattern Analyzer	<i>negative_phrase = TextBlob("Not great")</i> <i>print (negative_phrase.sentiment)</i>
Saída dos Dados	<i>Sentiment[(polarity=-0,4), (subjectivity=0,75)]</i>

A Tabela 4.9 mostra que o valor da polaridade em frases com duas ou mais palavras de negação, adicionadas antes de uma palavra com polaridade positiva ou negativa não altera.

Tabela 4.9 – Cálculo da polaridade e subjetividade da frase "Not not never no great" pelo algoritmo *Pattern Analyzer*.

Entrada dos Dados	<i>Not not never no great</i>
Algoritmo Pattern Analyzer	<i>negative_phrase = TextBlob("Not not never no great")</i> <i>print (negative_phrase.sentiment)</i>
Saída dos Dados	<i>Sentiment[(polarity=-0,4), (subjectivity=0,75)]</i>

4.1.1.6 Polaridade e Subjetividade em Frases com Advérbios

Palavras marcadas com POS *tagger* RB (advérbio) tem como finalidade alterar o valor da polaridade da próxima palavra com POS *tagger* JJ (adjetivo) ou NN (substantivo). Os advérbios são utilizados como modificadores.

Para calcular o valor da polaridade (*PFA*) e da subjetividade (*SFA*) de uma frase com advérbios, o algoritmo multiplica o valor da intensidade (*I*) da palavra marcada como advérbio (*aI*) pelo valor da polaridade (\bar{P}) ou da subjetividade (\bar{P}) da próxima palavra, seja ela,

um adjetivo ou um substantivo (Equações 4.15 e 4.16).

$$PFA_{(a1,x1)} = I_{(a1)} * \bar{P}_{(x1)} \quad (4.15)$$

$$SFA_{(a1,x1)} = I_{(a1)} * \bar{S}_{(x1)} \quad (4.16)$$

Por exemplo, "very" é uma palavra intensificadora com uma entrada no conjunto de dados léxico que possui POS *tagger* RB. A palavra "high" (JJ) possui cinco entradas. "Action" é um substantivo (NN) com duas entradas (Tabela 4.10).

Tabela 4.10 – Polaridade, subjetividade e intensidade das palavras "Very", "high" e "action" no conjunto de dados léxico.

Palavras	POS <i>Tagger</i>	Polaridade	Subjetividade	Intensidade
<i>Very</i>	RB	0,2	0,3	1,3
<i>high</i>	JJ	[(0,0), (0,2), (0,0), (0,3), (0,3)]	(0,3), (0,6), (0,4), (0,5), (0,9)]	1,0
<i>action</i>	NN	[(0,1), (0,1)]	[(0,1), (0,1)]	1,0

Na frase "Very high" a polaridade do adjetivo ('high', 'JJ') é alterada de 0,16 para 0,208 (Equação 4.17). O valor da subjetividade é maximizado para 0,702 (Equação 4.18). O valor da polaridade e da subjetividade do advérbio "very" não entra no cálculo para definir o valor da polaridade e subjetividade da frase.

$$PFA_{(a1,x1)} = 1,3 * 0,16 = 0,208 \quad (4.17)$$

$$SFA_{(a1,x1)} = 1,3 * 0,5399999999999999 = 0,702 \quad (4.18)$$

O mesmo processo é feito na frase "Very action", onde o valor da polaridade (Equação 4.19) e subjetividade (Equação 4.20) do substantivo "action" são maximizados. O cálculo para as duas frases pode ser observado nas Tabelas 4.11 e 4.12.

$$PFA_{(a1,x1)} = 1,3 * 0,1 = 0,13 \quad (4.19)$$

$$SFA_{(a1,x1)} = 1,3 * 0,1 = 0,13 \quad (4.20)$$

Tabela 4.11 – Cálculo da polaridade e subjetividade da frase "Very high" pelo algoritmo *Pattern Analyzer*.

Entrada dos Dados	<i>Very high</i>
Algoritmo Pattern Analyzer	<i>adverb_phrase = TextBlob("Very high")</i> <i>print (adverb_phrase.sentiment)</i>
Saída dos Dados	<i>Sentiment[(polarity=0,208), (subjectivity=0,702)]</i>

Tabela 4.12 – Cálculo da polaridade e subjetividade da frase "Very action" pelo algoritmo *Pattern Analyzer*.

Entrada dos Dados	<i>Very action</i>
Algoritmo Pattern Analyzer	<i>adverb_phrase = TextBlob("Very action")</i> <i>print (adverb_phrase.sentiment)</i>
Saída dos Dados	<i>Sentiment[(polarity=0,13), (subjectivity=0,13)]</i>

Outro exemplo pode ser observado na frase "The world is very beautiful and great", onde o algoritmo *Pattern Analyzer* calcula primeiro a polaridade (*PFA*) e a subjetividade (*SFA*) da junção (*bigrama*) das palavras adjacentes ('very', 'RB') e ('beautiful', 'JJ') - (advérbio e adjetivo). Neste exemplo, $I_{(a1)}$ representa o valor da intensidade da palavra "very". $\bar{P}_{(x1)}$ e $\bar{S}_{(x1)}$ representam os valores da polaridade e subjetividade da palavra "beautiful". $\bar{P}_{(x2)}$ e $\bar{S}_{(x2)}$ da palavra "great". Vale ressaltar, que conforme a Seção 4.1.1.4, as palavras "The", "world", "is" e "and" não estão no conjunto de dados léxico, portanto não possuem valor. A fórmula para calcular a polaridade e a subjetividade desta frase pode ser observada nas Equações 4.21 e 4.22.

$$\begin{aligned} \overline{PF}_{(a1,x1,x2)} &= \frac{PFA_{(a1,x1)} + \bar{P}_{(x2)}}{2} \\ &= \frac{(I_{(a1)} * \bar{P}_{(x1)}) + \bar{P}_{(x2)}}{2} \end{aligned} \quad (4.21)$$

$$\begin{aligned} \overline{SF}_{(a1,x1,x2)} &= \frac{SFA_{(a1,x1)} + \bar{S}_{(x2)}}{2} \\ &= \frac{(I_{(a1)} * \bar{S}_{(x1)}) + \bar{S}_{(x2)}}{2} \end{aligned} \quad (4.22)$$

As Equações 4.23 e 4.24 e a Tabela 4.13 (algoritmo *Pattern Analyzer*) representam os

valores da polaridade (0,9) e da subjetividade (0,875).

$$\overline{PF}_{(PFA,\overline{P})} = \frac{1,0 + 0,8}{2} = 0,9 \quad (4.23)$$

$$\overline{SF}_{(SFA,\overline{P})} = \frac{1,0 + 0,75}{2} = 0,875 \quad (4.24)$$

Tabela 4.13 – Cálculo da polaridade e subjetividade da frase "The world is very beautiful and great" pelo algoritmo *Pattern Analyzer*.

Entrada dos Dados	<i>The world is very beautiful and great</i>
Algoritmo Pattern Analyzer	<i>phrase = TextBlob("The world is very beautiful and great") print (phrase.sentiment)</i>
Saída dos Dados	<i>Sentiment[(polarity=0,9), (subjectivity=0,875)]</i>

4.1.1.7 Polaridade e Subjetividade em Frases com Palavras de Negação e Advérbios

Palavras de negação trabalham em conjunto com advérbios. Neste caso, para calcular a polaridade (*PFNA*) o algoritmo multiplica o valor (-0,5) da palavra de negação (*n1*) pelo inverso da intensidade (*I*) do advérbio (*a1*) e pela polaridade da próxima palavra (*x1*). A subjetividade (*SFNA*) é definida através da multiplicação do inverso da intensidade (*I*) do advérbio (*a1*) pela subjetividade da próxima palavra (*x1*). As Equações 4.25 e 4.26 representam as fórmulas para calcular a polaridade e a subjetividade.

$$PFNA_{(n1,a1,x1)} = n1 * \left(\frac{1}{I_{(a1)}} \right) * \overline{P}_{(x1)} \quad (4.25)$$

$$SFNA_{(a1,x1)} = \left(\frac{1}{I_{(a1)}} \right) * \overline{S}_{(x1)} \quad (4.26)$$

Por exemplo, a frase "Not very great" é composta por uma palavra de negação, um advérbio e um adjetivo. Para calcular a polaridade dessa frase, o valor (-0,5) da palavra de negação "not" é multiplicado pelo inverso da intensidade (1/1,3) do advérbio ('very', 'RB') e pela polaridade (0,8) da palavra ('great', 'JJ') (Equação 4.27). O cálculo da subjetividade pode ser observado na Equação 4.28.

$$PFNA_{(n1,a1,x1)} = (-0,5) * \left(\frac{1}{1,3} \right) * 0,8 = -0,307 \quad (4.27)$$

para a subjetividade (Tabela 4.16).

$$\overline{PF}_{(P)} = \frac{1}{n} \sum_{i=1}^n \overline{P}_i = \frac{1,0}{1} = 1,0 \quad (4.29)$$

Tabela 4.16 – Cálculo da polaridade e subjetividade da frase "<3 the world" pelo algoritmo *Pattern Analyzer*.

Entrada dos Dados	<3 the world
Algoritmo Pattern Analyzer	<i>emoticons_phrase = TextBlob("<3 the world")</i> <i>print (emoticons_phrase.sentiment)</i>
Saída dos Dados	<i>Sentiment[(polarity=1,0), (subjectivity=1,0)]</i>

Entretanto, a palavra "love" possui três entradas no conjunto de dados léxico classificadas com POS *tagger* VB, valores de polaridade (0,5) e subjetividade (0,6) iguais (Tabela 4.17). Assim, a palavra "love" aplicada no lugar do *emoticon* "<3" mantém o sentimento da frase, mas altera o valor da polaridade de 1,0 para 0,5 (Equação 4.30 e Tabela 4.18).

Tabela 4.17 – Entradas da palavra "love" no conjunto de dados léxico.

Palavra	Conjunto de Dados Léxico
love	<word form="love" wordnet_id="v-1775164" pos="VB" sense="have a great affection or liking for" polarity="0,5" subjectivity="0,6" intensity="1,0" confidence="0,9" />
	<word form="love" wordnet_id="v-1775535" pos="VB" sense="be enamored or in love with" polarity="0,5" subjectivity="0,6" intensity="1,0" confidence="0,9" />
	<word form="love" wordnet_id="v-1775535" pos="VB" sense="be enamored or in love with" polarity="0,5" subjectivity="0,6" intensity="1,0" confidence="0,9" />

$$\overline{PF}_{(P)} = \frac{1}{n} \sum_{i=1}^n \overline{P}_i = \frac{0,5}{1} = 0,5 \quad (4.30)$$

Tabela 4.18 – Cálculo da polaridade e subjetividade da frase "love the world" pelo algoritmo *Pattern Analyzer*.

Entrada dos Dados	<i>love the world</i>
Algoritmo Pattern Analyzer	<i>emoticons_phrase = TextBlob("love the world") print (emoticons_phrase.sentiment)</i>
Saída dos Dados	<i>Sentiment[(polarity=0,5), (subjectivity=0,6)]</i>

4.2 RESUMO DO CAPÍTULO 4

Este capítulo mostrou, através de engenharia reversa, como o algoritmo *Pattern Analyzer* utiliza o conjunto de dados léxico para calcular os valores da polaridade e da subjetividade em palavras, frases, frases com palavras de negação, frases com advérbios, frases com palavras de negação e advérbios e frases com *emoticons*. Essas informações serão utilizadas para identificar vulnerabilidades e gerar as estratégias de ataques adversariais no classificador de sentimento léxico *Pattern Analyzer* da biblioteca *TextBlob* propostas no Capítulo 5.

5

ATAQUES ADVERSARIAIS EM ANÁLISE DE SENTIMENTO

Atualmente técnicas de análise de sentimento estão sendo utilizadas em um amplo espectro de aplicações. No entanto, até o momento, algumas dessas técnicas desenvolvidas são vulneráveis a entrada de dados manipulados, chamados de exemplos adversariais [89], onde pequenas perturbações podem ser utilizadas para enganar algoritmos na classificação de sentimentos [90]. A quantidade de estudos especializados sobre estratégias desses ataques adversariais em aplicações que utilizam técnicas de análise de sentimento permanece relativamente pequena. Entretanto, é um campo que precisa de atenção especial devido às recentes pesquisas voltadas para esse tema [83, 85, 86, 87, 82, 88, 89, 90, 91].

Esta seção propõe duas estratégias de ataques adversariais *white-box* no classificador léxico de linguagem natural (*Pattern Analyzer*) da biblioteca *TextBlob*, utilizado em várias aplicações para calcular o sentimento dos usuários em aplicações de mídia social.

Através dos experimentos foi possível mostrar como um adversário minimamente capaz pode enganar a aplicação que está utilizando o algoritmo *Pattern Analyzer* da biblioteca *TextBlob* inserindo caracteres nas palavras (ataques de inserção), substituindo palavras negativas ou positivas de uma frase por sinônimos (ataques de substituição) que não estão no conjunto de dados léxico ou substituindo palavras de negação por outras palavras semelhantes que não fazem parte do código do algoritmo *Pattern Analyzer*. Além disso, apresentamos também para essas vulnerabilidades algumas contramedidas que podem ser utilizadas para mitigar esses ataques.

5.1 ESTRATÉGIA DE ATAQUE I: INSERÇÃO DE CARACTERES

Este modelo de ataque tem como finalidade fazer pequenas modificações através da inserção de caracteres em palavras de frases em inglês ou em outros idiomas, com o objetivo de gerar ruído com erros de ortografia. Essa modificação, segundo o funcionamento do algoritmo detalhado no Capítulo 4, seria capaz de alterar o resultado da saída do classificador de sentimento *Pattern Analyzer*.

Como exemplo, utilizamos duas amostras de textos originais: uma em inglês ("*This house is very nice and elegant*") e outra em português ("*Esta casa é muito agradável e elegante*"). Além disso, na sequência, apresentamos também duas contramedidas que podem ser utilizadas para mitigar esses ataques.

5.1.1 Inserção de Caracteres em Frases com Idioma Inglês

Neste ataque utilizamos a amostra de texto original em inglês "This house is very nice and elegant". Primeiro aplicamos o método *tags()* da biblioteca *TextBlob* para identificar o POS *tagger* de cada palavra na frase (Tabela 5.1). Verifica-se que a palavra ('very', 'RB') foi classificada pelo algoritmo como um advérbio.

Tabela 5.1 – POS *Tagger* de cada palavra da frase "This house is very nice and elegant".

Entrada - Amostra Original	<i>This house is very nice and elegant</i>
Algoritmo Pattern Analyzer	<i>phrase = TextBlob("This house is very nice and elegant")</i> <i>print (phrase.tags)</i>
Saída dos Dados	<i>[('This', 'DT'), ('house', 'NN'), ('is', 'VBZ'), ('very', 'RB'), ('nice', 'JJ'), ('and', 'CC'), ('elegant', 'JJ')]</i>

Depois identificamos no *corpus* léxico os valores da polaridade ($\bar{P}_{(x)}$), subjetividade ($\bar{S}_{(x)}$) e intensidade de cada palavra da amostra do texto original (Tabela 5.2). É possível verificar também que as palavras ('This', 'DT'), ('house', 'NN'), ('is', 'VBZ') e ('and', 'CC') não possuem valor de polaridade, subjetividade e intensidade. Isso significa que essas palavras não estão inseridas no conjunto de dados léxico.

Tabela 5.2 – Polaridade, subjetividade e intensidade das palavras da frase "This house is very nice and elegant" no conjunto de dados léxico.

Palavras	Polaridade	Subjetividade	Intensidade
<i>This</i>	-	-	-
<i>house</i>	-	-	-
<i>is</i>	-	-	-
<i>very</i>	0,2	0,3	1,3
<i>nice</i>	$(0,6 + 0,6)/2 = 0,6$	$(1,0 + 1,0)/2 = 1,0$	1,0
<i>and</i>	-	-	-
<i>elegant</i>	$(0,5 + 0,5 + 0,5)/3 = 0,5$	$(1,0 + 1,0 + 1,0)/3 = 1,0$	1,0

Para calcular os valores da polaridade (0,64) e da subjetividade (1,0) em frases com advérbios aplicamos as Equações 5.1 e 5.2. O cálculo do texto original pelo algoritmo *Pattern Analyzer* pode ser observado também na Tabela 5.3.

$$\begin{aligned}
 \overline{PF}_{(a1,x1,x2)} &= \frac{(I_{(a1)} * \bar{P}_{(x1)}) + \bar{P}_{(x2)}}{2} \\
 &= \frac{(1,3 * 0,6) + 0,5}{2} = 0,64
 \end{aligned}
 \tag{5.1}$$

$$\begin{aligned} \overline{SF}_{(a1,x1,x2)} &= \frac{(I_{(a1)} * \overline{S}_{(x1)}) + \overline{S}_{(x2)}}{2} \\ &= \frac{(1,3 * 1,0) + 1,0}{2} = 1,0 \end{aligned} \quad (5.2)$$

Tabela 5.3 – Cálculo da polaridade e subjetividade da frase "This house is very nice and elegant" pelo algoritmo *Pattern Analyzer*.

Entrada - Amostra Original	<i>This house is very nice and elegant</i>
Algoritmo Pattern Analyzer	<i>phrase = TextBlob("This house is very nice and elegant") print (phrase.sentiment)</i>
Saída dos Dados	<i>Sentiment[(polarity=0,64), (subjectivity=1,0)]</i>

Na perspectiva do atacante é importante ter o conhecimento que o algoritmo analisa apenas adjetivos, advérbios, alguns substantivos e algumas palavras de negação (Tabelas 5.1 e 5.2). Assim, criamos uma amostra de texto adversarial e inserimos pequenas perturbações com os caracteres "e", "c" e "g" nas palavras ('very', 'RB'), ('nice', 'JJ') e ('elegant', 'JJ').

A Figura 5.1 apresenta o ataque com as amostras de texto original ("This house is very nice and elegant") e texto adversarial ("This house is *ve*ry *ni*ice and *eleg*gant"). Nota-se que os valores da polaridade e subjetividade mudam drasticamente de [(p=0,64), (s=1,0)] para [(p=0,0), (s=0,0)]. Neste caso, as palavras "ve~~r~~y", "ni~~c~~e" e "eleg~~g~~ant" não são mais conhecidas pelo algoritmo *Pattern Analyzer*, pois, conforme explicitado anteriormente, elas deixam de fazer parte do conjunto de dados léxico. Assim, após o ataque de inserção de caracteres, o algoritmo classifica a amostra de texto adversarial como neutra.

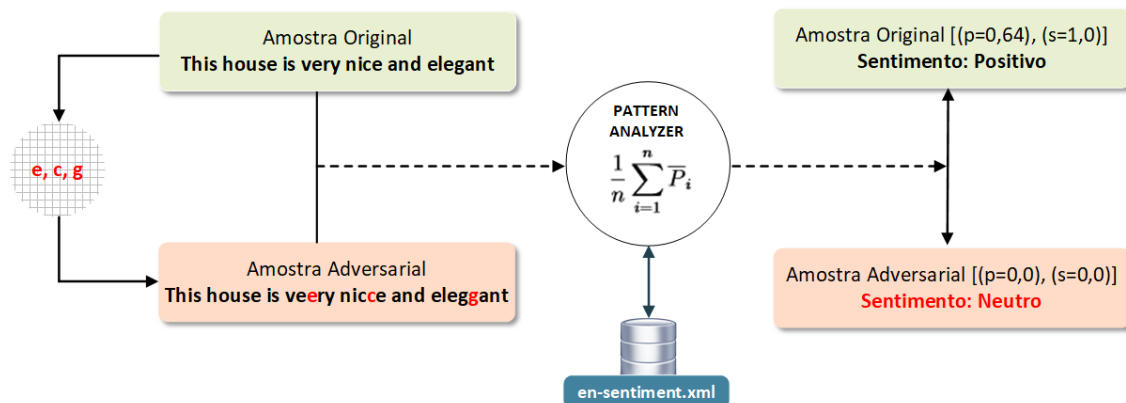


Figura 5.1 – Ataque adversarial através de ruído com erros de ortografia no algoritmo *Pattern Analyzer*.

O ataque pode ser observado também na Tabela 5.4, onde a mesma amostra de texto

adversarial ("This house is *veery niice and eleggant*") é processada em um *script* pelo algoritmo *Pattern Analyzer* sem nenhum método de correção. Nota-se que o algoritmo classifica o texto adversarial como neutro [(*polarity*=0,0), (*subjectivity*=0,0)]. A representação visual desse ataque pode ser observada na Figura 5.2.

Tabela 5.4 – Ataque adversarial através de ruído com erros de ortografia na frase "This house is *veery niice and eleggant*".

Entrada - Amostra Adversarial	<i>This house is veery niice and eleggant</i>
Algoritmo Pattern Analyzer	<pre> phrase = TextBlob("This house is veery niice and eleggant") print (phrase) if phrase.sentiment.polarity > 0: print (phrase.sentiment) print ("Polarity: Positive") elif phrase.sentiment.polarity == 0: print (phrase.sentiment) print ("Polarity: Neutral") else: print (phrase.sentiment) print ("Polarity: Negative") </pre>
Saída dos Dados	<p><i>This house is veery niice and eleggant</i> <i>Sentiment[(polarity=0,0), (subjectivity=0,0)]</i> <i>Polarity: Neutral</i></p>

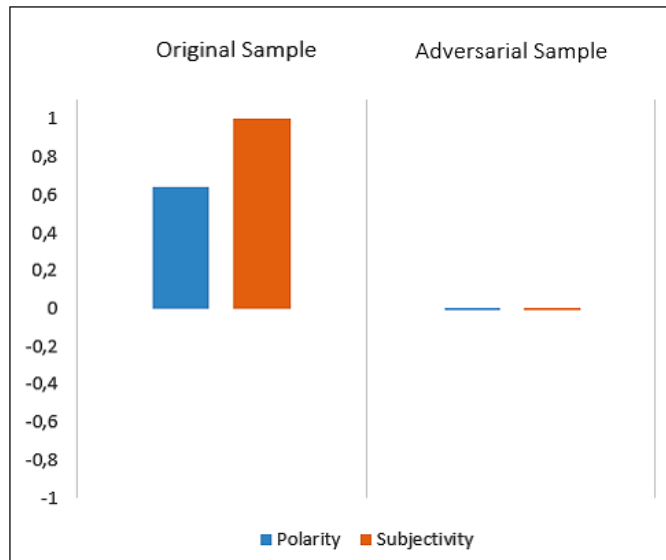


Figura 5.2 – Representação visual do ataque de inserção em frases com o idioma inglês.

5.1.1.1 Contramedida para os Ataques de Inserção em Frases com Idioma Inglês

Como contramedida para este ataque, deve-se aplicar o conceito de correção ortográfica nas frases. O algoritmo *Pattern Analyzer* utiliza o método *correct()* da biblioteca *TextBlob* para corrigir as palavras do idioma inglês [10]. Essa contramedida leva em consideração que todo esse processo de correção deverá ser feito antes de qualquer pré-processamento, evitando que o atacante, por conhecer a forma de processamento do algoritmo, seja capaz de desenvolver ataques dessa natureza.

No *script* da Tabela 5.5 é possível verificar a aplicação do método de correção para a mesma amostra de texto adversarial. Verifica-se que após a correção através do método *correct()* os valores da polaridade e da subjetividade são alterados pelo algoritmo para [(*polarity*=0,64), (*subjectivity*=1,0)]. A palavra ('*This*', '*DT*') também é alterada para ('*His*', '*PRP\$*'). A amostra de texto adversarial foi corrigida e classificada pelo algoritmo como positiva.

Tabela 5.5 – Correção do ataque adversarial através do método *correct()* da biblioteca *TextBlob* na frase "*This house is veery niice and eleggant*".

Entrada - Amostra Adversarial	<i>This house is veery niice and eleggant</i>
Algoritmo Pattern Analyzer	<pre>phrase = TextBlob("This house is veery niice and eleggant") p = phrase.correct() print (p) if p.sentiment.polarity > 0: print (p.sentiment) print ("Polarity: Positive") elif f.sentiment.polarity == 0: print (p.sentiment) print ("Polarity: Neutral") else: print (p.sentiment) print ("Polarity: Negative")</pre>
Saída dos Dados	<pre>His house is very nice and elegant Sentiment[(polarity=0,64), (subjectivity=1,0)] Polarity: Positive</pre>

A Figura 5.3 representa a correção do ataque. Nela é possível observar que para neutralizar o ataque todas as frases deverão ser pré-processadas primeiro pelo método *correct()* antes de serem enviadas para o classificador de sentimento. Vale ressaltar que a simplicidade da contramedida não gera elevado impacto nem de processamento e nem de complexidade de implementação. Entretanto, conforme foi observado nos estudos do estado da arte [44, 53, 54, 55, 56, 59, 60], essa contramedida não é utilizada, facilitando a possibilidade de ataques bem sucedidos nos trabalhos relacionados.

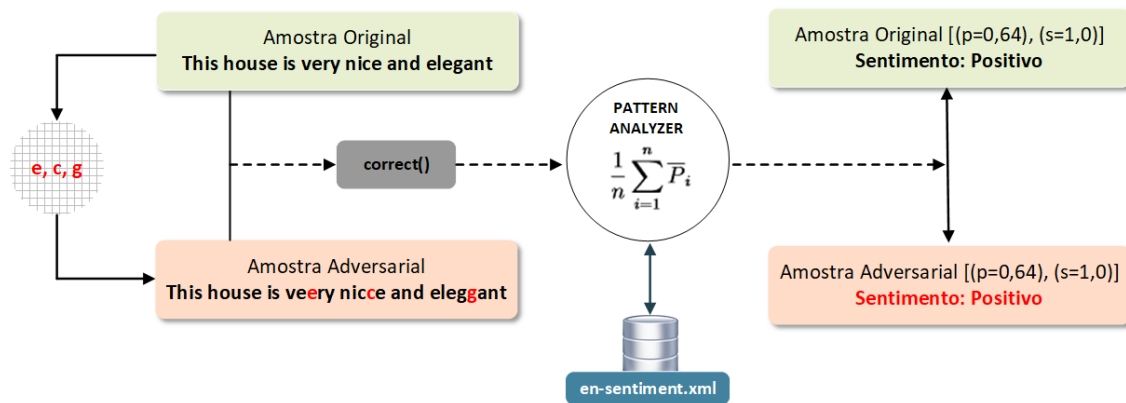


Figura 5.3 – Correção do ataque adversarial através do método *correct()*.

5.1.2 Inserção de Caracteres em Frases com Outros Idiomas

Este ataque pode ser aplicado em frases com outros idiomas. Conforme [10], o algoritmo *Pattern Analyzer* utiliza o parâmetro *wordnet_id* para identificar as palavras em um *corpus* léxico inglês. Assim, as frases de outros idiomas deverão ser traduzidas através dos métodos *get_languages()*, *detect_language()* e *translate()* para o idioma inglês antes de serem analisadas pelo classificador de sentimento [100].

Para mostrar o processo do ataque de inserção em frases com outros idiomas inserimos os caracteres ":", "/", e "?" na amostra de texto original em português ("*Esta casa é muito agradável e elegante*") para criar a amostra de texto adversarial ("*Esta casa é mu:into agr/adável e eleg?ante*"). A Figura 5.4 representa o ataque com essas amostras (texto original e texto adversarial em português).

É possível observar que antes do processo de classificação existem duas funções. A função *translate()* para traduzir os textos para o idioma inglês e a função *correct()* para corrigir as palavras em inglês. Mais uma vez observa-se que após o ataque a frase é classificada pelo algoritmo como neutra.

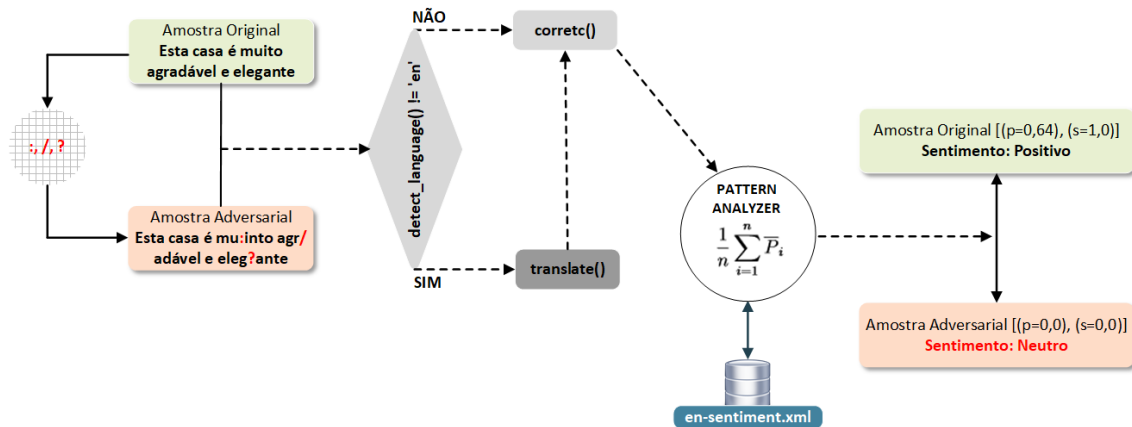


Figura 5.4 – Ataque adversarial através de ruído com erros de ortografia em frases com outros idiomas no algoritmo *Pattern Analyzer*.

A Tabela 5.6 apresenta um *script* com os dois métodos (*translate()* e *correct()*) sendo utilizados durante o pré-processamento do texto. Verifica-se que as palavras de peso ("mu:into", "agr/adável" e "eleg?ante") da amostra adversarial não foram corrigidas pelos métodos de tradução e correção ("my: into, air / adve, leg? Ante") [100]. Aplicando a função *tags()* é possível observar que grande parte das palavras desconhecidas foram classificadas pelo algoritmo como substantivos - POS *tagger* NN (Tabela 5.7). Neste caso, como o algoritmo não conhece essas palavras, ele atribui os valores $[(polarity=0,0), (subjectivity=0,0)]$ para a amostra adversarial. A representação visual desse ataque pode ser observada na Figura 5.5.

Tabela 5.6 – Ataque adversarial através de ruído com erros de ortografia na frase "Esta casa é mu:into agr/adável e eleg?ante".

Entrada - Amostra Adversarial	<i>Esta casa é mu:into agr/adável e eleg?ante</i>
Algoritmo Pattern Analyzer	<pre> phrase = TextBlob("Esta casa é mu:into agr/adável e eleg?ante") if phrase.detect_language() != 'en': translate_to_english = TextBlob(str(phrase.translate(to='en'))) t = translate_to_english.correct() print (t) sentiment.polarity(t) else: t = phrase.correct() print (t) sentiment.polarity(t) </pre>
Saída dos Dados	<pre> His house is my: into air / adve e leg? Ante Sentiment[(polarity=0,0), (subjectivity=0,0)] Polarity: Neutral </pre>

Tabela 5.7 – POS *Tagger* de cada palavra da frase "His house is my: into air / adve e leg? Ante".

Entrada dos Dados	<i>His house is my: into air / adve e leg? Ante</i>
Algoritmo Pattern Analyzer	<i>phrase = TextBlob("His house is my: into air / adve e leg? Ante") print (phrase.tags)</i>
Saída dos Dados	<i>[('His', 'PRP\$'), ('house', 'NN'), ('is', 'VBZ'), ('my', 'PRP\$'), ('into', 'IN'), ('air', 'NN'), ('/', 'JJ'), ('adve', 'NN'), ('e', 'NN'), ('leg', 'NN'), ('Ante', 'NN')]</i>

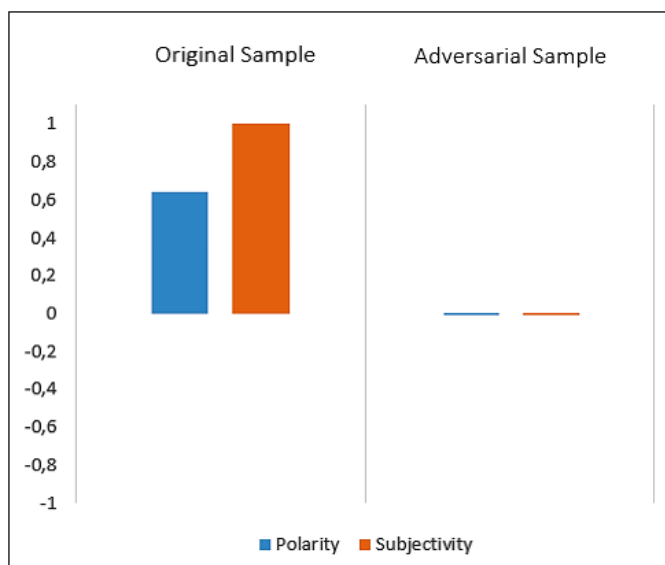


Figura 5.5 – Representação visual do ataque de inserção em frases com outros idiomas.

5.1.2.1 Contramedida para os Ataques de Inserção em Frases com Outros Idiomas

Para mitigar esse ataque deve-se aplicar antes do método de tradução uma função para corrigir as palavras do idioma que está sendo utilizada na aplicação. Por exemplo, para o texto adversarial escrito em português ("Esta casa é mu:into agr/adável e eleg?ante") deve-se criar uma função com um dicionário escrito em português para corrigir as palavras dessa frase. Neste caso, a aplicação terá uma função de tradução (*translate()*) e duas funções de correção (*correct_language_portuguese()* e *correct()*).

A Figura 5.6 e o *script* da Tabela 5.8 mostram a aplicação dos três métodos para mitigar o ataque e corrigir os valores da polaridade (0,64) e da subjetividade (1,0) da amostra de texto adversarial escrita com o idioma português. Uma função (*correct_language_portuguese()*) é aplicada antes e outra (*correct()*) após o método de tradução (*translate()*). Assim, antes da tradução, as palavras da amostra adversarial escrita em português serão primeiro corrigidas para depois serem traduzidas para o idioma inglês. Após esta fase, a amostra passa ainda por

outra função de correção para garantir que as palavras com o idioma inglês estejam corretas antes de serem enviadas para o analisador de sentimento.

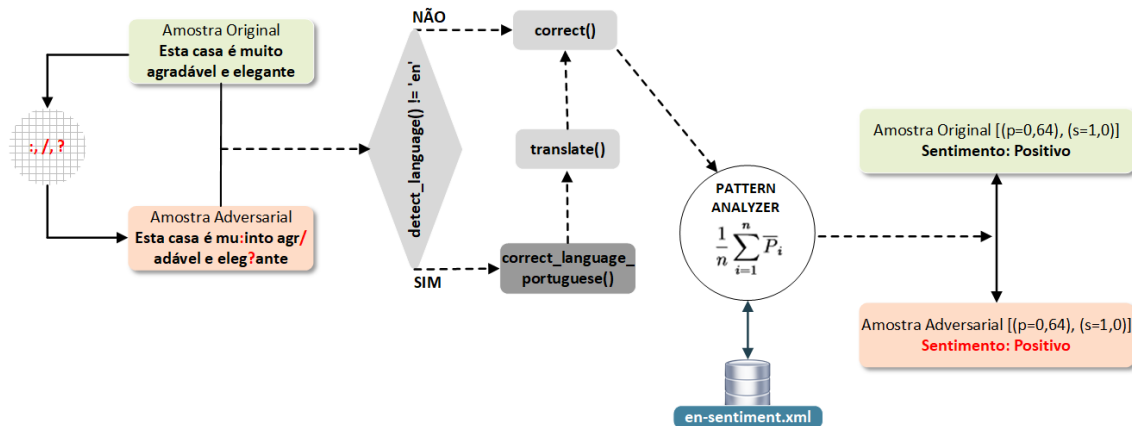


Figura 5.6 – Correção do ataque adversarial através dos métodos *correct_language_portuguese()*, *translate()* e *correct()*.

Tabela 5.8 – Correção do ataque adversarial através dos métodos *correct_language_portuguese()*, *translate()* e *correct()* na frase "Esta casa é mu:into agr/adável e eleg?ante".

Entrada - Amostra Adversarial	<i>Esta casa é mu:into agr/adável e eleg?ante</i>
Algoritmo Pattern Analyzer	<pre> phrase = TextBlob("Esta casa é mu:into agr/adável e eleg?ante") if phrase.detect_language() != 'en': p = phrase.correct_language_portuguese() print (p) translate_to_english = TextBlob(str(p.translate(to='en'))) t = translate_to_english.correct() print (t) sentiment.polarity(t) else: t = phrase.correct() print (t) sentiment.polarity(t) </pre>
Saída dos Dados	<pre> Esta casa é muito agradável e elegante His house is very nice and elegant Sentiment[(polarity=0,64), (subjectivity=1,0)] Polarity: Positive </pre>

Percebe-se que essa contramedida, além de ser mais complexa do que a contramedida da Seção 5.1.1.1, permite evitar que palavras escritas em outros idiomas com erros de ortografia (ataques de inserção) possam ser utilizadas por usuários mal-intencionados na tentativa de alterar o resultado do classificador de sentimento. Além disso, em termos de pré-processamento, ela pode melhorar também a precisão do algoritmo *Pattern Analyzer* durante o processo de classificação dos textos.

5.2 ESTRATÉGIA DE ATAQUE II: SUBSTITUIÇÃO DE PALAVRAS

Este modelo de ataque substitui palavras de uma frase por sinônimos. Tem como finalidade identificar na frase os adjetivos, advérbios e substantivos para alterá-los por sinônimos que não estão no conjunto de dados léxico. As palavras de negação também podem ser alteradas por sinônimos que não são utilizados pelo algoritmo *Pattern Analyzer*. Com esse ataque, as frases com sentimento positivo ou negativo terão o mesmo sentido para os humanos, mas serão sempre classificadas pelo algoritmo *Pattern Analyzer* de forma incorreta, sendo considerada, neste caso, um falso positivo.

Para aplicar este ataque utilizamos três amostras de textos originais: "*The girl is evil*", "*She is charming*" e "*It's not pretty what you are doing Alice*". Apresentamos também as contramedidas para mitigar esses ataques.

5.2.1 Substituição em Frases Negativas ou Positivas

Neste ataque utilizamos como alvo a amostra de texto original "*The girl is evil*". Através do POS *Tagger* identificamos que a palavra "*evil*" foi classificada pelo algoritmo como um adjetivo (Tabela 5.9).

Tabela 5.9 – POS *Tagger* de cada palavra da frase "*The girl is evil*".

Entrada - Amostra Original	<i>The girl is evil</i>
Algoritmo Pattern Analyzer	<code>phrase = TextBlob("The girl is evil") print (phrase.tags)</code>
Saída dos Dados	<code>[('The', 'DT'), ('girl', 'NN'), ('is', 'VBZ'), ('evil', 'JJ')]</code>

No *corpus* léxico foi possível verificar também que apenas a palavra "*evil*" ('JJ') possui valor em sua polaridade, subjetividade e intensidade (Tabelas 5.10 e 5.11). Portanto, essa palavra foi escolhida como objeto para realizar o ataque.

Tabela 5.10 – Entradas da palavra "evil" no conjunto de dados léxico.

Palavra	Conjunto de Dados Léxico
evil	<code><word form="evil" cornetto_synset_id="n_a-516402" wordnet_id="a-00224515" pos="JJ" sense="having or exerting a malignant influence" polarity="-1,0" subjectivity="1,0" intensity="1,0" confidence="0,9" /></code>
	<code><word form="evil" wordnet_id="a-01131043" pos="JJ" sense="morally bad or wrong" polarity="-1,0" subjectivity="1,0" intensity="1,0" confidence="0,9" /></code>
	<code><word form="evil" wordnet_id="a-02514099" pos="JJ" sense="having the nature of vice" polarity="-1,0" subjectivity="1,0" intensity="1,0" confidence="0,9" /></code>

Tabela 5.11 – Polaridade, subjetividade e intensidade das palavras da frase "The girl is evil" no conjunto de dados léxico.

Palavras	Polaridade	Subjetividade	Intensidade
The	-	-	-
girl	-	-	-
is	-	-	-
evil	$[-1,0 + (-1,0) + (-1,0)]/3 = -1,0$	$(1,0 + 1,0 + 1,0)/3 = 1,0$	1,0

O cálculo da polaridade ($\overline{PF}_{(P)}$) e da subjetividade ($\overline{SF}_{(S)}$) em frases sem advérbios e palavras de negação é feito através das Equações 5.3 e 5.4. A Tabela 5.12 mostra o mesmo cálculo feito pelo algoritmo *Pattern Analyzer*. Percebe-se que esta frase possui sentimento negativo [(polarity=-1,0), (subjectivity=1,0)].

$$\overline{PF}_{(P)} = \frac{1}{n} \sum_{i=1}^n \overline{P}_i = \frac{-1,0}{1} = -1,0 \quad (5.3)$$

$$\overline{SF}_{(S)} = \frac{1}{n} \sum_{i=1}^n \overline{S}_i = \frac{1,0}{1} = 1,0 \quad (5.4)$$

Tabela 5.12 – Cálculo da polaridade e subjetividade da frase "The girl is evil" pelo algoritmo *Pattern Analyzer*.

Entrada - Amostra Original	<i>The girl is evil</i>
Algoritmo Pattern Analyzer	<code>phrase = TextBlob("The girl is evil") print (phrase.sentiment)</code>
Saída dos Dados	<code>Sentiment[(polarity=-1,0), (subjectivity=1,0)]</code>

Neste caso, tendo o conhecimento de que apenas a palavra "evil" tem valor na frase, criamos um texto adversarial ("The girl is *wicked*") substituindo a palavra "evil" por um sinônimo ("*wicked*") que não faz parte do conjunto de dados léxico. A Figura 5.7 mostra a aplicação do ataque com as amostras de textos original e adversarial. Observa-se que mesmo com o método *correct()*, os valores da polaridade alteram de [(p=-1,0), (s=1,0)] para [(p=0,0), (s=0,0)]. Assim, o algoritmo classifica a amostra de texto adversarial como neutra.

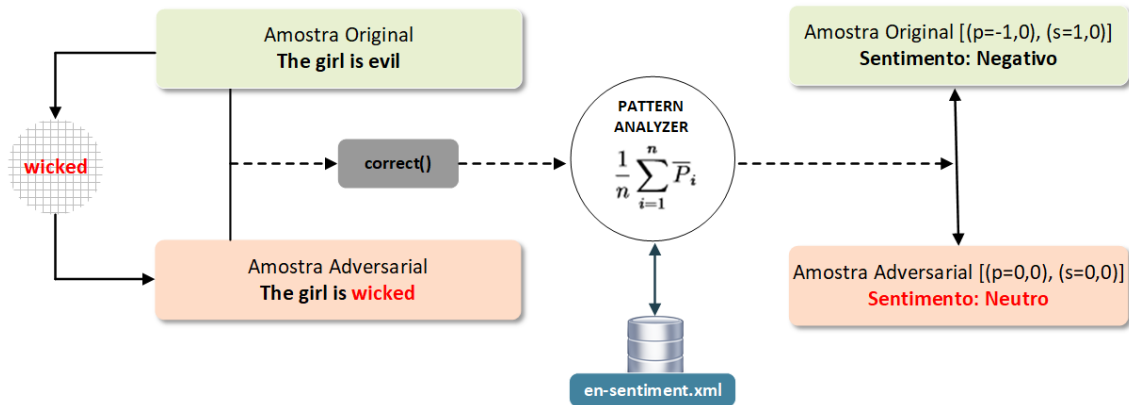


Figura 5.7 – Ataque adversarial através da substituição de palavras por seus sinônimos.

No *script* da Tabela 5.13 é possível verificar a efetividade do ataque com a mesma amostra de texto adversarial ("The girl is *wicked*"). Percebe-se novamente que o algoritmo classifica o texto como neutro [(polarity=0,0), (subjectivity=0,0)]. A representação visual desse ataque pode ser observada na Figura 5.8.

Tabela 5.13 – Ataque adversarial através da substituição de palavras na frase "The girl is *wicked*".

Entrada - Amostra Adversarial	<i>The girl is wicked</i>
Algoritmo Pattern Analyzer	<pre> phrase = TextBlob("The girl is wicked") p = phrase.correct() print (p) print (p.sentiment) </pre>
Saída dos Dados	<i>The girl is wicked</i> Sentiment[(polarity=0,0), (subjectivity=0,0)]

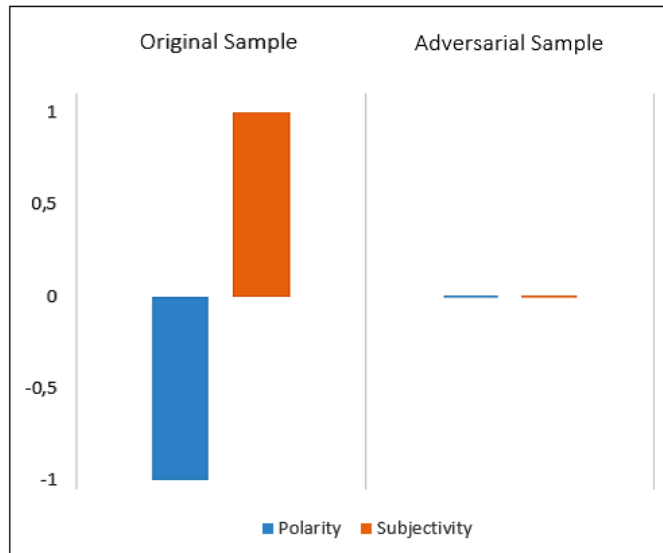


Figura 5.8 – Representação visual do ataque de substituição em frases negativas ou positivas.

Este mesmo modelo de ataque pode ser aplicado também através do texto adversarial "*She is **seductive***". Observa-se que o texto adversarial gerado a partir do texto original (*She is charming*) tem o mesmo sentimento para os humanos (Tabela 5.14). Entretanto, o algoritmo classifica a amostra de texto adversarial como neutra (Tabela 5.15).

Tabela 5.14 – Cálculo da polaridade e subjetividade da frase "*She is charming*" pelo algoritmo *Pattern Analyzer*.

Entrada - Amostra Original	<i>She is charming</i>
Algoritmo Pattern Analyzer	<code>phrase = TextBlob("She is charming")</code> <code>print (phrase.sentiment)</code>
Saída dos Dados	<code>Sentiment[(polarity=0,7), (subjectivity=1,0)]</code>

Tabela 5.15 – Ataque adversarial através da substituição de palavras na frase "*She is **seductive***".

Entrada - Amostra Adversarial	<i>She is seductive</i>
Algoritmo Pattern Analyzer	<code>phrase = TextBlob("She is seductive")</code> <code>p = phrase.correct()</code> <code>print (p)</code> <code>print (p.sentiment)</code>
Saída dos Dados	<i>She is seductive</i> <code>Sentiment[(polarity=0,0), (subjectivity=0,0)]</code>

5.2.2 Substituição em Frases com Palavras de Negação

Esse ataque inverte a polaridade de uma frase negativa para positiva. Esse tipo de ataque é interessante porque pode alterar a tomada de decisão em um determinado contexto de interesse de análise, se não for percebido pela ferramenta de classificação e nem pelo analista.

Conforme a Seção 4.1.1.5, as palavras de negação "no", "not" e "never" invertem a polaridade de uma palavra positiva ou negativa. Essas palavras de negação representam na frase -0,5. Assim, a palavra "not" inverte a polaridade do texto original "It's not beautiful what you are doing Alice". A Tabela 5.16 mostra o POS Tagger e os valores da polaridade, subjetividade e intensidade das palavras do texto original. Verifica-se que apenas a palavra "beautiful" tem valores [(p=0,85), (s=1,0) e (i=1,0)] na frase. As Equações 5.5 e 5.6 e a Tabela 5.17 apresentam o cálculo da polaridade e da subjetividade dessa frase (sentimento negativo).

Tabela 5.16 – POS Tagger, polaridade, subjetividade e intensidade das palavras da frase "It's not beautiful what you are doing Alice" no conjunto de dados léxico.

Palavras	POS Tagger	Polaridade	Subjetividade	Intensidade
<i>It</i>	PRP	-	-	-
<i>'s</i>	VBZ	-	-	-
<i>not</i>	RB	-	-	-
<i>beautiful</i>	JJ	$(0,7 + 1,0)/2 = 0,85$	$(1,0 + 1,0)/2 = 1,0$	1,0
<i>what</i>	WP	-	-	-
<i>you</i>	PRP	-	-	-
<i>are</i>	VBP	-	-	-
<i>doing</i>	VBG	-	-	-
<i>Alice</i>	NNP	-	-	-

$$PFN_{(n1,x1)} = n1 * \bar{P}_{(x1)} = (-0,5) * 0,85 = -0,425 \quad (5.5)$$

$$\bar{SF}_{(s)} = \frac{1}{n} \sum_{i=1}^n \bar{S}_i = \frac{1,0}{1} = 1,0 \quad (5.6)$$

Tabela 5.17 – Cálculo da polaridade e subjetividade da frase "It's not beautiful what you are doing Alice" pelo algoritmo *Pattern Analyzer*.

Entrada - Amostra Original	<i>It's not beautiful what you are doing Alice</i>
Algoritmo Pattern Analyzer	<pre>phrase = TextBlob("It's not beautiful what you are doing Alice") p = phrase.correct() print (p.sentiment)</pre>
Saída dos Dados	<i>Sentiment[(polarity=-0,425), (subjectivity=1,0)]</i>

Para criar o texto adversarial "*Nothing* beautiful what are you doing Alice" substituímos a palavra "It's" e a palavra de negação "not" do texto original "It's not beautiful what you are doing Alice" por outra palavra de negação ("*Nothing*") que não é utilizada pelo algoritmo *Pattern Analyzer*. Percebe-se que o algoritmo classifica os dois textos original e adversarial de forma diferente, um como negativo e outro como positivo (Figura 5.9). Assim esse tipo de ataque faz o classificador inverter a percepção de sentimento, mesmo os dois textos sendo analisado por humanos como negativos.

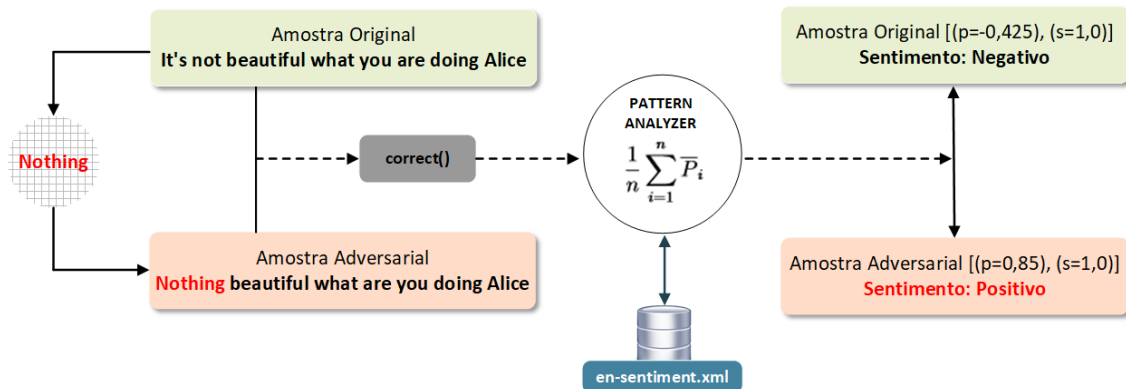


Figura 5.9 – Ataque adversarial através da substituição de palavras de negação por seus sinônimos.

As Equações 5.7 e 5.8 e a Tabela 5.18 mostram a efetividade do ataque. Observa-se que a palavra "*Nothing*" ($n1$) não possui valor porque não é utilizada pelo algoritmo para calcular a polaridade. Verifica-se também que o ataque não afeta a subjetividade do texto adversarial. A representação visual desse ataque pode ser observada na Figura 5.10.

$$\begin{aligned}
 PFN_{(n1,x1)} &= n1 * \bar{P}_{(x1)} \\
 &= \mathit{nil} * \bar{P}_{(x1)} = \frac{1}{n} \sum_{i=1}^n X_i = 0,85
 \end{aligned}
 \tag{5.7}$$

$$\overline{SF}_{(S)} = \frac{1}{n} \sum_{i=1}^n \overline{S}_i = \frac{1,0}{1} = 1,0 \quad (5.8)$$

Tabela 5.18 – Ataque adversarial através da substituição de palavras na frase "*Nothing beautiful what are you doing Alice*".

Entrada - Amostra Adversarial	<i>Nothing beautiful what are you doing Alice</i>
Algoritmo Pattern Analyzer	<pre> phrase = TextBlob("Nothing beautiful what are you doing Alice") p = phrase.correct() print (p) print (p.sentiment) </pre>
Saída dos Dados	<i>Nothing beautiful what are you doing Alice</i> Sentiment[(polarity=0,85), (subjectivity=1,0)]

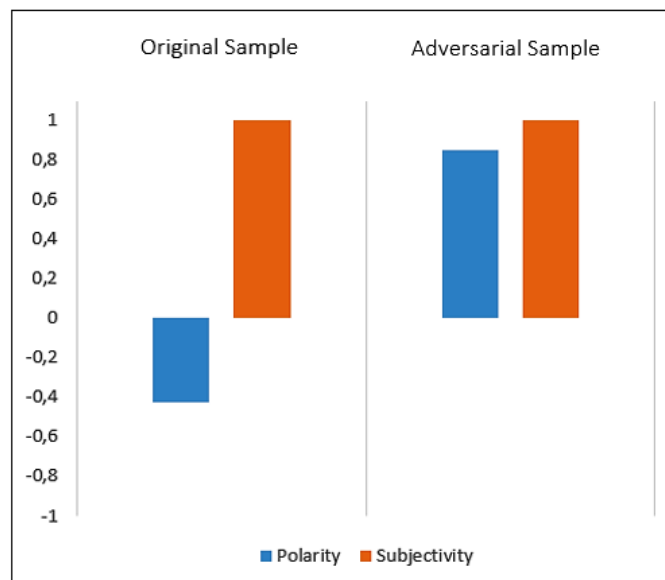


Figura 5.10 – Representação visual do ataque de substituição em frases com palavras de negação.

5.2.3 Contramedida para os Ataques de Substituição

O ataque de substituição em frases negativas ou positivas deve ser mitigado através da inserção de sinônimos no conjunto de dados léxico. Assim, as palavras "*wicked*" e "*seductive*" dos textos adversariais "*The girl is wicked*" e "*She is seductive*" serão consultadas no *corpus* pelo algoritmo para entrar no cálculo da polaridade e subjetividade.

Na Tabela 5.19 é possível observar as palavras "*wicked*" e "*seductive*" no *corpus* léxico. Elas foram inseridas no arquivo *en-sentiment.xml* com os mesmos parâmetros (POS Tagger,

polaridade, subjetividade, intensidade e confiança) das palavras "evil" e "charming" dos textos originais "The girl is evil" e "She is charming". Conforme a Seção 4.1.1.1, o arquivo *en-sentiment.xml* contém o conjunto de dados léxico que é utilizado pelo algoritmo *Pattern Analyzer* para calcular a polaridade e a subjetividade dos textos.

Tabela 5.19 – Inserção das palavras "wicked" e "seductive" no conjunto de dados léxico.

Palavras	Conjunto de Dados Léxico
wicked	<word form="wicked" pos="JJ" polarity="-1,0" subjectivity="1,0" intensity="1,0" confidence="0,9" />
seductive	<word form="seductive" pos="JJ" polarity="0,7" subjectivity="1,0" intensity="1,0" confidence="0,8" />

A Figura 5.11 mostra que após a inserção dessas palavras no conjunto de dados léxico (arquivo *en-sentiment.xml*) o sentimento dos textos adversariais foi classificado de maneira correta pelo algoritmo *Pattern Analyzer*. Agora os textos originais e adversariais tem a mesma polaridade e subjetividade.

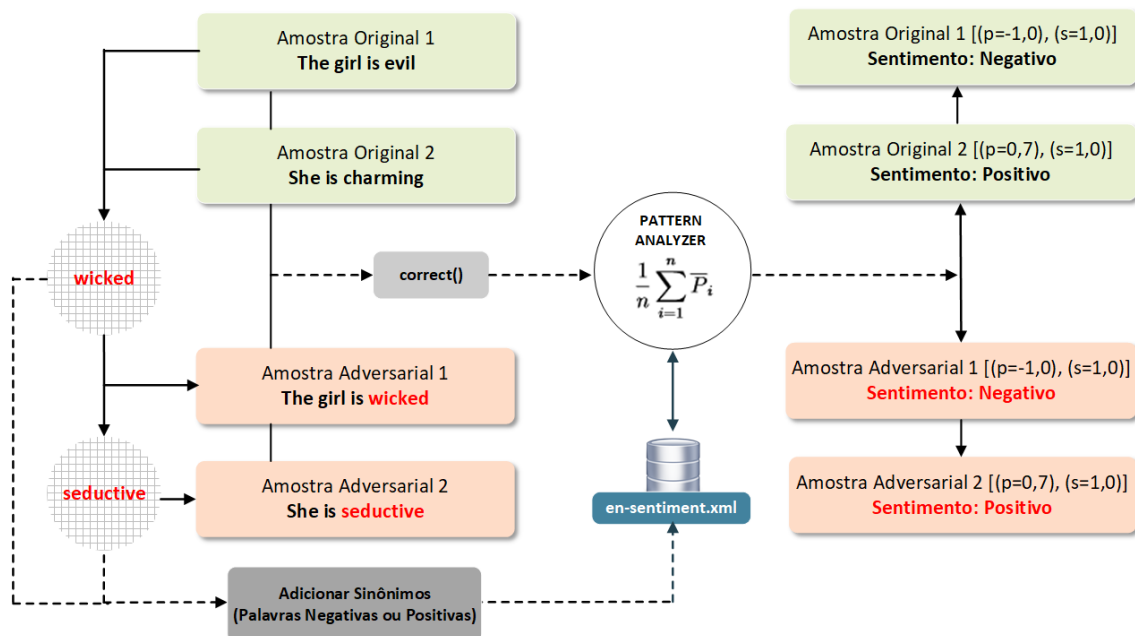


Figura 5.11 – Contramedida para o ataque de substituição em frases positivas e negativas.

A contramedida para o ataque de substituição com palavras de negação segue o mesmo processo anterior. A diferença é que nesta contramedida deve-se inserir outras palavras de negação no código do algoritmo *Pattern Analyzer*. Assim, ele utilizará o valor -0,5 da palavra inserida para calcular a polaridade e a subjetividade dos textos. Na Figura 5.12 é possível verificar que após a inserção da palavra "Nothing" o texto adversarial foi classificado pelo algoritmo como negativo.

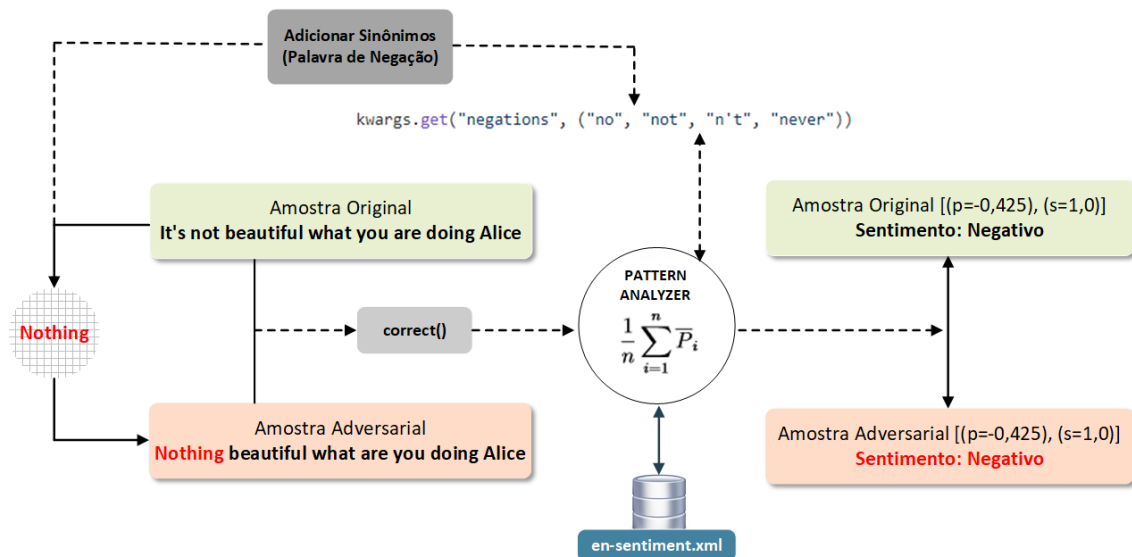


Figura 5.12 – Contramedida para o ataque de substituição em frases com palavras de negação.

5.3 RESUMO DO CAPÍTULO 5

Os resultados apresentados neste capítulo, mostram que um atacante motivado e com condições mínimas é capaz de alterar a percepção de realidade realizando ataques adversariais que interferem nos classificadores de sentimento. Um ataque bem-sucedido é capaz de gerar desinformação ou inverter uma percepção de realidade. Portanto, esses algoritmos de classificação devem ser analisados ou não deveriam ser utilizados sem elevados níveis de customização para corrigir possíveis falhas e vulnerabilidades.

6 REPRESENTAÇÃO VISUAL DOS ATAQUES NO OCTOPUSVIZ

Este capítulo tem como objetivo principal utilizar a arquitetura *OctopusViz* para validar e apresentar os resultados do processo das estratégias de ataques adversariais com seus respectivos procedimentos de defesa. Com esse propósito, inicialmente, emprega-se uma conta do *Twitter* para inundar a fonte de dados (*Twitter*) com *tweets* adversariais. Em seguida, efetua-se uma análise visual com a finalidade de mostrar como o ambiente *OctopusViz* se comporta frente aos ataques adversariais de inserção de caracteres e substituição de palavras.

A Figura 6.1 apresenta como exemplo, uma metodologia dividida basicamente em três fases, que pode ser aplicada por um usuário mal intencionado para realizar estratégias de ataques adversariais em sistemas de análise de sentimento: a) coleta de informações através de *Open Source Intelligence* (OSINT) ou técnicas de engenharia social para identificar o classificador e *corpus* utilizado pelo sistema alvo para classificar textos; b) análise de vulnerabilidades através do processo de engenharia reversa no *corpus* e classificador (Capítulo 4). Nesta fase, o possível atacante, após identificar as vulnerabilidades, implementa também as estratégias de ataques adversariais para verificar o comportamento do classificador (Capítulo 5); c) exploração das vulnerabilidades através de textos adversariais na fonte de dados utilizada pelo sistema para alterar a percepção do classificador de sentimento (Seção 6.1).

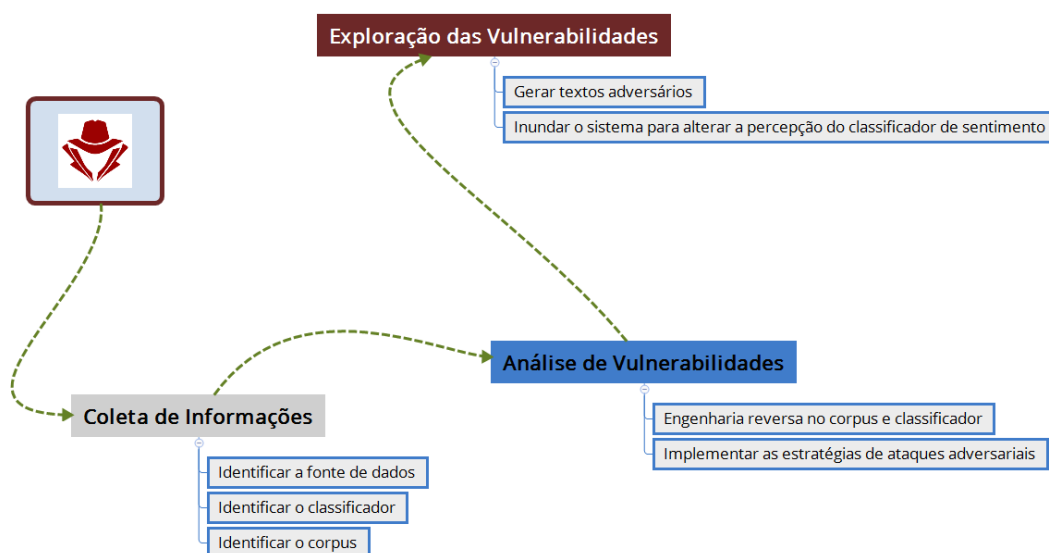


Figura 6.1 – Metodologia de ataque que pode ser aplicada por um usuário mal-intencionado.

6.1 APLICAÇÃO DOS ATAQUES NO OCOTPUSVIZ

Com o propósito de validar o processo dos ataques de inserção de caracteres e substituição de palavras no ambiente *OctopusViz*, são realizadas algumas atividades que se encontram descritas nos itens a seguir.

- Criar uma conta na rede social *Twitter* (fonte de dados do ambiente *OctopusViz*);
- Publicar *tweets* adversariais (ataques de inserção e substituição de palavras);
- Mostrar a representação visual dos ataques de inserção e substituição no *OctopusViz*.

O processo de engenharia reversa (identificar vulnerabilidades no *corpus* e classificador de sentimento léxico) e as estratégias de ataques adversariais (ataques de inserção de caracteres e substituição de palavras para verificar o comportamento do classificador) estão detalhadas nos Capítulos 4 e 5.

Para realizar os ataques de inserção e substituição utilizamos duas amostras de *tweets* originais: uma em português ("A eleição nos Estados Unidos não foi muito agradável") e outra em inglês ("It's not beautiful what are you doing mister president"). A Figura 6.2 representa o perfil da conta criada no *Twitter* (09 de novembro de 2020) para publicar os *tweets* originais e adversariais. Cabe ressaltar que a conta *cbxyz* foi utilizada apenas para esta finalidade.



Figura 6.2 – Perfil da conta criada no *Twitter* para publicar os *tweets* originais e adversariais.

A Figura 6.3 mostra como os ataques serão aplicados no ambiente *OctopusViz*. A conta *cbxyz* é usada para gerar *tweets* adversariais na plataforma (*Twitter*) que é utilizada como fonte de dados pelo *OctopusViz*.

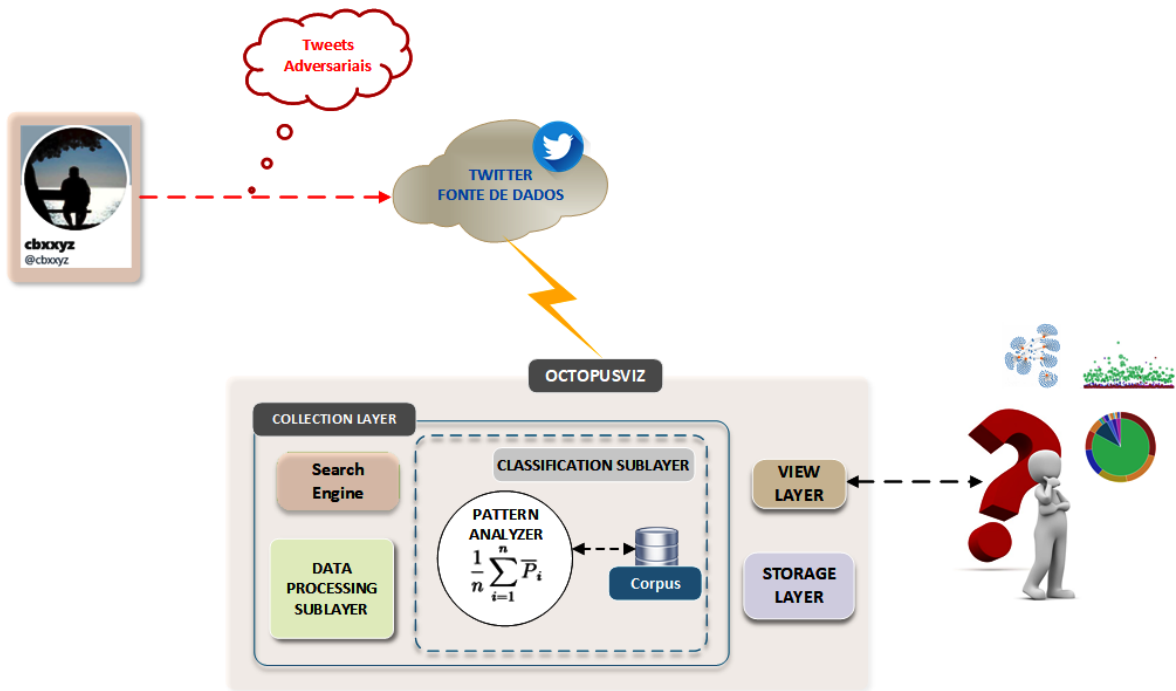


Figura 6.3 – Aplicação dos ataques no ambiente *OctopusViz*.

6.1.1 Ataques de Inserção e Substituição

O ataque de inserção consistiu na utilização da amostra de um *tweet* adversarial ("A eleição nos Estados Unidos *n.ã.o* foi muito agradável") criado a partir do *tweet* original "A eleição nos Estados Unidos não foi muito agradável". A Figura 6.4 mostra os *tweets* original e adversarial publicados pela conta *cbxyz* às 12:13hs do dia 09 de novembro de 2020.



Figura 6.4 – *Tweets* original e adversarial publicados pela conta *cbxyz*.

O segundo ataque substituiu a palavra "It's" e a palavra de negação "not" do *tweet* original ("It's not beautiful what are you doing mister president") pela palavra "Nothing" para gerar o *tweet* adversarial "Nothing beautiful what are you doing mister president". Os dois *tweets* (original e adversarial) publicados pela conta *cbxyz* às 12:13hs do dia 09 de novembro de 2020 podem ser observados na Figura 6.5.



Figura 6.5 – *Tweets* original e adversarial publicados pela conta *cbxyz*.

6.1.2 Representação Visual dos Ataques

O *OctopusViz* foi configurado para coletar os *tweets* publicados pela conta *cbxyz*. A coleta dos dados ocorreu logo após a publicação dos *tweets* originais e adversariais no *Twitter*. A camada de visualização do *OctopusViz* apresenta um *histogram* com os quatro *tweets* publicados pela conta *cbxyz* (Figura 6.6).

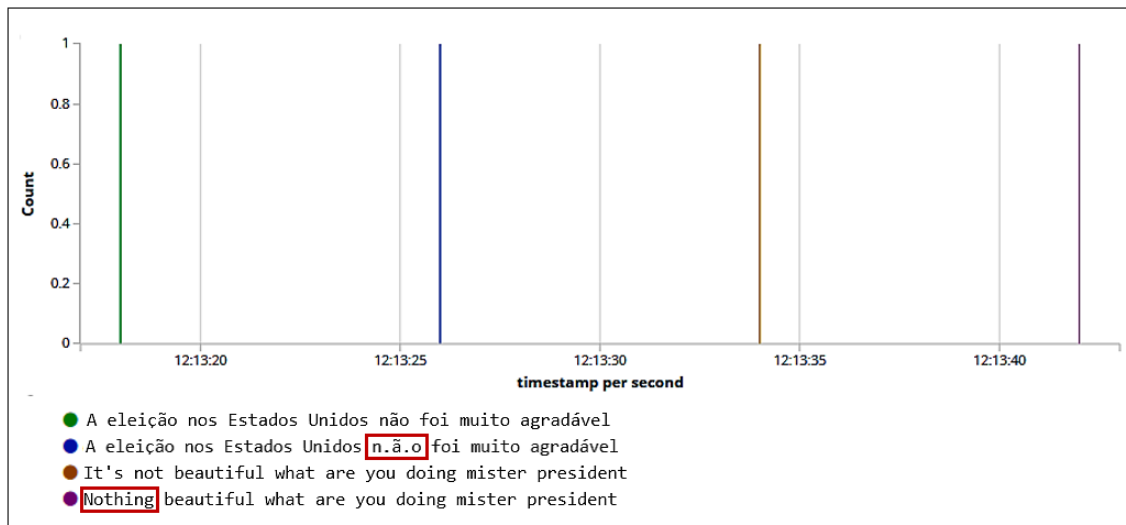


Figura 6.6 – *Histogram* com os *tweets* publicados pela conta *cbxyz*.

Nas Figuras 6.7 e 6.8 é possível verificar a representação visual dos ataques de inserção e substituição. Verifica-se que os quatro *tweets* originais e adversariais publicados pela conta *cbxyz* são classificados pelo algoritmo de forma diferente, dois como negativos ("A eleição nos Estados Unidos não foi muito agradável" - "It's not beautiful what are you doing mister president") e dois como positivos ("A eleição nos Estados Unidos *n.ã.o* foi muito agradável" - "*Nothing* beautiful what are you doing mister president").

November 9th 2020, 12:13:42.940	cbxyz		Nothing beautiful what are you doing mister president	positivo
November 9th 2020, 12:13:34.763	cbxyz		It's not beautiful what are you doing mister president	negativo
November 9th 2020, 12:13:26.552	cbxyz		A eleição nos Estados Unidos n.ã.o foi muito agradável	positivo
November 9th 2020, 12:13:18.406	cbxyz		A eleição nos Estados Unidos não foi muito agradável	negativo

Figura 6.7 – *Tweets* originais e adversariais publicados pela conta *cbxyz*.

Percebe-se que o ataque inverteu a polaridade dos *tweets* adversariais de negativa para positiva, mesmo os dois *tweets* sendo analisados por humanos como negativos. A palavra escrita errada "*n.ã.o*" e a palavra "*Nothing*" não são utilizadas pelo classificador do ambiente *OctopusViz* para calcular a polaridade dos *tweets* adversariais (Seção 5.2.2 do Capítulo 5).

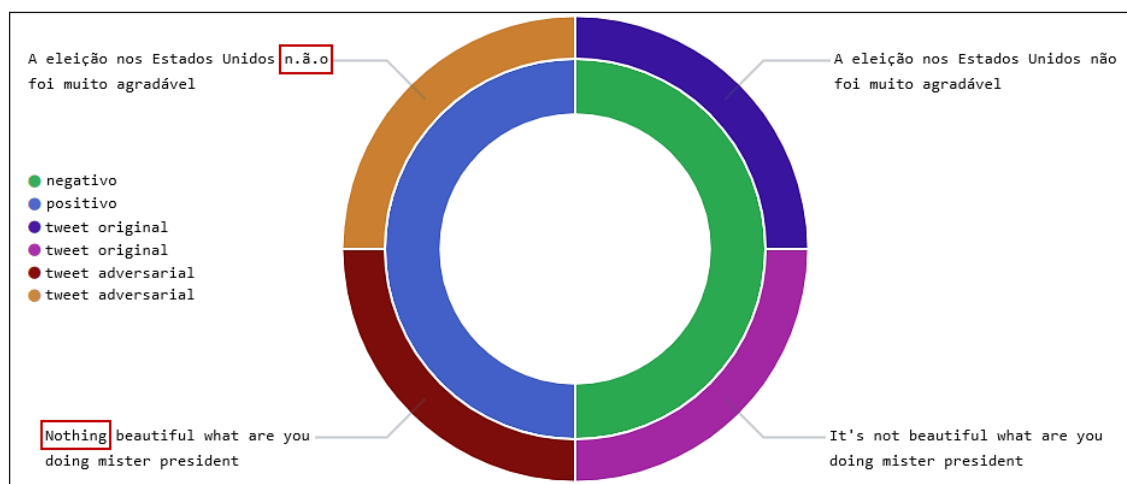


Figura 6.8 – Representação visual dos ataques de inserção e substituição no ambiente *OctopusViz*.

As Equações 6.1 e 6.2 mostram o cálculo da polaridade do *tweet* original ("A eleição nos Estados Unidos não foi muito agradável") e adversarial ("A eleição nos Estados Unidos *n.ã.o* foi muito agradável"). O *tweet* original tem uma palavra de negação ("*não*"), um advérbio

("muito") e um adjetivo ("agradável"). Assim, a Equação 6.1 utiliza o conceito de cálculo de polaridade em frases com palavras de negação, advérbios e adjetivos (Seção 4.1.1.7). A Tabela 6.1 representa o mesmo cálculo feito pelo classificador.

$$\begin{aligned}
 PFNA_{(n1,a1,x1)} &= n1 * \left(\frac{1}{I_{(a1)}}\right) * \bar{P}_{(x1)} \\
 &= (-0,5) * \left(\frac{1}{1,3}\right) * 0,73333 = -0,28205 \quad (6.1)
 \end{aligned}$$

Tabela 6.1 – Cálculo da polaridade e subjetividade do *tweet* original "A eleição nos Estados Unidos não foi muito agradável" pelo algoritmo *Pattern Analyzer*.

Entrada - Amostra Adversarial	<i>A eleição nos Estados Unidos não foi muito agradável</i>
Algoritmo Pattern Analyzer	<pre> tweet = TextBlob("A eleição nos Estados Unidos não foi muito \ agradável") if tweet.detect_language() != 'en': translate_to_english = TextBlob(str(tweet.translate(to='en'))) t = translate_to_english.correct() print (t) sentiment.polarity(t) else: t = tweet.correct() print (t) sentiment.polarity(t) </pre>
Saída dos Dados	<pre> The election in the United States was not very pleasant Sentiment[(polarity=-0,28205), (subjectivity=0,74358)] Polarity: Negative </pre>

No *tweet* adversarial, a palavra escrita errada "n.ã.o" não está no código do classificador. Neste caso, o algoritmo desconsidera o valor (-0,5) referente a palavra de negação ("não") e utiliza o conceito de cálculo de polaridade e subjetividade em frases com advérbios e adjetivos (Seção 4.1.1.6). Assim, o *tweet* adversarial que é analisado por um humano como negativo é classificado pelo algoritmo como positivo. A Equação 6.2 mostra a efetividade do ataque. A Tabela 6.2 representa o mesmo cálculo feito pelo classificador. É possível observar na saída dos dados do *script* que o algoritmo remove do *tweet* adversarial a palavra com erros de ortografia ("n.ã.o"). Assim, o algoritmo utiliza no cálculo da polaridade apenas o advérbio "very" e o adjetivo "pleasant".

$$\begin{aligned}
PFA_{(a1,x1)} &= I_{(a1)} * \bar{P}_{(x1)} \\
&= 1,3 * 0,73333 = 0,9533
\end{aligned}
\tag{6.2}$$

Tabela 6.2 – Cálculo da polaridade e subjetividade do *tweet* adversarial "A eleição nos Estados Unidos n.ã.o foi muito agradável" pelo algoritmo *Pattern Analyzer*.

Entrada - Amostra Adversarial	<i>A eleição nos Estados Unidos n.ã.o foi muito agradável</i>
Algoritmo Pattern Analyzer	<pre> tweet = TextBlob("A eleição nos Estados Unidos n.ã.o foi muito \ agradável") if tweet.detect_language() != 'en': translate_to_english = TextBlob(str(tweet.translate(to='en'))) t = translate_to_english.correct() print (t) sentiment.polarity(t) else: t = tweet.correct() print (t) sentiment.polarity(t) </pre>
Saída dos Dados	<pre> The election in the United States was very pleasant Sentiment[(polarity=0,9533), (subjectivity=1,0)] Polarity: Negative </pre>

6.2 CONTRAMEDIDAS PARA MITIGAR OS ATAQUES NO OCTOPUSVIZ

As contramedidas foram implementadas para mitigar os dois ataques (inserção de caracteres e substituição de palavras). O ataque de inserção foi mitigado através dos métodos de correção (dicionários português e inglês) e tradução, implementados na subcamada de pré-processamento do *OctopusViz* (Seção 5.1.2.1). Para mitigar o ataque de substituição, inseriu-se no código do classificador a palavra de negação "*Nothing*" (Seção 5.2.3). A Figura 6.9 representa os *tweets* adversariais publicados novamente pela conta *cbxyz* às 14:09hs do dia 13 de novembro de 2020.

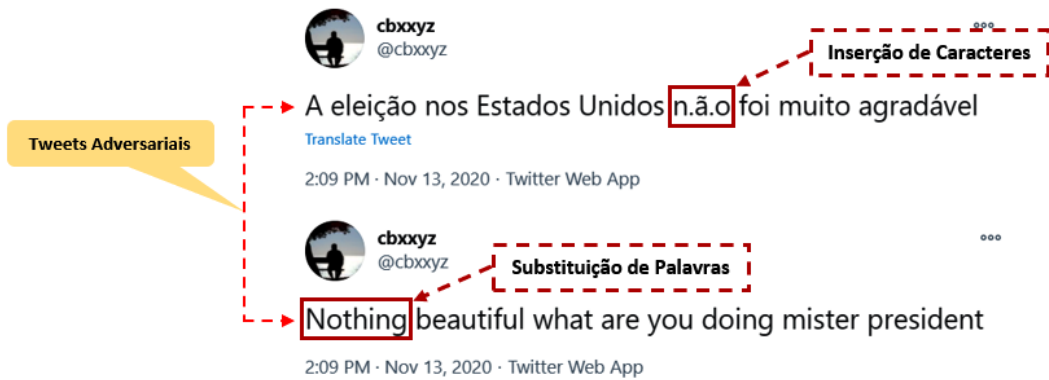


Figura 6.9 – *Tweets* adversariais publicados pela conta *cbxyz*.

A Figura 6.10 mostra um *histogram* com os dois *tweets* coletados pelo ambiente *OctopusViz*. Percebe-se no *tweet* adversarial ("A eleição nos Estados Unidos *n.ã.o* foi muito agradável") que a palavra de negação escrita errada ("*n.ã.o*") foi corrigida pelo primeiro método de correção (dicionário escrito em português) implementado na subcamada de pré-processamento do ambiente.

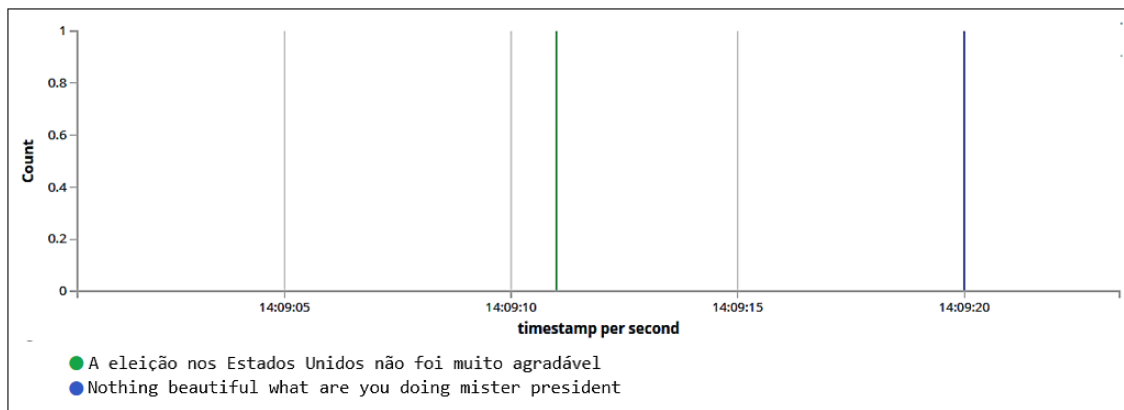


Figura 6.10 – *Histogram* com os *tweets* adversariais publicados pela conta *cbxyz*.

Na Figura 6.11 é possível observar que os dois *tweets* foram classificados como negativos. Os ataques de inserção e substituição foram mitigados pelos métodos de correção, tradução e inserção da palavra de negação "*Nothing*" no código do classificador.



▶	November 13th 2020, 14:09:20.311	cbxyz		Nothing beautiful what are you doing mister president	negativo
▶	November 13th 2020, 14:09:11.767	cbxyz		A eleição nos Estados Unidos não foi muito agradável	negativo

Figura 6.11 – *Tweets* adversariais mitigados pelos métodos de correção, tradução e inserção de palavras de negação no código do classificador.

6.3 RESUMO DO CAPÍTULO 6

Este capítulo mostrou que é possível inverter a percepção do classificador de sentimento léxico (algoritmo *Pattern Analyzer*) utilizado na subcamada de classificação do ambiente *OctopusViz*. As contramedidas implementadas foram capazes de mitigar os dois tipos de ataques (inserção de caracteres e substituição de palavras).

Esta tese desenvolveu e validou duas estratégias de ataques adversariais *white-box* com seus respectivos procedimentos de defesa em um classificador léxico de linguagem natural que é muito utilizado em diversas aplicações de mídia social para classificar textos. Para tanto, uma arquitetura de coleta, denominada de *OctopusViz*, foi desenvolvida com esse classificador e colocada em produção para estudar, identificar vulnerabilidades e testar as estratégias dos ataques.

Assim, os resultados desse trabalho podem ser divididos em duas partes, uma com relação a eficiência e eficácia da arquitetura de coleta, e a outra para as estratégias dos ataques adversariais no classificador de sentimento léxico implementado na subcamada de classificação dessa arquitetura.

7.1 ESTRATÉGIAS DE ATAQUES ADVERSARIAIS

Os resultados dos ataques propostos e implementados neste trabalho mostram a fragilidade de determinados algoritmos de classificação utilizados em sistemas de análise de sentimentos, demonstrando que eles podem ser influenciados para gerar percepção equivocada. Nesse sentido, se tais sistemas que coletam dados de redes sociais, são utilizados para gerar consciência situacional de uma situação real (eleições, greves, protestos, ciberataques, etc.), um atacante, de posse de informações sobre o tipo de sistema e poucos detalhes sobre o *corpus* e o algoritmo de classificação utilizado, pode programar *bots* de mídias sociais para gerar informações nas plataformas com conteúdo em número mais elevado do que a quantidade real de postagens com o intuito de alterar a percepção da ferramenta de apoio a tomada de decisão. Nesse caso em particular, se a percepção real é de que algo irá acontecer devido ao tipo de texto e frases nos conteúdos, dependendo da efetividade do ataque e da capacidade do atacante, essa percepção pode ser alterada para que o sistema apresente dados com informações de que não está acontecendo nada ou de que algo teria probabilidade reduzida de acontecer, dado o grande volume de dados coletados, mas não lidos efetivamente por seres humanos e nem por ferramentas ou algoritmos que tenham a correta percepção dos textos. Os nossos resultados, conforme demonstrado nos ataques implementados, indicam essas possibilidades em tais sistemas.

O resumo das contribuições deste trabalho está relacionado à aplicação de duas estratégias de ataques adversariais *white-box* em um classificador de sentimento léxico que é muito utilizado em ambientes de coleta para calcular a polaridade e a subjetividade dos textos em

aplicações de mídia social. Do ponto de vista dos classificadores de sentimento, para fundamentar este trabalho, foram discutidos os aspectos relativos a análise de sentimentos na classificação de textos. Do lado da aplicação dos ataques adversariais, o código do classificador de sentimento (*Pattern Analyzer*) da biblioteca *TextBlob* e o conjunto de dados léxico foi estudado e utilizado para identificar vulnerabilidades com a finalidade de desenvolver estratégias de ataques adversariais.

Os resultados demonstraram após a implementação dos ataques, inclusive no ambiente *OctopusViz*, que aplicações que utilizam esse algoritmo para classificar sentimento podem ser enganadas através dos seguintes ataques: a) inserção de caracteres (ruídos com erros de ortografia) em palavras de frases em inglês ou em outros idiomas (Seção 5.1); b) substituição de palavras negativas ou positivas por sinônimos que não estão no conjunto de dados léxico ou substituição de palavras de negação por outras palavras semelhantes que não fazem parte do código do algoritmo *Pattern Analyzer* (Seção 5.2).

Para todas as vulnerabilidades encontradas neste trabalho, foram apresentadas contramedidas que podem ser utilizadas para mitigar os ataques adversariais propostos. Além disso, para os sistemas de classificação, essas contramedidas podem oferecer vantagens computacionais se forem implementadas na fase de pré-processamento, podendo melhorar a eficácia e a precisão do classificador de sentimento. Por exemplo, grande parte dos trabalhos analisa apenas textos com o idioma inglês por não possuírem capacidade de processamento de textos com outros idiomas. Os métodos de correção e tradução aplicados para mitigar o ataque da Seção 5.1 resolve esse problema através da correção e tradução das palavras de qualquer idioma para o idioma inglês.

7.2 ARQUITETURA DE COLETA

A arquitetura *OctopusViz* se mostrou capaz de capturar grande quantidade de *tweets* em tempo real. Como diferencial, o ambiente permite realizar análise de sentimento, extração de informações, métricas e estatísticas de usuários, *hashtags*, *tweets*, *retweets*, identificação de *bots* sociais através da análise de *Outliers* e dados quantitativos que podem ser configurados de acordo com a necessidade e do interesse de quem precisa analisar dados em grande volume e com velocidade.

Como estudos de casos para validar a solução, foram detalhados os dados do *Twitter* referente aos temas "**seleção brasileira**" durante os jogos da Copa do Mundo FIFA 2018 e "**coronavírus (COVID-19)**". No primeiro estudo de caso pode-se identificar *bots*, o sentimento dos usuários com relação a seleção brasileira e a *hashtag* mais comentada nas quartas de final. No segundo estudo de caso foi possível verificar que os usuários estavam mais otimistas com relação a doença COVID-19.

A análise dos resultados indica que tais técnicas permitem a utilização do ambiente proposto em diversas aplicações de análise. O algoritmo *Pattern Analyzer* implementado no módulo *textblob.sentiments()* da biblioteca *TextBlob* se mostrou eficaz, apresentando em tempo real resultados consistentes sobre os sentimentos dos usuários (polaridade e subjetividade dos textos). A solução proposta permite ainda a visualização de detalhes de *tweets* para tomadas decisões sem o risco da influência de *bots*, uma vez que os mesmos podem ser facilmente identificados com o auxílio da ferramenta.

7.3 TRABALHOS FUTUROS

As linhas de trabalhos futuros estão divididas em duas vertentes. Na primeira pretende-se identificar outras vulnerabilidades no conjunto de dados léxico e classificador de sentimento léxico, criar novos métodos de amostras e estratégias de ataques adversariais para alterar a percepção do classificador de sentimento, utilizar *botnet* para automatizar o processo de ataque e implementar contramedidas distintas das propostas por esse trabalho para mitigar outras estratégias de ataques adversariais.

A outra linha está relacionada a arquitetura de coleta *OctopusViz*, onde pretende-se implementar novos procedimentos de defesa para mitigar outras estratégias de ataques adversariais, criar um novo *corpus* em português, testar outros analisadores de sentimentos que utilizam a abordagem léxica, utilizar o *dataset* do ambiente em algoritmos de aprendizado de máquina para identificar melhores resultados, integrar dados de outras fontes abertas, automatizar o processo de identificação de *bots* e utilizar *The Onion Router* (TOR) para coletar e analisar dados da *Deep Web* e *Dark Web*.

REFERÊNCIAS BIBLIOGRÁFICAS

- 1 MARQUES, L. K. d. S.; VIDIGAL, F. Prosumers and social networks as marketing information sources. an analysis from the perspective of competitive intelligence in brazilian companies. *Transinformação*, SciELO Brasil, v. 30, n. 1, p. 1–14, 2018. [Google Scholar].
- 2 PEREIRA-KOHATSU, J. C. et al. Detecting and monitoring hate speech in twitter. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 19, n. 21, p. 4654, 2019. [Google Scholar].
- 3 TWITTER. (2020). Disponível em: <<https://twitter.com/>> (acessado em 07 de maio de 2020).
- 4 TWITTER Company. (2020). Disponível em: <https://about.twitter.com/en_us/company.html> (acessado em 07 de maio de 2020).
- 5 ANJARIA, M.; GUDDETI, R. M. R. Influence factor based opinion mining of twitter data using supervised learning. In: IEEE. *2014 Sixth International Conference on Communication Systems and Networks (COMSNETS)*. [S.l.], 2014. p. 1–8. [Google Scholar].
- 6 RUSSELL, M. A. *Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More. 2 ed.* [S.l.]: "O' Reilly Media, Inc.", 2013. [Google Scholar].
- 7 HERNANDEZ-SUAREZ, A. et al. Social sentiment sensor in twitter for predicting cyber-attacks using 1 regularization. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 18, n. 5, p. 1380, 2018. [Google Scholar].
- 8 ZANG, Y. et al. Word-level textual adversarial attacking as combinatorial optimization. *arXiv preprint arXiv:1910.12196v3*,, 2020. [Google Scholar].
- 9 QIU, S. et al. Review of artificial intelligence adversarial attack and defense technologies. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 9, n. 5, p. 909, 2019. [Google Scholar].
- 10 TEXTBLOB. *TextBlob: Simplified Text Processing Online*. (2019). Disponível em: <<https://textblob.readthedocs.io/en/dev/index.html>> (acessado em 30 de abril de 2019).
- 11 BROOKS, M. *Human centered tools for analyzing online social data*. Tese (Doutorado) — University of Washington Libraries, 2015. [Google Scholar].
- 12 HEER, J.; SHNEIDERMAN, B. Interactive dynamics for visual analysis. *Commun. ACM*, ACM, New York, NY, USA, v. 55, n. 4, p. 45–54, 2012. ISSN 0001-0782. [Google Scholar]. Disponível em: <<http://doi.acm.org/10.1145/2133806.2133821>>.
- 13 CHIN JR., G.; KUCHAR, O. A.; WOLF, K. E. Exploring the analytical processes of intelligence analysts. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2009. (CHI '09), p. 11–20. ISBN 978-1-60558-246-7. [Google Scholar]. Disponível em: <<http://doi.acm.org/10.1145/1518701.1518704>>.

- 14 DIAKOPOULOS, N.; NAAMAN, M.; KIVRAN-SWAIN, F. Diamonds in the rough: Social media visual analytics for journalistic inquiry. In: . [S.l.: s.n.], 2010. p. 115 – 122. [Google Scholar].
- 15 DIAKOPOULOS, N.; CHOUDHURY, M. D.; NAAMAN, M. Finding and assessing social media information sources in the context of journalism. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2012. (CHI '12), p. 2451–2460. ISBN 978-1-4503-1015-4. [Google Scholar]. Disponível em: <<http://doi.acm.org/10.1145/2207676.2208409>>.
- 16 KARINE, N.; KEVIN, C. Introduction to the digital and social media track. In: *IEEE. 2016 49th Hawaii International Conference on System Sciences (HICSS)*. [S.l.], 2016. p. 1808–1808. [Google Scholar].
- 17 FERRARA, E. et al. The rise of social bots. *Communications of the ACM*, ACM, v. 59, n. 7, p. 96–104, 2016. [Google Scholar].
- 18 KITZIE, V. L.; MOHAMMADI, E.; KARAMI, A. “life never matters in the democrats mind”: Examining strategies of retweeted social bots during a mass shooting event. *Proceedings of the Association for Information Science and Technology*, Wiley Online Library, v. 55, n. 1, p. 254–263, 2018. [Google Scholar].
- 19 BOSHMAF, Y. et al. Design and analysis of a social botnet. *Computer Networks*, Elsevier, v. 57, n. 2, p. 556–578, 2013. [Google Scholar].
- 20 HWANG, T.; PEARCE, I.; NANIS, M. Socialbots: Voices from the fronts. *interactions*, ACM, v. 19, n. 2, p. 38–45, 2012. [Google Scholar].
- 21 CONOVER, M. D. et al. Political polarization on twitter. In: *Fifth international AAAI conference on weblogs and social media*. [S.l.: s.n.], 2011. [Google Scholar].
- 22 EDWARDS, C. et al. Is that a bot running the social media feed? testing the differences in perceptions of communication quality for a human agent and a bot agent on twitter. *Computers in Human Behavior*, Elsevier, v. 33, p. 372–376, 2014. [Google Scholar].
- 23 MESSIAS, J. et al. You followed my bot! transforming robots into influential users in twitter. 2013. [Google Scholar].
- 24 KRAMER, A. D.; GUILLORY, J. E.; HANCOCK, J. T. Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 111, n. 24, p. 8788–8790, 2014. [Google Scholar].
- 25 EDMAN, M.; YENER, B. On anonymity in an electronic society: A survey of anonymous communication systems. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 42, n. 1, p. 5:1–5:35, 2009. ISSN 0360-0300. [Google Scholar]. Disponível em: <<http://doi.acm.org/10.1145/1592451.1592456>>.
- 26 SCHANZENBACH, M. Hiding from big brother. In: CARLE, G.; RAUMER, D.; SCHWAIGHOFER, L. (Ed.). *Proceedings of the Seminars Future Internet (FI) and Innovative Internet Technologies and Mobile Communications (IITM), Winter Semester 2013/2014*. Munich, Germany: Chair for Network Architectures and Services, Department

- of Computer Science, Technische Universität München, 2014. (Network Architectures and Services (NET), NET-2014-03-1), p. 67–73. [Google Scholar]. Disponível em: <http://www.net.in.tum.de/fileadmin/TUM/NET/NET-2014-03-1/NET-2014-03-1_08.pdf>.
- 27 ÇALIŞKAN, E.; MINÁRIK, T.; OSULA, A.-M. Technical and legal overview of the tor anonymity network. *NATO Cooperative Cyber Defence Centre of Excellence.*, 2015. [Google Scholar].
- 28 IVPN. *Privacy Guides. Including VPN's and Threat Models Guide.* (2019). Disponível em: <<https://www.ivpn.net/privacy-guides>> (acessado em 18 de abril de 2019).
- 29 TWITTER Developer. (2020). Disponível em: <<https://developer.twitter.com/en/docs/twitter-api/rate-limits>> (acessado em 07 de maio de 2020).
- 30 DHULAVVAGOL, P. M.; BHAJANTRI, V. H.; TOTAD, S. Performance analysis of distributed processing system using shard selection techniques on elasticsearch. *Procedia Computer Science*, Elsevier, v. 167, p. 1626–1635, 2020. [Google Scholar].
- 31 JOYCE, J. et al. Monitoring distributed systems. *ACM Transactions on Computer Systems (TOCS)*, ACM New York, NY, USA, v. 5, n. 2, p. 121–150, 1987. [Google Scholar].
- 32 ELASTIC. *Elastic Stack Product Documentation.* (2019). Disponível em: <<https://www.elastic.co/guide/index.html>> (acessado em 18 de abril de 2019).
- 33 JÚNIOR, G. A. O. et al. Honeyselk: Um ambiente para pesquisa e visualização de ataques cibernéticos em tempo real. In: *XVI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais. Sociedade Brasileira de Computação*, Niterói, RJ, BR, p. 697–706, 2016. [Google Scholar].
- 34 KONONENKO, O. et al. Mining modern repositories with elasticsearch. In: *Proceedings of the 11th working conference on mining software repositories.* [S.l.: s.n.], 2014. p. 328–331. [Google Scholar].
- 35 BAJER, M. Building an iot data hub with elasticsearch, logstash and kibana. In: *IEEE. 2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW).* [S.l.], 2017. p. 63–68. [Google Scholar].
- 36 MURRAY, S. *Interactive data visualization for the web: an introduction to designing with.* [S.l.]: " O'Reilly Media, Inc.", 2017. [Google Scholar].
- 37 GERSHON, N.; PAGE, W. What storytelling can do for information visualization. *Association for Computing Machinery. Communications of the ACM*, Proquest ABI/INFORM, v. 44, n. 8, p. 31–37, 2001. [Google Scholar].
- 38 HEER, J.; BOSTOCK, M.; OGIEVETSKY, V. A tour through the visualization zoo. *Commun. ACM*, ACM, New York, NY, USA, v. 53, n. 6, p. 59–67, 2010. ISSN 0001-0782. [Google Scholar]. Disponível em: <<http://doi.acm.org/10.1145/1743546.1743567>>.
- 39 GRAY, J.; CHAMBERS, L.; BOUNEGRU, L. *The data journalism handbook: how journalists can use data to improve the news.* [S.l.]: " O'Reilly Media, Inc.", 2012. [Google Scholar].

- 40 GOOGLE. *Google Translate Online*. (2019). Disponível em: <<https://translate.google.com/>> (acessado em 30 de abril de 2019).
- 41 MARCUS, A. et al. Twitinfo: Agregating and visualizing microblogs for event exploration. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2011. (CHI '11), p. 227–236. ISBN 978-1-4503-0228-9. [Google Scholar]. Disponível em: <<http://doi.acm.org/10.1145/1978942.1978975>>.
- 42 SIJTSMA, B.; QVARFORDT, P.; CHEN, F. Tweetviz: Visualizing tweets for business intelligence. In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2016. (SIGIR '16), p. 1153–1156. ISBN 978-1-4503-4069-4. [Google Scholar]. Disponível em: <<http://doi.acm.org/10.1145/2911451.2911470>>.
- 43 RODRIGUES, G. A. P. et al. Cybersecurity and network forensics: Analysis of malicious traffic towards a honeynet with deep packet inspection. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 7, n. 10, p. 1082, 2017. [Google Scholar].
- 44 BISWAS, S. et al. Examining the effects of pandemics on stock market trends through sentiment analysis. *Xi'an Dianzi Keji Daxue Xuebao/Journal of Xidian University*, v. 14, p. 1–14, 06 2020. [Google Scholar].
- 45 SEBASTIANI, F. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 34, n. 1, p. 1–47, 2002. [Google Scholar].
- 46 HASAN, A. et al. Machine learning-based sentiment analysis for twitter accounts. *Mathematical and Computational Applications*, Multidisciplinary Digital Publishing Institute, v. 23, n. 1, p. 11, 2018. [Google Scholar].
- 47 PRABOWO, R.; THELWALL, M. Sentiment analysis: A combined approach. *Journal of Informetrics*, Elsevier, v. 3, n. 2, p. 143–157, 2009. [Google Scholar].
- 48 MEDHAT, W.; HASSAN, A.; KORASHY, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, Elsevier, v. 5, n. 4, p. 1093–1113, 2014. [Google Scholar].
- 49 RAVI, K.; RAVI, V. A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, Elsevier, v. 89, p. 14–46, 2015. [Google Scholar].
- 50 KUMAR, C. P.; BABU, L. D. Novel text preprocessing framework for sentiment analysis. In: *Smart Intelligent Computing and Applications*. [S.l.]: Springer, 2019. p. 309–317. [Google Scholar].
- 51 GOMES, H.; NETO, M. de C.; HENRIQUES, R. Text mining: Sentiment analysis on news classification. In: *2013 8th Iberian Conference on Information Systems and Technologies (CISTI)*. [S.l.: s.n.], 2013. p. 1–6. ISSN 2166-0727. [Google Scholar].
- 52 ARAQUE, O.; ZHU, G.; IGLESIAS, C. A. A semantic similarity-based perspective of affect lexicons for sentiment analysis. *Knowledge-Based Systems*, Elsevier, v. 165, p. 346–359, 2019. [Google Scholar].

- 53 MADHU, S. An approach to analyze suicidal tendency in blogs and tweets using sentiment analysis. *International Journal of Scientific Research in Computer Science and Engineering*, v. 6, n. 4, p. 34–36, 2018. [Google Scholar].
- 54 FAROOQUI, N. A.; RITIKA, A. S. Sentiment analysis of twitter accounts using natural language processing. *International Journal of Engineering and Advanced Technology*, v. 8, 2019. [Google Scholar].
- 55 PATIL, R.; GADA, N.; GALA, K. Twitter data visualization and sentiment analysis of article 370. In: IEEE. *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*. [S.l.], 2019. p. 1–4. [Google Scholar].
- 56 AHMED, M. E.; RABIN, M. R. I.; CHOWDHURY, F. N. Covid-19: Social media sentiment analysis on reopening. *arXiv preprint arXiv:2006.00804*, 2020. [Google Scholar].
- 57 NLTK. *NLTK Corpora*. (2020). Disponível em: <http://www.nltk.org/nltk_data/> (acessado em 07 de maio de 2020).
- 58 PANG, B.; LEE, L. *Corpus Movie Review*. 2004. Disponível em: <<http://www.cs.cornell.edu/people/pabo/movie-review-data/>> (acessado em 07 de maio de 2020).
- 59 RAJPUT, N. K.; GROVER, B. A.; RATHI, V. K. Word frequency and sentiment analysis of twitter messages during coronavirus pandemic. *arXiv preprint arXiv:2004.03925*, 2020. [Google Scholar].
- 60 KAUR, C.; SHARMA, A. *Twitter Sentiment Analysis on Coronavirus using Textblob*. [S.l.], 2020. [Google Scholar].
- 61 JÚNIOR, G. A. de O. et al. Anonymous real-time analytics monitoring solution for decision making supported by sentiment analysis. *Sensors, Multidisciplinary Digital Publishing Institute*, v. 20, n. 16, p. 4557, 2020. [Google Scholar].
- 62 NEETHU, M.; RAJASREE, R. Sentiment analysis in twitter using machine learning techniques. In: IEEE. *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*. [S.l.], 2013. p. 1–5. [Google Scholar].
- 63 ARYA, A. et al. A review: Sentiment analysis and opinion mining. *Available at SSRN 3602548*, 2020. [Google Scholar].
- 64 CERÓN-GUZMÁN, J. A.; LEÓN-GUZMÁN, E. A sentiment analysis system of spanish tweets and its application in colombia 2014 presidential election. In: IEEE. *2016 IEEE international conferences on big data and cloud computing (BDCloud), social computing and networking (socialcom), sustainable computing and communications (sustaincom)(BDCloud-socialcom-sustaincom)*. [S.l.], 2016. p. 250–257. [Google Scholar].
- 65 JOYCE, B.; DENG, J. Sentiment analysis of tweets for the 2016 us presidential election. In: *2017 IEEE MIT Undergraduate Research Technology Conference (URTC)*. [S.l.: s.n.], 2017. p. 1–4. [Google Scholar].
- 66 KOLCHYNA, O. et al. Twitter sentiment analysis: Lexicon method, machine learning method and their combination. In: *Cornell University Library*. [S.l.: s.n.], 2015. [Google Scholar].

- 67 MICU, A. et al. Analyzing user sentiment in social media: Implications for online marketing strategy. *Psychology & Marketing*, v. 12, n. 34, p. 1094–1100, 2017. [Google Scholar].
- 68 SINGH, A. K.; GUPTA, D. K.; SINGH, R. M. Sentiment analysis of twitter user data on punjab legislative assembly election, 2017. *International Journal of Modern Education and Computer Science*, Modern Education and Computer Science Press, v. 9, n. 9, p. 60, 2017. [Google Scholar].
- 69 POKHAREL, B. P. Twitter sentiment analysis during covid-19 outbreak in nepal. Available at SSRN 3624719, 2020. [Google Scholar].
- 70 MANGURI, K.; RAMADHAN, R.; AMIN, P. M. Twitter sentiment analysis on worldwide covid-19 outbreaks. *Kurdistan Journal of Applied Research*, p. 54–65, 05 2020. [Google Scholar].
- 71 GUPTA, I.; JOSHI, N. Enhanced twitter sentiment analysis using hybrid approach and by accounting local contextual semantic. *Journal of Intelligent Systems*, De Gruyter, v. 29, n. 1, p. 1611–1625, 2019. [Google Scholar].
- 72 ALAOUI, I. E. et al. A novel adaptable approach for sentiment analysis on big social data. *Journal of Big Data*, Springer, v. 5, n. 1, p. 12, 2018. [Google Scholar].
- 73 HASAN, A. et al. Machine learning-based sentiment analysis for twitter accounts. *Mathematical and Computational Applications*, v. 23, n. 1, 2018. ISSN 2297-8747. [Google Scholar]. Disponível em: <<http://www.mdpi.com/2297-8747/23/1/11>>.
- 74 SAHA, S.; YADAV, J.; RANJAN, P. Proposed approach for sarcasm detection in twitter. *Indian Journal of Science and Technology*, v. 10, n. 25, p. 1–8, 2017. [Google Scholar].
- 75 KUNAL, S. et al. Textual dissection of live twitter reviews using naive bayes. *Procedia computer science*, Elsevier, v. 132, p. 307–313, 2018. [Google Scholar].
- 76 PRACIANO, B. J. G. et al. Spatio-temporal trend analysis of the brazilian elections based on twitter data. In: . [S.l.: s.n.], 2018. p. 1355–1360. [Google Scholar].
- 77 TUMITAN, D.; BECKER, K. Sentiment-based features for predicting election polls: A case study on the brazilian scenario. In: *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 02*. Washington, DC, USA: IEEE Computer Society, 2014. (WI-IAT '14), p. 126–133. ISBN 978-1-4799-4143-8. [Google Scholar]. Disponível em: <<http://dx.doi.org/10.1109/WI-IAT.2014.89>>.
- 78 SOHANGIR, S.; PETTY, N.; WANG, D. Financial sentiment lexicon analysis. In: IEEE. *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*. [S.l.], 2018. p. 286–289. [Google Scholar].
- 79 SHEKHAWAT, B. S. *Sentiment Classification of Current Public Opinion on BREXIT: Naïve Bayes Classifier Model vs Python's TextBlob Approach*. Tese (Doutorado) — Dublin, National College of Ireland, 2019. [Google Scholar].
- 80 SENTIMENT140. *Dataset with 1.6 million tweets*. (2019). Disponível em: <<https://www.kaggle.com/kazanova/sentiment140>> (acessado em 13 de fevereiro de 2020).

- 81 LAKSONO, R. A. et al. Sentiment analysis of restaurant customer reviews on tripadvisor using naïve bayes. In: IEEE. *2019 12th International Conference on Information & Communication Technology and System (ICTS)*. [S.l.], 2019. p. 49–54. [Google Scholar].
- 82 ALZANTOT, M. et al. Generating natural language adversarial examples. *CoRR*, abs/1804.07998, 2018. [Google Scholar].
- 83 HOSSEINI, H. et al. Deceiving google’s perspective api built for detecting toxic comments. *arXiv preprint arXiv:1702.08138*, 2017. [Google Scholar].
- 84 PERSPECTIVE. *API that makes it easier to host better conversations*. (2017). Disponível em: <<https://www.perspectiveapi.com>> (acessado em 18 de abril de 2019).
- 85 WONG, C. Dancin seq2seq: Fooling text classifiers with adversarial text example generation. *arXiv preprint arXiv:1712.05419*, 2017. [Google Scholar].
- 86 LI, J. et al. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*, 2018. [Google Scholar].
- 87 SAMANTA, S.; MEHTA, S. Generating adversarial text samples. In: SPRINGER. *European Conference on Information Retrieval*. [S.l.], 2018. p. 744–749. [Google Scholar].
- 88 GAO, J. et al. Black-box generation of adversarial text sequences to evade deep learning classifiers. In: IEEE. *2018 IEEE Security and Privacy Workshops (SPW)*. [S.l.], 2018. p. 50–56. [Google Scholar].
- 89 TSAI, Y.-T.; YANG, M.-C.; CHEN, H.-Y. Adversarial attack on sentiment classification. In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. [S.l.: s.n.], 2019. p. 233–240. [Google Scholar].
- 90 VIJAYARAGHAVAN, P.; ROY, D. Generating black-box adversarial examples for text classifiers using a deep reinforced model. *arXiv preprint arXiv:1909.07873*, 2019. [Google Scholar].
- 91 JIN, D. et al. Is bert really robust? natural language attack on text classification and entailment. *arXiv preprint arXiv:1907.11932*, 2020.
- 92 BARBOSA, G. A. R. et al. Characterizing the effectiveness of twitter hashtags to detect and track online population sentiment. In: *CHI '12 Extended Abstracts on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2012. (CHI EA '12), p. 2621–2626. ISBN 978-1-4503-1016-1. [Google Scholar]. Disponível em: <<http://doi.acm.org/10.1145/2212776.2223846>>.
- 93 2020, L. E. *The Power of Real-Time Political Analytics at your Fingertips*. (2020). Disponível em: <<https://luxelection2020.com/>>" (acessado em 05 de outubro de 2020).
- 94 FORBES. *New Big Data Sentiment Analysis Show Potential Biden Election Landslide*. (2020). Disponível em: <<https://www.forbes.com/sites/waynerash/2020/09/29/new-big-data-sentiment-analysis-show-potential-biden-election-landslide/#efb4cfd0b63>>" (acessado em 05 de outubro de 2020).
- 95 CITRIX. *XenServer Current Release*. (2019). Disponível em: <<https://docs.citrix.com/en-us/xenserver/current-release.html>> (acessado em 30 de abril de 2019).

- 96 NETWORK, K. *Network Plugin for Kibana*. (2019). Disponível em: <https://github.com/dlumbreter/kbn_network> (acessado em 26 de novembro de 2019).
- 97 TWEETPY. *An easy-to-use Python library for accessing the Twitter API*. (2019). Disponível em: <<https://www.tweepy.org/>> (acessado em 20 de abril de 2019).
- 98 PYTHON. *Python: A programming language that lets you work quickly and integrate systems more effectively*. (2019). Disponível em: <<https://www.python.org/>> (acessado em 20 de abril de 2019).
- 99 NLTK. *Natural Language Toolkit*. (2019). Disponível em: <<https://www.nltk.org/>> (acessado em 07 de maio de 2019).
- 100 GOOGLE. *Google Images*. (2019). Disponível em: <https://images.google.com/imghp?hl=en&gl=ar&gws_rd=ssl>" (acessado em 03 de maio de 2019).
- 101 CLIPS. *Computational Linguistics & Psycholinguistics*. (2018). Disponível em: <<https://www.clips.uantwerpen.be/pattern>> (acessado em 30 de abril de 2019).
- 102 KIRCHHOFF, G. Ueber die auflösung der gleichungen, auf welche man bei der untersuchung der linearen vertheilung galvanischer ströme geführt wird. *Annalen der Physik*, Wiley Online Library, v. 148, n. 12, p. 497–508, 1847. [Google Scholar].
- 103 CARRÉ, B. *Graphs and networks*. Clarendon Press, 1979. [Google Scholar].
- 104 FOOTBALL, I. F. of A. *2018 FIFA World Cup Russia*. (2018). Disponível em: <<https://www.fifa.com/worldcup/archive/russia2018/>> (acessado em 07 de setembro de 2019).
- 105 TINEYE. *TinEye Image Recognition*. (2019). Disponível em: <<https://www.tineye.com/>>" (acessado em 03 de maio de 2019).
- 106 KEARNEY, M. W. *TweetBotOrNot: An R package for classifying Twitter accounts as bot or not*. (2019). Disponível em: <<https://github.com/mkearney/tweetbotornot>>" (acessado em 03 de abril de 2019).
- 107 WANG, H. et al. Phase-adjusted estimation of the number of coronavirus disease 2019 cases in wuhan, china. *Cell discovery*, Nature Publishing Group, v. 6, n. 1, p. 1–8, 2020. [Google Scholar].
- 108 ORGANIZATION, W. H. *Director-General's statement on IHR Emergency Committee on Novel Coronavirus (2019-nCoV), Geneva*. (2020). Available online: <[https://www.who.int/dg/speeches/detail/who-director-general-s-statement-on-ihf-emergency-committee-on-novel-coronavirus-\(2019-ncov\)](https://www.who.int/dg/speeches/detail/who-director-general-s-statement-on-ihf-emergency-committee-on-novel-coronavirus-(2019-ncov))> (accessed Sept 07, 2020).
- 109 PANG, B.; LEE, L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. [S.l.], 2004. p. 271. [Google Scholar].
- 110 TAYLOR, A.; MARCUS, M.; SANTORINI, B. The penn treebank: an overview. In: *Treebanks*. [S.l.]: Springer, 2003. p. 5–22. [Google Scholar].