FACULDADE DE ENGENHARIA DE SOROCABA - FACENS
PÓS GRADUAÇÃO EM ESPECIALIZAÇÃO EM CIÊNCIA DE DADOS

ALEX COZER ABRANTES

BRUNO ALVES COMITRE

DETECÇÃO AUTOMATIZADA DE NOTÍCIAS FALSAS:

UM ESTUDO SOBRE O RECONHECIMENTO DE INTEGRIDADE DAS INFORMAÇÕES

Tese apresentada ao Programa de Pós-Graduação em Ciências de Dados da Faculdade de Engenharia de Sorocaba - FACENS, como requisito parcial para a obtenção do título de Pós Graduado em Ciência de Dados.

Orientador: Prof. Matheus Mota
Coordenador: Prof. Fernando Vieira da Silva

SOROCABA

2019

Preâmbulo

Dedicatória

Este trabalho é dedicado a Alan Turing, gênio que dedicou sua vida ao avanço da ciência da computação, que tem como objetivo buscar métodos computacionais que possam permitir a capacidade de resolver problemas cada vez mais complexos através da evolução tecnológica que, hoje, nos permite viver melhor e sonhar cada vez mais alto. Dedicamos este trabalho a ele como forma de reiterar nossa profunda gratidão e admiração por sua dedicação, já que, em vida, estes sentimentos lhes foram negligenciados.
Dedicamos este trabalho, também, aos nossos pais, que a despeito das nossas muitas ausências, sempre estiveram presentes.

Agradecimentos

Gostaríamos de aproveitar este espaço para agradecer:

À FACENS, por nos proporcionar dias de muita aprendizagem.

Aos professores, pelo esforço *gigante*, pela paciência, pela sabedoria e, principalmente, pelos recursos técnicos e pessoais a nós ofertados através da exposição à novas experiências.

Às nossas queridas famílias e queridos amigos, pelos incentivos, inspirações, gestos e palavras que nos ajudaram a superar todas as dificuldades e ausências.

À todas as pessoas que de alguma forma nos ajudaram e nos inspiraram, acreditaram, torceram, mas que, injustamente, aparecem anônimas nestes gradecimentos. À cada um de vocês, o nosso sincero e profundo agradecimento.

Resumo

A propagação da desinformação através de meios digitais de comunicação, popularmente conhecida pelo termo “fake news”, ganhou a atenção da sociedade nos últimos anos. Seja para obter algum tipo de vantagem política, científica ou social, seja por um erro honesto durante a validação das informações, é notório que a propagação da desinformação pode causar prejuízos graves à sociedade. Em uma sociedade cada vez mais digitalmente conectada, a capacidade de viralização de notícias cresce e cria demanda por mecanismos de assistência ao processo de detecção automática de conteúdo errado ou dissimulado. Este trabalho descreve uma estratégia para detecção de notícias falsas baseada em aprendizado de máquina. Para isso, propõe-se um modelo capaz de aprender padrões a partir de textos previamentes anotados como desinformação para posterior classificação de novos conteúdos. Este modelo é baseado em Rede Neural Recorrente e Deep Learning, tendo como principal diferencial a preocupação em resolver problemas de disseminação de notícias falsas. Conforme detalhamento apresentado neste trabalho, o modelo comportou-se de maneira satisfatória para o conjunto de dados utilizado, podendo ser utilizado como uma ferramenta de auxílio ao processo de classificação de conteúdo.

PALAVRAS-CHAVE: *Fake News*; Dados; Comunicação; Aprendizado de Máquina.

Abstract

The spread of misinformation through digital media, popularly known as the term "fake news", has garnered society's attention in recent years. Whether to gain some kind of political, scientific or social advantage, or an honest error in validating information, it is well known that the spread of misinformation can cause serious harm to society. In an increasingly digitally connected society, the capacity for news viralization grows and creates demand for mechanisms to assist in the process of automatically detecting wrong or concealed content. This paper describes a strategy for detecting false news related to machine learning. To do so, it offers a model capable of learning patterns from previous annotated texts as misinformation for later classification of new content. This model is based on the Recurring Neural Network and Deep Learning, having as main differential the concern to solve false news dissemination problems. As detailed in this paper, the model behaves satisfactorily for the data set used and can be used as an aid in the content classification process.

KEYWORDS: Fake News; Data; Communication; Machine Learning.

SUMÁRIO

Introdução e Contextualização

No livro intitulado *Réflexions d'un historien sur les fausses nouvelles de la guerre* (Allia, 2012) [1], que pode ser traduzido como *Reflexões de um historiador sobre as falsas notícias da guerra*, publicado originalmente em 1921 por Marc Bloch, pode-se ler: *"Histórias falsas aumentaram a multidão"* [1].
Segundo o autor, as notícias falsas, em todos os seus meios e formatos - contos simples, imposturas, lendas -, sempre permearam a humanidade. *"Como eles surgem?" "De quais elementos são compostas?" "Como se espalham e ganham força?" "Quais os meios que potencializam sua viralização?"* Mesmo antes da revolução na comunicação, estas eram perguntas que Marc Bloch, escritor e historiador assassinado pelos nazistas em 1944, tentou responder a partir da década de 20 -- quando retornou das trincheiras da Primeira Guerra Mundial obcecado com a importância que as notícias falsas haviam tido neste capítulo da história. Mesmo sendo um problema antigo, notícias falsas seguem afetando a humanidade e, por isso, são objeto de intenso estudo nas mais diversas áreas do conhecimento, da psicologia à computação.
A propagação de notícias falsas por um indivíduo podem ser intencionais ou não intencionais, usualmente dependente do grau de ciência a respeito da presença ou não da desinformação no conteúdo. Muitos estudos cientìficos indicam que diversos fatores podem influenciar este processo, desde aspectos sociais (vantagens políticas, econômicas etc.) até aspectos cognitivos (viés de confirmação e envolvimento emocional com o conteúdo prejudica a capacidade crítica do consumidor).
Um estudo avaliou a disseminação de importantes notícias falsas e estimou que o americano médio teve contado com de um a três conteúdos produzidos por produtores de conteúdo falso no mês anterior à eleição de 2016 [3]. Um segundo estudo mostrou que informações falsas são muito mais compartilhadas no Twitter do que informações verdadeiras, especialmente quando o assunto está associado à questões com forte apelo emocional -- como assuntos políticos [4]. Este estudo apontou, também, que entre 9% e 15% dos usuários ativos do Twitter são robôs (perfis falsos controlados por uma autoridade central). Já no Facebook, estima-se que existam 60 milhões de robôs se passando por usuários. O trabalho apresenta ainda fortes evidências que uma parte substancial do conteúdo político publicado durante a campanha eleitoral dos EUA de 2016 e da eleição francesa de 2017 tenha partido destas contas-robôs.
Este trabalho investigou o desafio da classificação automática de notícias falsas e aplicou estratégias e tecnologias disponíveis para assistir o processo de identificação deste tipo de conteúdo.

Evolução da Pesquisa

A primeira hipótese a ser investigada por este trabalho foi a de que a desinformação está fortemente associada ao discurso ofensivo e de ódio, de tal maneira que conteúdos com sinais de discurso de ódio e liguagem ofensiva poderiam ser um vetor mais recorrente de notícias falsas.

A partir desta hipótese, este trabalho investigou a criação de um modelo de aprendizado de máquina para reconhecimento de linguagem natural a partir de um conjunto de dados classificados quanto à sua natureza do discurso (duas classes: discurso de ódio e não discurso de ódio). Tal modelo, então, seria utilizado para classificação de uma segunda base contendo dados classificados como notícias falsas. Esta primeira parte do trabalho teve como base a pesquisa *"Automated Hate Speech Detection and the Problem of Offensive Language"* [5].

Segundo Victoria Rubin em seu trabalho *"Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News"* [6], existem nove pontos importantes para a coleta dos dados:

- Paridade de notícias verdadeiras e falsas
- Notícias em versão texto
- Possibilidade de confirmar a verdade
- Homogeneidade no tamanho dos artigos
- Homogeneidade no estilo de escrita
- Janela de tempo bem definida para coleta
- Definir o estilo de notícia
- Fatores pragmáticos, como custo de aquisição e disponibilidade das notícias
- Definir idioma e cultura das notícias

Por dificuldades com as divergências estruturais entre as bases de dados, em um primeiro momento, esta pesquisa procurou construir um dataset próprio a partir da extração automática de conteúdo de fontes de conteúdo disponíveis na Web (sites de notícia e blogs). Após a extração, seria feito um processo de anotação manual do conteúdo estraído quanto à veracidade das informações. Por limitações de tempo, este processo foi interrompido e o foco passou a ser o desenvolvimento de um modelo para aprendizado de máquina que fosse capaz de identificar uma notícia falsa a partir de datasets já disponibilizados na Web. No entanto, como trabalho futuro, esta pesquisa pretende retomar a investigação da associação entre a análise de sentimento do conteúdo e a veracidade das informações.

A estrutura hierarquica abaixo, serializada em formato JSON, apresenta as features que o crawler inicialmente implementado deveria extrair das fontes selecionadas.

```
{"article": {
   "id": "number sequencial",
   "type": "object",
   "url": "string",
   "base_url": "string",
   "main_language": "string",
   "authors": "string",
   "title": "string",
   "subtitle": "string",
   "body": {
        "paragraphs": "qtdy paragraphs",
        "paragraphs_body": "array each paragraphs",
        "paragraphs_main_entities": "array each main entities",
        "number_of_words": "string",
        "tags": "string",
        "images_url": "string",
        "image_legends": "string",
        "datetime": "string"
    }
 }
}
```


Encontramos um conjunto de dados em um desafio do site Kaggle, intitulado: Fake News - Build a system to identify unreliable news articles. A pesquisa contém 20.000 dados extraídos na web e estão rotulados como confiável e não confiável. O conjunto de dados é composto por artigos de âmbito político e no idioma inglês, portanto todo o estudo seria baseado neste idioma.

- **Link: https://www.kaggle.com/c/fake-news/data** (https://www.kaggle.com/c/fake-news/data)

O Schema do conjunto de dados se aproximou com o que estávamos planejando, e por esse motivo nos ajudou na escolha desse desafio:

Conceitos e Trabalhos Relacionados

Nessa seção será abordado a definição de conceitos prévios necessários para um melhor entendimento desse trabalho.

Neurônio

Biologicamente o cérebro humano possui bilhões de neurônios conectados entre si, constituído por um corpo celular ligado por ramificações. Fazendo assim uma sinapse que faz a ligação entre dois neurônios, ligados e através de pulsos nervosos que se comunicam, e, os mesmos são recebidos e processados até atingir um limiar de ação, que produz uma substância neurotransmissora que flui do corpo celular, completando assim o funcionamento de um neurônio.

Redes Neurais

As Redes Neurais (RN) são técnicas computacionais que apresentam um modelo inspirado na estrutura neural do cérebro humano, ou seja, procuram adquirir seu conhecimento através da experiência (HAYKIN, 1998).[7].

RNA

As Redes Neurais Artificiais (RNAs) apresentam modelos matemáticos inspirados na estrutura do córtex cerebral humano capaz de adquirirem conhecimentos por meio de experiências e padrões. Estes modelos são formados por funções de ativações que são responsáveis por realizar o processo da combinação linear das entradas e dos pesos ajustando a saída de cada neurônio formando uma rede complexa de processamento.

RNN

Segundo Marcus Liwicki (2019, p.3):"[...]Redes neurais recorrentes (RNNs) são uma escolha natural para reconhecimento de manuscrito online, pois, eles podem acessar uma ampla gama de contexto ao transcrever letras, palavras ou sequências de palavras. Até recentemente, no entanto, as RNNs estavam limitadas a fazer classificações separadas a cada passo do tempo em uma sequência de entrada.[...]"[8].

Machine Learning

Existem inúmeros tipos de algoritmo de aprendizado de máquinas. E estes algoritmos se diferenciam entre si pelo método de classificação e modificação dos pesos e de suas conexões. E entre os métodos mais conhecidos podemos ter como referências: Aprendizado Supervisionado; Aprendizado Não Supervisionado; Aprendizado Hebbiano; Aprendizado por Reforço; Aprendizado por Competição.

Com base na leitura dos métodos, optamos nesta pesquisa a utilização como base o aprendizado de máquina supervisionado por reforço.

Deep Learning

Deep Learning é uma subclasse da inteligência artificial que está em constante ascensão e é capaz de ser aplicada em vários tipos de tecnologias que requerem grande volume de dados. Alguns exemplos são: Reconhecimento de fala; reconhecimento de caligrafia; classificação de imagens, entre outras aplicações [9].

Para resolver os problemas da RNN, Hochreiter & Schmidhuber introduziram um modelo de Rede de Memória Longa de Curto Prazo (LSTM) capaz de manter o fluxo do erro constante através de unidades camadas de ”portões” (gates), que permitem ajustes de pesos da mesa forma que o trucamento da sequência quando a informação é desnecessária.(MICHAEL, 2017,p.28)[10]

Sumarização

Entre as tarefas aplicadas neste trabalho, a tratativa dos textos chamado Text Summarization foi utilizada para a criação e utilização de derivações do texto sendo necessário a derivação de features, a fim de obter atributos quantitativos e nominais, que capta as ideias principais de uma notícia, não se limitando apenas a escolhas de apenas alguns trechos para análise.

Algoritmos de Aprendizados

O algoritmo de aprendizado escolhido para o desenvolvimento do modelo em Redes Nerais foi a rede LSTM, que é uma rede baseada em comportamentos. Sua vantagem é a capacidade de gerenciar toas as palavras antecessoras, não se limitando A um limite fixo, porém, o seu problema está na dificuldade de realizar o treinamento do modelo, dado o tempo de demora relacionado a capacidade (processamento) da máquina, que irá realizar esta tarefa.

Overview do Dataset Alvo

Este capítulo apresenta um overview do conjunto de dados, descrevendo e demonstrando a análise descritiva, análise gráfica e os códigos utilizados no aprendizado de máquina.

A melhor forma de começar uma análise descritiva é verificar os tipos de variáveis disponíveis e classificá-las de acordo com seus tipos.

Neste estudo foi utilizado a biblioteca em Python chamada pandas_profiling para gerar um relatório com a análise exploratória, esta biblioteca nos auxiliou muito nessa etapa.

Overview

Dataset info

Number of variables	5
Number of observations	20800
Missing cells	2554 (2.5%)
Duplicate rows	0 (0.0%)
Total size in memory	812.6 KiB
Average record size in memory	40.0 B

Variables types

Numeric	1
Categorical	3
Boolean	1
Date	0
URL	0
Text (Unique)	0
Rejected	0
Unsupported	0

Warnings

`author` has a high cardinality: 4202 distinct values	Warning
`author` has 1957 (9.4%) missing values	Missing
`text` has a high cardinality: 20387 distinct values	Warning
`title` has a high cardinality: 19804 distinct values	Warning
`title` has 558 (2.7%) missing values	Missing

Variables

author
Categorical

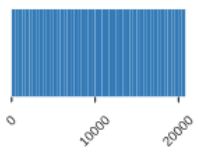
Distinct count	4202
Unique (%)	20.2%
Missing (%)	9.4%
Missing (n)	1957

Pam Key	243
admin	193
Jerome Hudson	166
Other values (4198)	18241
(Missing)	1957

Toggle details

id
Numeric

Distinct count	20800	Mean	10399.5
Unique (%)	100.0%	Minimum	0
Missing (%)	0.0%	Maximum	20799
Missing (n)	0	Zeros (%)	< 0.1%
Infinite (%)	0.0%		
Infinite (n)	0		



Toggle details

label
Boolean

Distinct count	2
Unique (%)	< 0.1%
Missing (%)	0.0%
Missing (n)	0

1	10413
0	10387

Toggle details

text
Categorical

Distinct count	20387
Unique (%)	98.0%
Missing (%)	0.2%
Missing (n)	39

	75
Leave a Reply Click here to ge...	28
Click Here To Learn More Abo...	26
Other values (20383)	20632
(Missing)	39

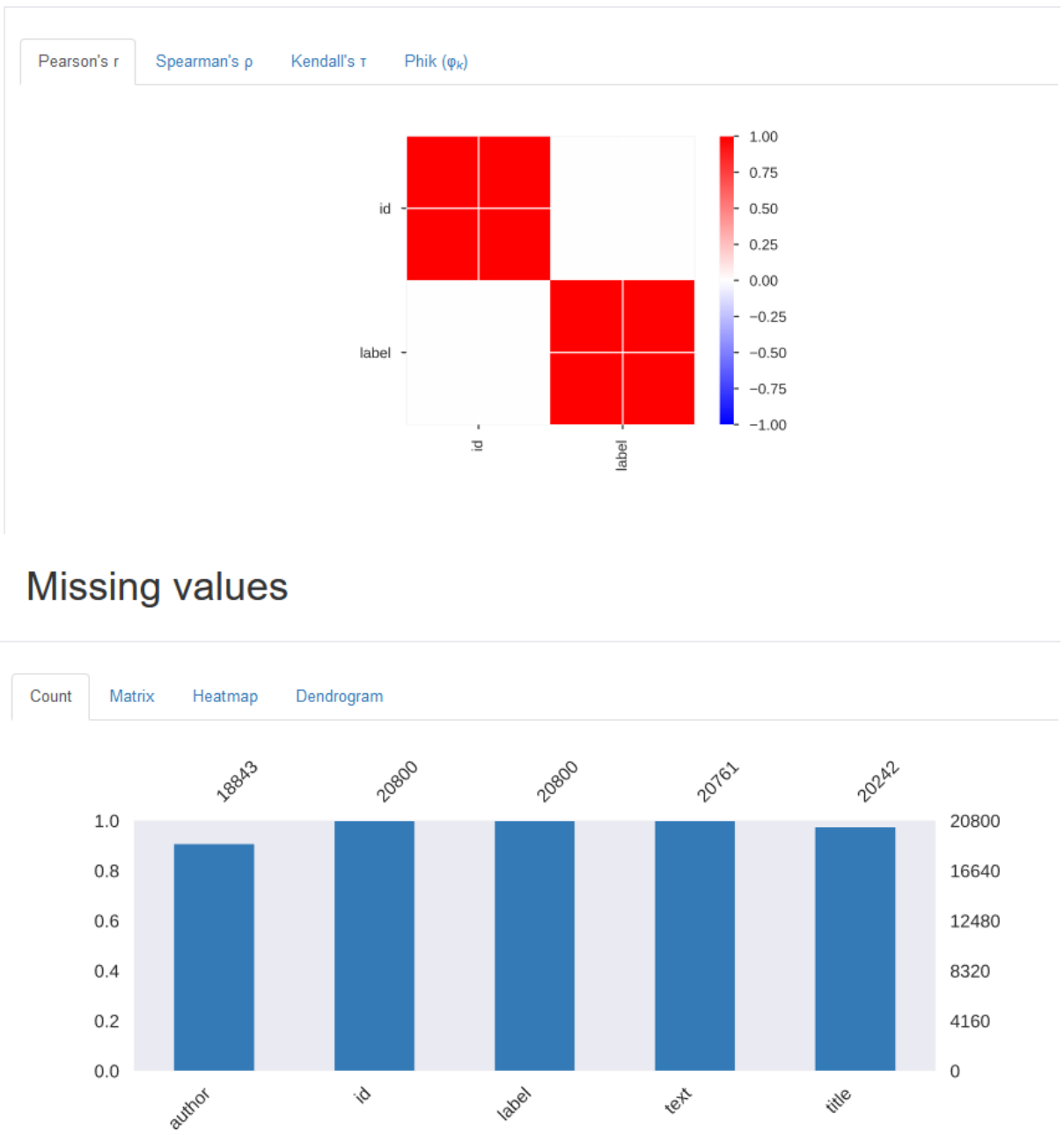
Toggle details

title
Categorical

Distinct count	19804
Unique (%)	95.2%
Missing (%)	2.7%
Missing (n)	558

The Dark Agenda Behind Glob...	5
Get Ready For Civil Unrest: S...	5
Televisión: lo más visto ayer	4
Other values (19800)	20228
(Missing)	558

Toggle details



Incompletude no Dataset

Pudemos observar a existência de incompletude no conjunto de dados, cerca de 2.554 registros contêm células com valores nulos. O atributo Autor é o que mais possui amostras ausentes, cerca de 9,41%, o atributo Título tem 2,68% e o Texto 0,19%.

Após esta observação, realizamos o tratamento adequado eliminando os registros com valores nulos.

Então, nosso conjunto de dados ficou com 18.285 registros após a eliminação.

Derivação de Features

A análise de texto com conteúdo extenso não é uma tarefa fácil de se realizar, embora seja possível através da extração de informações obter uma melhor compreensão do corpus do mesmo.

Para estudar o comportamento do texto foi necessário a derivação de features, a fim de obter atributos quantitativos e nominais.

```
Index(['id', 'title', 'author', 'text', 'label', 'title_token',
       'title_token_distint', 'title_comprimento', 'title_num_palavras',
       'title_num_palavras_unicas', 'title_palavras_vs_unico',
       'title_substantivos', 'title_adjetivos', 'title_verbos',
       'title_contagem_substantivos', 'title_contagem_adjetivos',
       'title_contagem_verbos', 'text_token', 'text_token_distint',
       'text_comprimento', 'text_num_palavras', 'text_num_palavras_unicas',
       'text_palavras_vs_unico', 'text_substantivos', 'text_adjetivos',
       'text_verbos', 'text_contagem_substantivos', 'text_contagem_adjetivos',
       'text_contagem_verbos'],
      dtype='object')
```

Desbalanceamento de Classes e Random Shuffle

A eliminação dos dados faltantes trouxe um problema de desbalanceamento no conjunto, cerca de 2.437 registros constam a mais com o valor de target igual a 1.

Para manter o conjunto de dados equilibrado, eliminamos randomicamente 2.437 registros da target igual a 1, resultando um total de 15.848 registros distribuídos igualmente em 7.924 confiáveis e não confiáveis.

Análise Gráfica do Dataset

Os gráficos são recursos utilizados para explanar eventos que possam ser mensurados e quantificados, fornecendo uma dimensão estatística sobre um determinado fato. Existem diversos tipos de gráficos e interpretar corretamente é de grande importância para compreender determinados fenômenos. Neste trabalho abordaremos apenas alguns tipos para responder as questões levantadas:

- Quais são as palavras mais utilizadas?
- Será que existe diferença de tamanho dos textos confiáveis e não confiáveis? E para os títulos, ocorre o mesmo fenômeno?
- Quais são os valores mínimos, máximos, mediana, primeiro e terceiro quartil? Existe outliers? Será que existe diferença entre textos confiáveis e não confiáveis?
- Podemos verificar alguma correlação entre as derivações?
- A forma de escrita do texto tem influência na classificação?

Word Cloud

A ideia central ao plotar o gráfico de nuvem de palavras é encontrar o volume de vezes que determinada palavra se repete dentro do corpus. Este tipo de gráfico se trata de um método heurístico a fim de encontrar respostas viáveis, ainda que imperfeitas, pois mesmo observando as palavras de destaques não avança na compreensão do fato, porém nos guia para alguns questionamentos.[22]

Os 10 gráficos plotados abaixo representam a frequência de palavras segmentadas por: Títulos, Textos, Substantivos, Adjetivos e Verbos.

Distinguimos que as palavras mais relacionadas quando se trata de notícia falsa são ligadas a pronomes pessoais. Na observação notamos a frequência de Hilarry Clinton e Trump. Confirmando fortes evidências relatadas na pesquisa de S. Vosoughi.[4]

Histograma</p>

A construção de histogramas tem caráter preliminar em qualquer estudo e é um importante indicador da distribuição de dados.[23]

Neste estudo utilizamos a frequência absoluta, que é o número que representa a quantidade de dados em uma determinada amostra ou o intervalo de classe específico, indicando a frequência (absoluta) com que uma classe aparece no conjunto de dados.

A seguir, os gráficos plotados representam os histogramas da frequência de palavras e adjetivos encontrados nos títulos e textos. Estes gráficos facilitam na análise de influência na avaliação: Verdade ou Mentira.

Quando uma notícia falsa é transmitida, a profundidade, tamanho, comprimento máximo, ou seja, a estrutura no geral, são menores em relação a uma notícia verdadeira. Os textos jornalísticos têm a preocupação de seguir uma estrutura básica respeitando limites de parágrafos, palavras e comprimento. E os gráficos nos demostra que cada tipo de notícia consiste em um padrão diferente.

Boxplot

O diagrama de caixa é construído utilizando as referências de valores mínimos e máximos, primeiro e terceiro quartil, mediana e os outliers da base de dados.

Diferentemente do histograma que possibilita ter uma melhor visualização das médias e desvio padrão, o diagrama de caixa tem como objetivo estudar as medidas estatísticas e identificar os valores atípicos dentro do conjunto de dados.

As informações extraídas dos gráficos boxplot reafirma a hipótese de que as notícias falsas tendem a ter uma estrutura menor, evidenciando a transmissão da mensagem de forma rápida e com objetivo claro.

Além de conter um número maior de outliers, os números mostraram que 75% dos textos falsos são menores que a mediana das notícias verdadeiras, demonstrando a diferença de estrutura entre os tipos.

Correlação

A matriz de correlação mostra os valores de correlação de Pearson, que medem o grau de relação linear entre cada par de itens ou variáveis. Os valores de correlação podem cair entre -1 e +1.

Com os valores dos atributos gerados na derivação do dataset, observa-se uma análise de correlação a fim de identificar relacionamento entre eles.

Infelizmente, não identificamos um fato relevante que possa nos guiar a um fenômeno inesperado no levantamento das correlações. Apenas evidenciamos que classe gramaticais tem relacionamento mais fortes em textos confiáveis, mas não muito diferentes.

Aprendizado de Máquina

Na etapa de aprendizado de máquina, decidimos utilizar um modelo baseado em redes neurais LSTM desenvolvido com a biblioteca Keras TensorFlow.

A primeira parte do desenvolvimento foi separar as features e o target do conjunto de dados alocando em duas variáveis: train_features e train_targets.

Em seguida, separar os dados de treino e teste para evitar overfitting, ou seja, evitar que o modelo se ajusta somente com os dados da base. O conjunto de dados que contém 15.848 registros, foi dividido em 80% de treinamento e 20% de teste, resumindo, 12.678 registros de treinamento e 3.170 registros de testes.

Para a etapa de separação, foi utilizado a função train_test_split() da biblioteca NTLK. Esta função exige quatro parâmetros: Conjunto de features, Conjuntos de targets, porcentagem da separação e o valor de Randon State).

A terceira parte foi criar um dicionário de tokens com 7.000 palavras. A função Tokenizer() gera este dicionario e com a função fit_on_texts() atualiza o vocabulário com base no dicionário.

Para sequenciar os tokens foi utilizado a função texts_to_sequences(), esta função basicamente pega cada token e a substitui pelo valor inteiro correspondente do dicionário word_index.

Para garantir que todas as sequências tenham o mesmo comprimento, foi utilizado a função pad_sequences() -- passando como parâmetro a variável com os tokens sequenciados e o tamanho do dicionário.

A quarta parte é parametrizar o modelo para executar o treinamento.

O modelo proposto tem duas camadas ocultas LSTM com função de ativação RELU e Sigmoid, propondo treinar os dados em 3 épocas.

Primeiramente foi criado um modelo sequencial. Um modelo sequencial contém uma pilha de camadas que serão adicionadas sequencialmente.

Para a primeira camada da sequência, foi adicionado a camada de incorporação. Esta camada precisa ser definida com o tamanho do dicionário, que é 7.000 palavras, em seguida passa o tamanho do vetor de saída, que é 32, e por último o número de palavras por amostra, no caso 7.000.

Na segunda camada da sequência, foi adicionada uma camada oculta LSTM com 128 nós, junto com outro argumento (return_sequences = True) responsável por permitir a conectividade entre camadas ocultas.

Como nosso modelo é uma rede neural recorrente, geralmente têm o problema de sobre ajuste (overfitting), para diminuir esse problema é necessário a técnica de eliminação (dropout) que ignora aleatoriamente neurônios durante o treinamento. Então, para evitar o problema foi adicionado a camada dropout na terceira camada da sequência.

Para a quarta camada da sequência, foi adicionada mais uma LSTM com 64 nós e em sequência na camada cinco, outro dropout.

Em nosso estudo enfrentamos um problema de classificação binária, precisamos descobrir se determinado texto é verdadeiro ou falso. Para realizar as previsões foi adicionado na sexta camada uma função de ativação RELU com 64 neurônios intermediários (Dense(64, activation='relu')) para lidar com as saídas não-lineares.

E para finalizar foi adicionado na última camada a função de ativação sigmoide (Dense(1, activation='sigmoid')) para classificar binariamente a saída.

As métricas de resultado foram parametrizadas como logloss e acurácia. O objetivo é mensurar a precisão dos acertos. Compilando com o algoritmo de otimização ADAM (loss='binary_crossentropy', optimizer='adam', metrics=['accuracy']).

Foi definido 3 épocas para treinamento, cada uma com lote de 64 classificações para espaçar as atualizações de pesos (X_train_seq, y_train, nb_epoch=3, batch_size=64, validation_data=(X_test_seq,y_test)).

Desenvolvimento

Análise Explorátoria Detalhada

Bibliotecas e preparação do ambiente

In [39]:
```
#!pip install names

import numpy as np
import pandas as pd
import csv
import random
import pandas_profiling #conda install -c conda-forge pandas-profiling
#import names #pip install names
import nltk
import matplotlib.pyplot as plt
import plotly.offline as py
import plotly.graph_objs as go
import seaborn as sns
import string
import spacy
from spacy import displacy
from nltk import pos_tag
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.tokenize import RegexpTokenizer
from nltk.tokenize import TweetTokenizer
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
from pandas.io.json import json_normalize
from plotly import tools
from tqdm import tqdm


nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Error loading averaged_perceptron_tagger: <urlopen error
[nltk_data]     [Errno -3] Temporary failure in name resolution>
```

Out[39]:
```
False
```

Leitura do Dataset

In [40]:
```
# Train Dataset
PATH_DATA_TRAIN = '../input/fakenews/train.csv'

## arquivo disponivel em: https://drive.google.com/drive/folders/1LqNzxY8l0EgznlCD-g873VD-ys1BRN2-?fbclid=IwAR3PPLu4hgNdKfQDJzrLGwV6L42Vm3xBrrcquuCOR4ySS97bVvU46JDaR2s

train_data = pd.read_csv(PATH_DATA_TRAIN)
train_data = train_data.sample(frac = 1) # Randomly Smaple data, ratio is 100%
train_data.head()
```

Out[40]:

	id	title	author	text	label
17858	17858	Re: America Is The Loneliest Country In The Wo...	Celine	America Is The Loneliest Country In The Worl...	1
151	151	Students At Black College Just Got Beaten And ...	Colin Taylor	Comments \nLast night, Louisiana Senate candid...	1
1236	1236	Rigged Election: Hillary & Trump Caught Partyi...	Editor	By Covert Geopolitics\nWe have been very, very...	1
5386	5386	Why So Few Whistleblowers? A Former CIA Agent'...	NaN	Why So Few Whistleblowers? A Former CIA Agent'...	1
8485	8485	NaN	Irishcommander	Too bad the cops can't shoot them at random. W...	1

Classificação das Features

In [41]:
```
table = [["id","Nominal Qualitativo"],["title","Nominal Qualitativo"],
         ["author","Nominal Qualitativo"],["text","Nominal Qualitativo"],
         ["label","Quantitativo Discreto"]]

filing = pd.DataFrame(table, columns=["Variável", "Classificação"])
filing
```

Out[41]:

	Variável	Classificação
0	id	Nominal Qualitativo
1	title	Nominal Qualitativo
2	author	Nominal Qualitativo
3	text	Nominal Qualitativo
4	label	Quantitativo Discreto

Dicionário dos dados

O Dataset "fake news" contém as seguintes informações:

- **ID:** id único da notícia
- **TITLE:** título da notícia
- **AUTHOR:** autor da notícia
- **TEXT:** texto da notícia
- **LABEL:** rótulo que marca se a notícia é potencialmente não confiável
  - 1: não confiável
  - 0: confiável

Análise dos dados

In [42]:
```
train_data_profiling = pd.read_csv(PATH_DATA_TRAIN)
```

In [43]:
```
profile = pandas_profiling.ProfileReport(train_data_profiling)
display(profile)
```

Overview

Dataset info

Number of variables	5
Number of observations	20800
Missing cells	2554 (2.5%)
Duplicate rows	0 (0.0%)
Total size in memory	812.6 KiB
Average record size in memory	40.0 B

Variables types

Numeric	1
Categorical	3
Boolean	1
Date	0
URL	0
Text (Unique)	0
Rejected	0
Unsupported	0

Warnings

`author` has a high cardinality: 4202 distinct values	**Warning**
`author` has 1957 (9.4%) missing values	**Missing**
`text` has a high cardinality: 20387 distinct values	**Warning**
`title` has a high cardinality: 19804 distinct values	**Warning**
`title` has 558 (2.7%) missing values	**Missing**

Variables

Incompletude no Dataset

In [44]:
```
print('Antes do dropna tínhamos {} registros'.format(train_data.shape[0]))
train_data.dropna(inplace=True)
print('Depois do dropna temos {} registro'.format(train_data.shape[0]))
```

```
Antes do dropna tínhamos 20800 registros
Depois do dropna temos 18285 registro
```

Derivação de Features

In [45]:
```
print('Função retorna a POS TAG')
def tag_part_of_speech(text):
    text_splited = text.split(' ')
    text_splited = [''.join(c for c in s if c not in string.punctuation) for s in text_splited]
    text_splited = [s for s in text_splited if s]
    pos_list = pos_tag(text_splited)
    #palavra
    noun = [w[0] for w in pos_list if w[1] in ('NN','NNP','NNPS','NNS')]
    adjective = [w[0] for w in pos_list if w[1] in ('JJ','JJR','JJS')]
    verb = [w[0] for w in pos_list if w[1] in ('VB','VBD','VBG','VBN','VBP','VBZ')]
    #quantidade
    noun_count = len([w for w in pos_list if w[1] in ('NN','NNP','NNPS','NNS')])
    adjective_count = len([w for w in pos_list if w[1] in ('JJ','JJR','JJS')])
    verb_count = len([w for w in pos_list if w[1] in ('VB','VBD','VBG','VBN','VBP','VBZ')])

    return[noun, adjective, verb, noun_count, adjective_count, verb_count]

tokenizer = RegexpTokenizer(r'\w+')
```

```
Função retorna a POS TAG
```

In [46]:
```
print('Função retorna tokens unicos')
def token_distint(df):
    list_words = []
    tokenizer = RegexpTokenizer(r'\w+')
    for item in df:
        for token in tokenizer.tokenize(item.lower()):
            if not token in list_words:
                list_words.append(token)
    return list_words
```

```
Função retorna tokens unicos
```

In [47]:
```
print("Geração de recursos baseados nos titulos")
tokenizer = RegexpTokenizer(r'\w+')
for df in ([train_data]):
    df['title_token'] = df['title'].apply(lambda x : tokenizer.tokenize(x.lower()))
    df['title_token_distint'] = df['title'].apply(lambda x : token_distint(x))
    df['title_comprimento'] = df['title'].apply(lambda x : len(str(x)))
    df['title_num_palavras'] = df['title'].apply(lambda x: len(str(x).split()))
    df['title_num_palavras_unicas'] = df['title'].apply(lambda x: len(set(w for w in str(x).split())))
    df['title_palavras_vs_unico'] = df['title_num_palavras_unicas'] / df['title_num_palavras']
    df['title_substantivos'], df['title_adjetivos'], df['title_verbos'], df['title_contagem_substantivos'], df['title_
contagem_adjetivos'], df['title_contagem_verbos'] = zip(*df['title'].apply(lambda text: tag_part_of_speech(str(text
))))
```

```
Geração de recursos baseados nos titulos
```

In [48]:
```
print("Geração de recursos baseados nos textos")
tokenizer = RegexpTokenizer(r'\w+')
for df in ([train_data]):
    df['text_token'] = df['text'].apply(lambda x : tokenizer.tokenize(x.lower()))
    df['text_token_distint'] = df['text'].apply(lambda x : token_distint(x))
    df['text_comprimento'] = df['text'].apply(lambda x : len(str(x)))
    df['text_num_palavras'] = df['text'].apply(lambda x: len(str(x).split()))
    df['text_num_palavras_unicas'] = df['text'].apply(lambda x: len(set(w for w in str(x).split())))
    df['text_palavras_vs_unico'] = df['text_num_palavras_unicas'] / df['text_num_palavras']
    df['text_substantivos'], df['text_adjetivos'], df['text_verbos'], df['text_contagem_substantivos'], df['text_conta
gem_adjetivos'], df['text_contagem_verbos'] = zip(*df['text'].apply(lambda text: tag_part_of_speech(str(text))))
```

```
Geração de recursos baseados nos textos
```

Desbalanceamento de Classes e Random Shuffle

In [49]:
```
train_data.label.describe()
```

Out[49]:
```
count    18285.000000
mean         0.433361
std          0.495553
min          0.000000
25%          0.000000
50%          0.000000
75%          1.000000
max          1.000000
Name: label, dtype: float64
```

In [50]:
```
unreliable = train_data[train_data['label'] == 1]
print('Não confiável: ', len(unreliable))

reliable = train_data[train_data['label'] == 0]
print('Confiável: ', len(reliable))

print('Desequilibrio nos dados de {} registros confiáveis'.format(len(reliable) - len(unreliable)))
```

```
Não confiável:  7924
Confiável:  10361
Desequilibrio nos dados de 2437 registros confiáveis
```

In [51]:
```
print('aplicando Random Shuffle')
mean = min(len(unreliable), len(reliable))

un_data = unreliable.sample(n = mean)
print('Não confiável: ', len(un_data))
r_data = reliable.sample(n = mean)
print('Confiável: ', len(r_data))

train_data = pd.concat([un_data, r_data])
```

```
aplicando Random Shuffle
Não confiável:  7924
Confiável:  7924
```

Exportando Dataset

In [52]:
```
train_data.to_csv ('train_refatorado_final.csv', index = None, header=True)
```

Análise Gráfica

Word Cloud

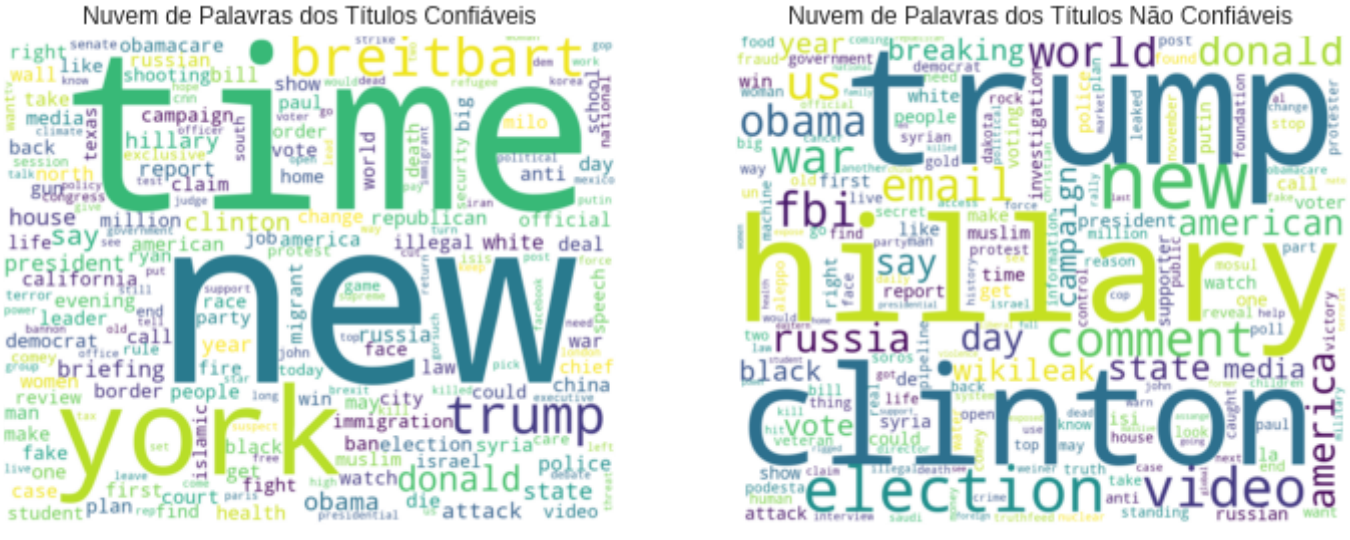
In [53]:
```
print('Função para listar as palavras')
def list_words(data):
    l_words = []
    tokenizer = RegexpTokenizer(r'\w+')
    for tokens in data:
        for token in tokens:
            l_words.append(token)
    return l_words
```

```
Função para listar as palavras
```

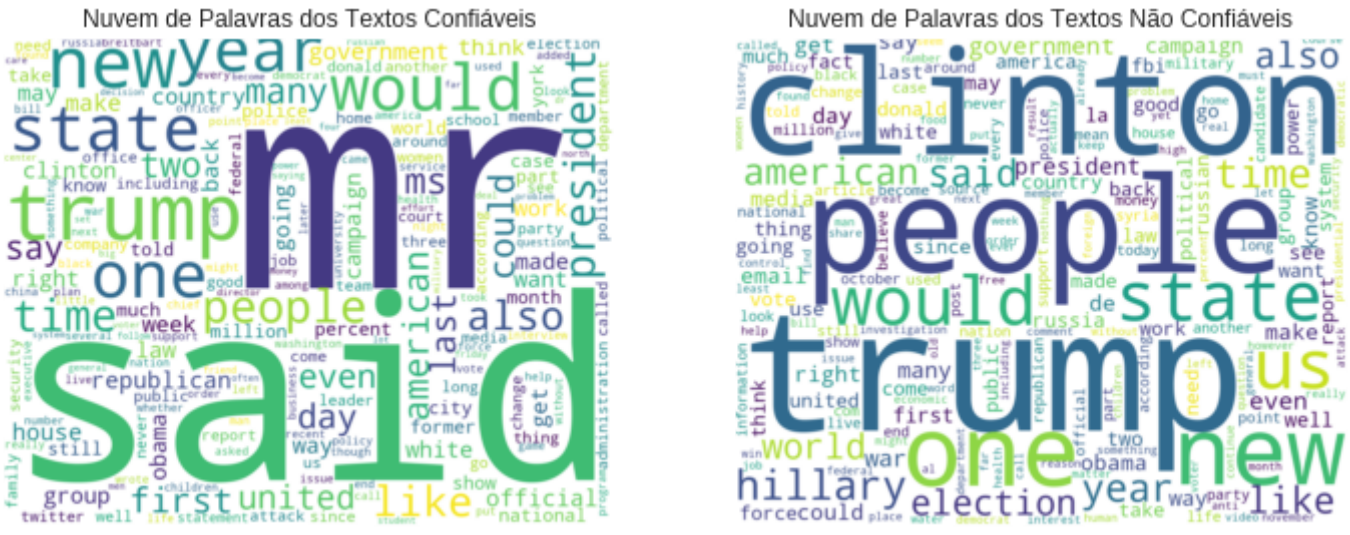
In [54]:
```
print('Função que plota o gráfico de Words Cloud')
def words_cloud(confiavel, nao_confiavel, title):


    ## Palavras dentro do texto
    words = list_words(confiavel)
    text_confiavel = " ".join(review for review in words)

    words = list_words(nao_confiavel)
    text_nao_confiavel = " ".join(review for review in words)

    ## Carregando a lista de stopwords
    stopwords = nltk.corpus.stopwords.words('english')

    # Generate a word cloud image
    wordcloud_confiavel = WordCloud(stopwords=stopwords, background_color="white", width=1000, height=800, margin=0, c
ollocations=False).generate(text_confiavel)
    wordcloud_nao_confiavel = WordCloud(stopwords=stopwords, background_color="white", width=1000, height=800, margin=
0, collocations=False).generate(text_nao_confiavel)

    ## plotagem
    plt.figure(figsize=(12,8))

    plt.subplot(121)
    plt.imshow(wordcloud_confiavel, interpolation='bilinear')
    plt.axis('off')
    plt.margins(x=0,y=0)
    plt.title(title + ' Confiáveis')
    plt.legend()

    plt.subplot(122)
    plt.imshow(wordcloud_nao_confiavel, interpolation='bilinear')
    plt.axis('off')
    plt.margins(x=0,y=0)
    plt.title(title + ' Não Confiáveis')
    plt.legend()

    plt.show()
```

```
Função que plota o gráfico de Words Cloud
```

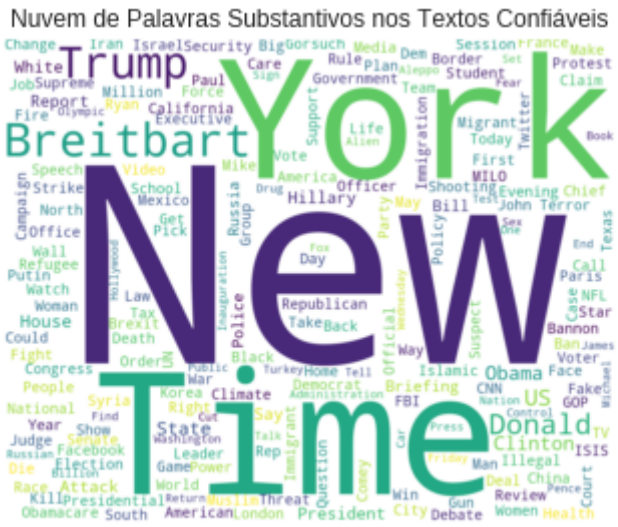
In [55]:
```
words_cloud(reliable['title_token'], unreliable['title_token'], 'Nuvem de Palavras dos Títulos')
```



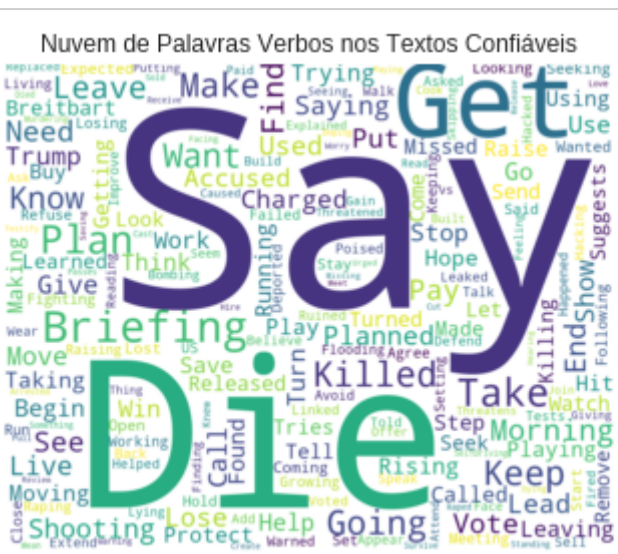
In [56]:
```
words_cloud(reliable['text_token'], unreliable['text_token'], 'Nuvem de Palavras dos Textos')
```

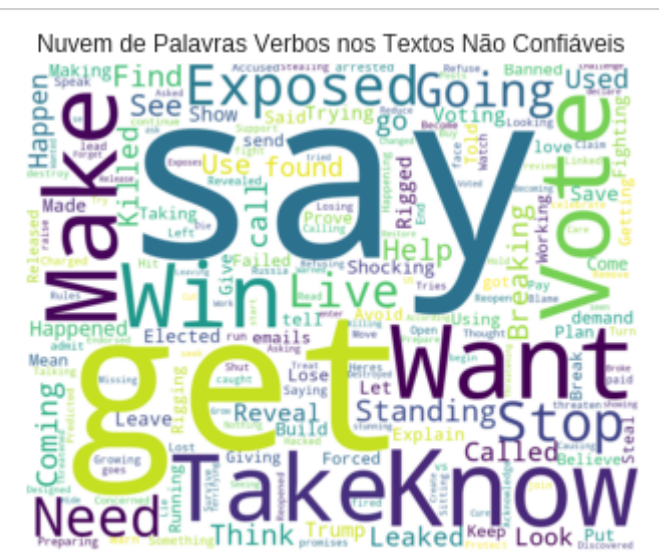


In [57]:
```
words_cloud(reliable['title_substantivos'], unreliable['title_substantivos'], 'Nuvem de Palavras Substantivos nos Textos')
```



In [58]:
```
words_cloud(reliable['title_verbos'], unreliable['title_verbos'], 'Nuvem de Palavras Verbos nos Textos')
```



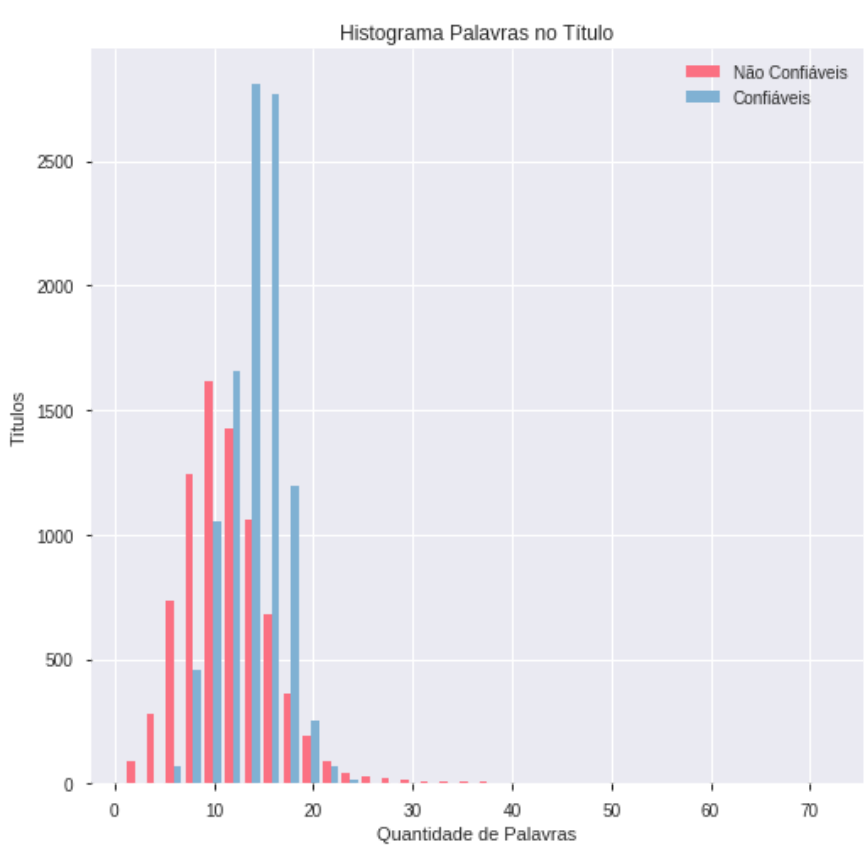


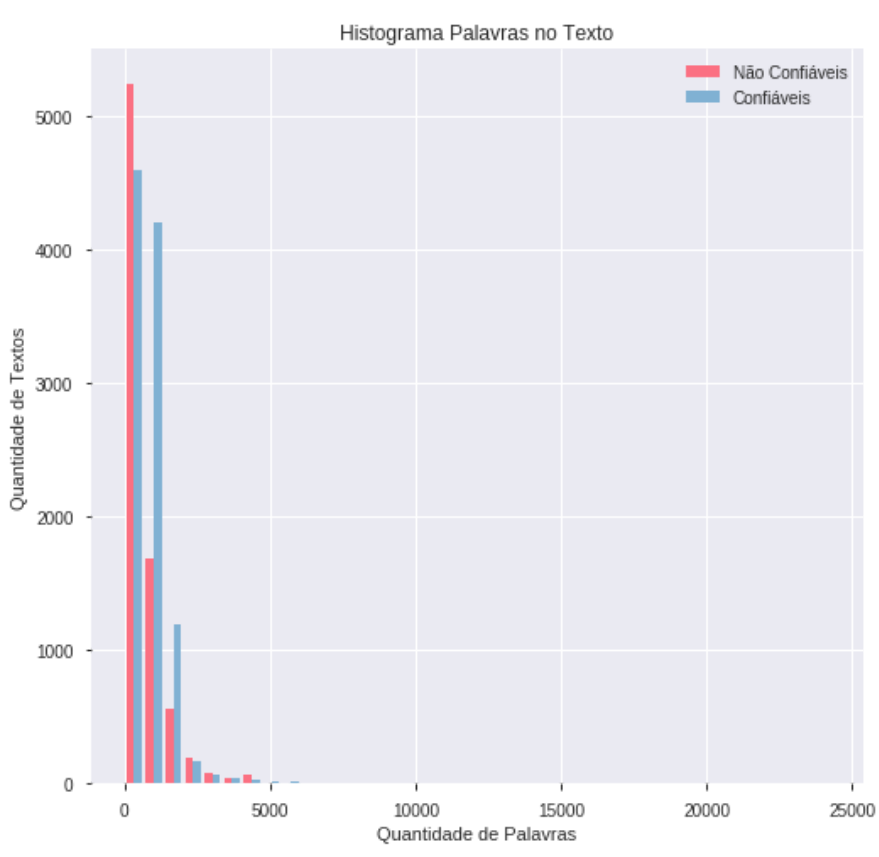
In [59]:
```
words_cloud(reliable['title_adjetivos'], unreliable['title_adjetivos'], 'Nuvem de Palavras Adjetivos nos Textos')
```

Histograma da Quantidade de Palavras

In [60]:
```
labels = ['Não Confiáveis','Confiáveis']
colors = ['#fb7082','#80b1d3']

plt.figure(figsize=(18,8))
plt.subplot(121)
plt.hist([unreliable.title_num_palavras,reliable.title_num_palavras], bins=int(180/5), normed=False, color = colors, l
abel=labels)
plt.xlabel('Quantidade de Palavras')
plt.ylabel('Títulos')
plt.title('Histograma Palavras no Título')
plt.legend()


plt.subplot(122)
plt.hist([unreliable.text_num_palavras,reliable.text_num_palavras], bins=int(180/5), normed=False, color = colors, lab
el=labels)
plt.xlabel('Quantidade de Palavras')
plt.ylabel('Quantidade de Textos')
plt.title('Histograma Palavras no Texto')
plt.legend()

plt.show()
```

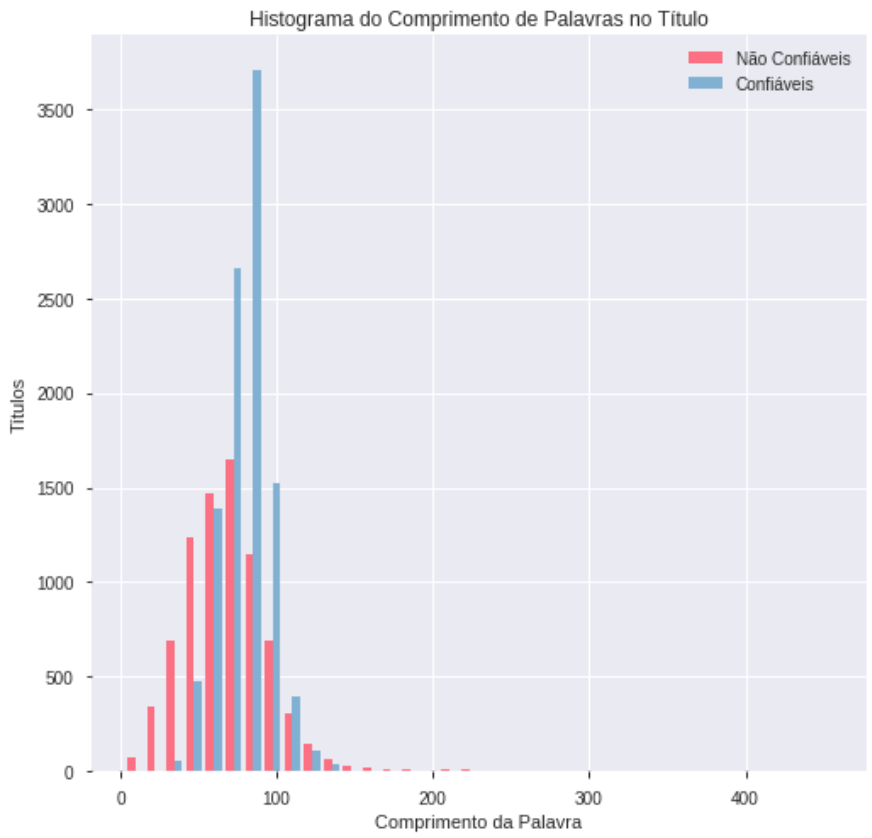
```
/opt/conda/lib/python3.6/site-packages/matplotlib/axes/_axes.py:6521: MatplotlibDeprecationWarning:


The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be removed in 3.1. Use 'density' instead.
```

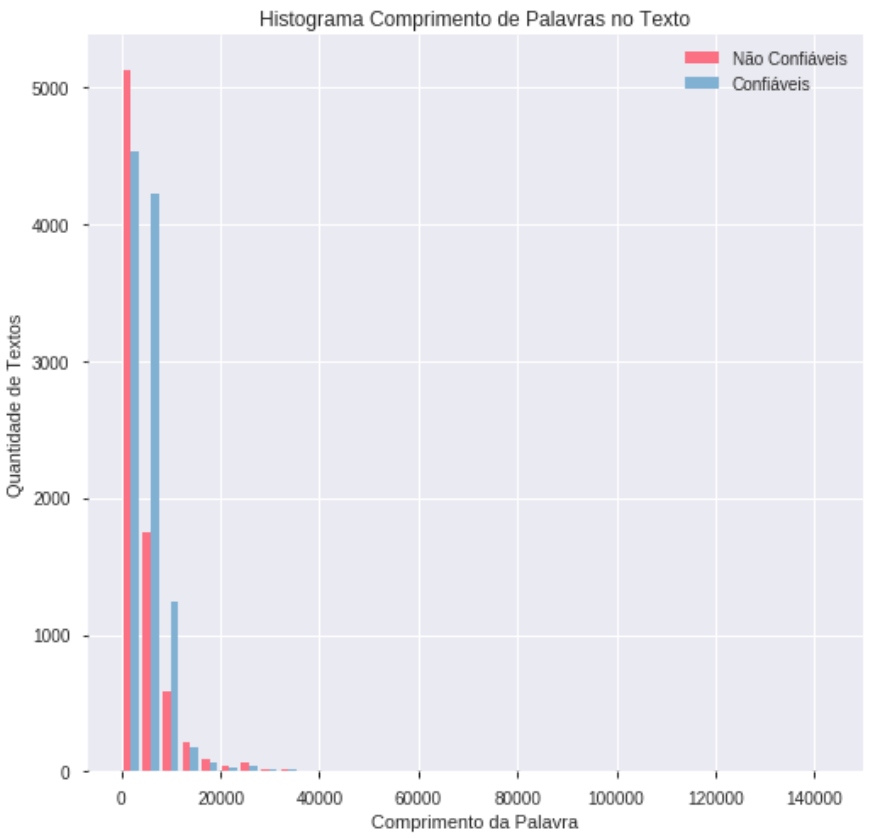




Histograma do Comprimento de Palavras

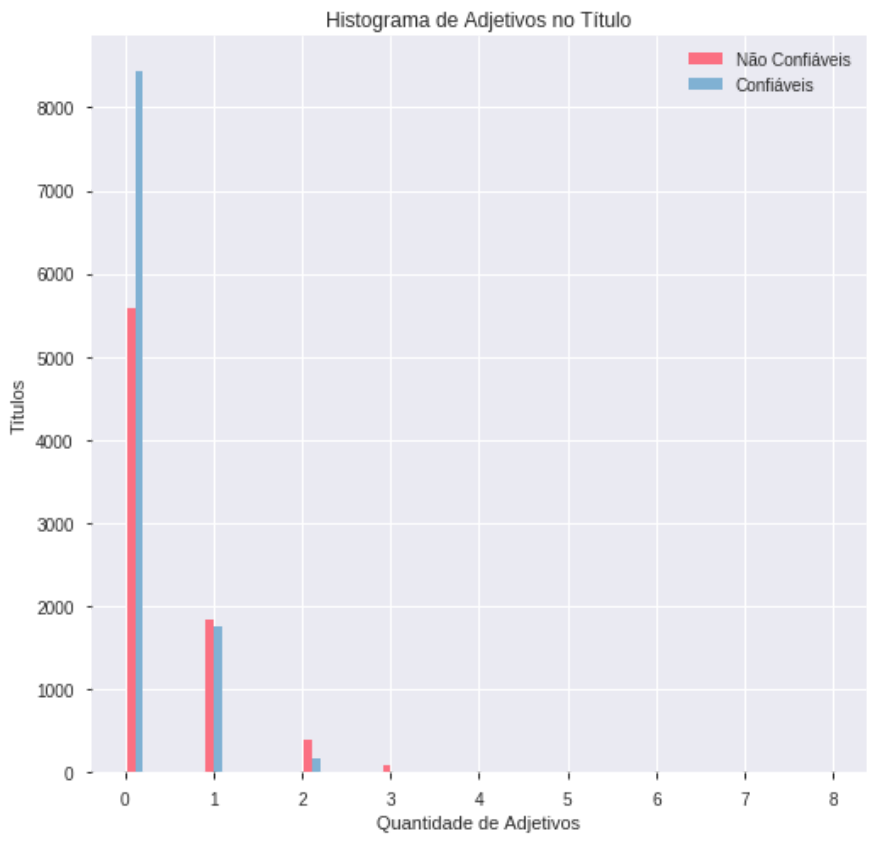
In [61]:
```
labels = ['Não Confiáveis','Confiáveis']
colors = ['#fb7082','#80b1d3']

plt.figure(figsize=(18,8))
plt.subplot(121)
plt.hist([unreliable.title_comprimento,reliable.title_comprimento], bins=int(180/5), normed=False, color = colors, lab
el=labels)
plt.xlabel('Comprimento da Palavra')
plt.ylabel('Títulos')
plt.title('Histograma do Comprimento de Palavras no Título')
plt.legend()

plt.subplot(122)
plt.hist([unreliable.text_comprimento,reliable.text_comprimento], bins=int(180/5), normed=False, color = colors, label
=labels)
plt.xlabel('Comprimento da Palavra')
plt.ylabel('Quantidade de Textos')
plt.title('Histograma Comprimento de Palavras no Texto')
plt.legend()

plt.show()
```

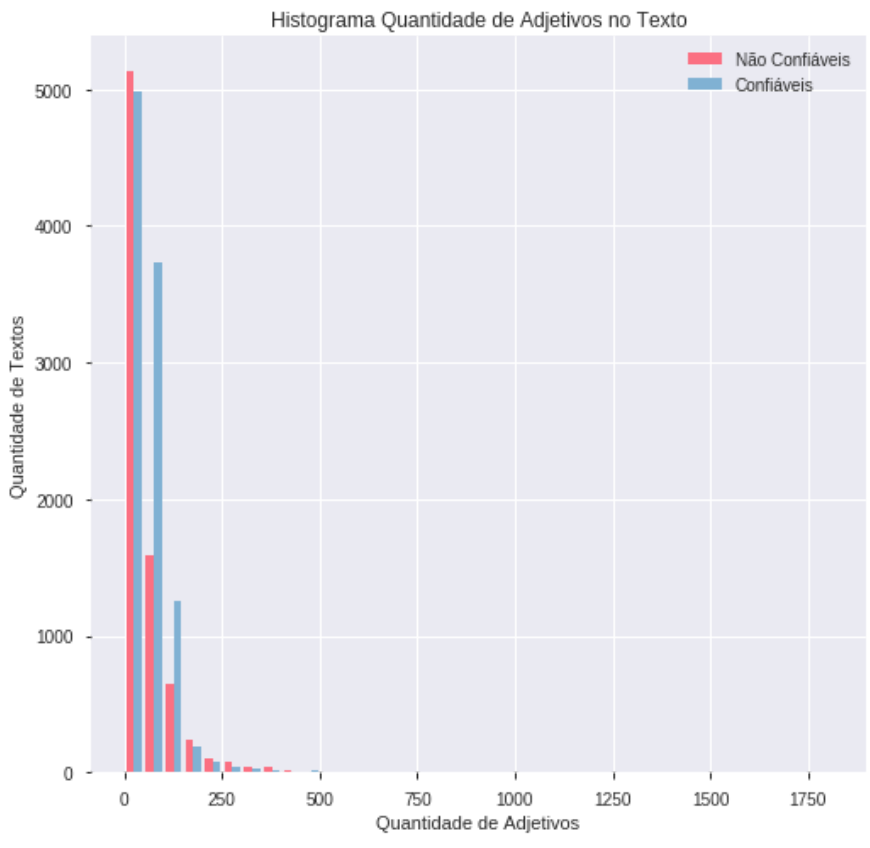




Histograma da Quantidade de Adjetivos

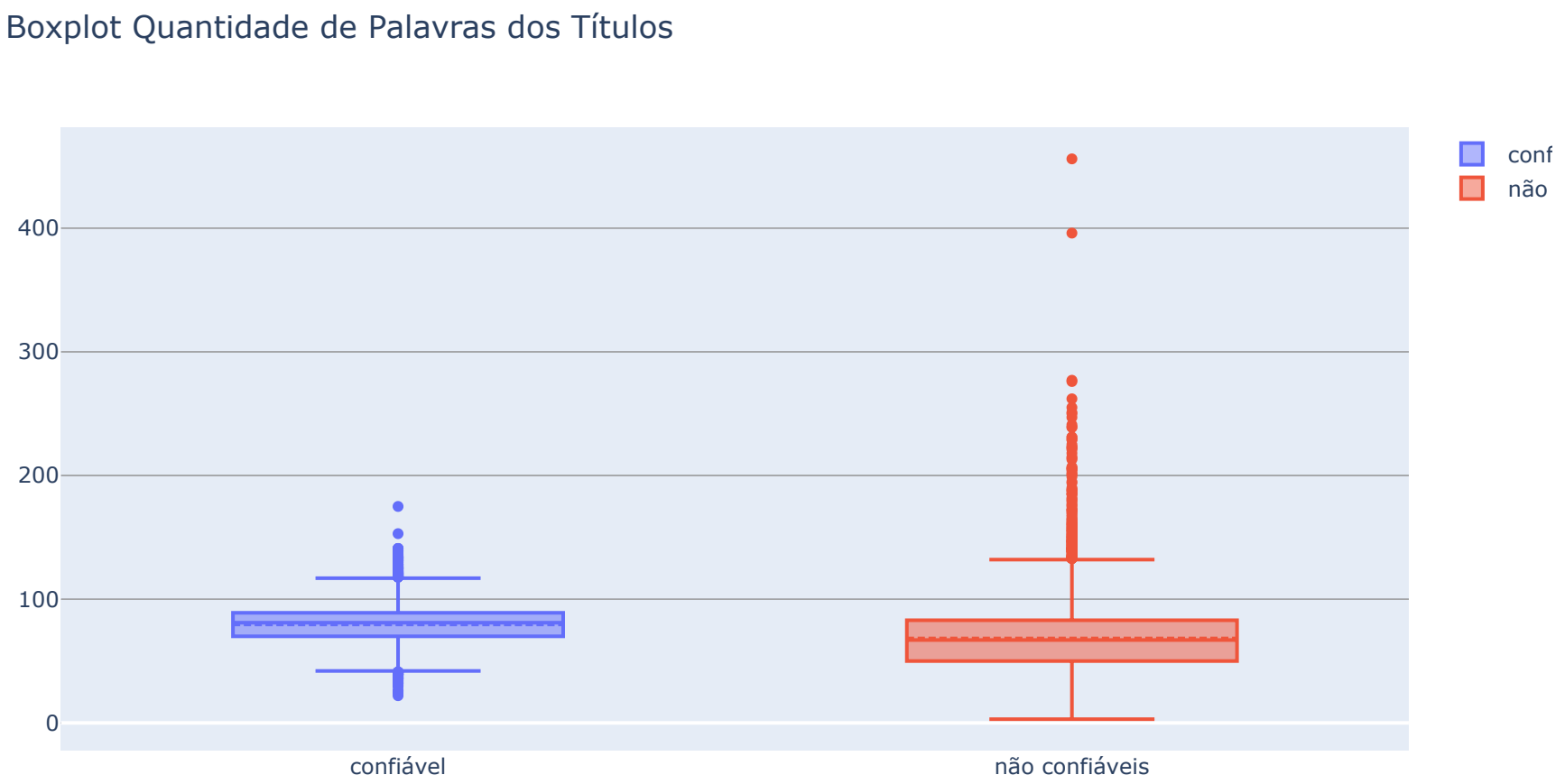
In [62]:
```
labels = ['Não Confiáveis','Confiáveis']
colors = ['#fb7082','#80b1d3']

plt.figure(figsize=(18,8))
plt.subplot(121)
plt.hist([unreliable.title_contagem_adjetivos,reliable.title_contagem_adjetivos], bins=int(180/5), normed=False, color
= colors, label=labels)
plt.xlabel('Quantidade de Adjetivos')
plt.ylabel('Títulos')
plt.title('Histograma de Adjetivos no Título')
plt.legend()

plt.subplot(122)
plt.hist([unreliable.text_contagem_adjetivos,reliable.text_contagem_adjetivos], bins=int(180/5), normed=False, color =
colors, label=labels)
plt.xlabel('Quantidade de Adjetivos')
plt.ylabel('Quantidade de Textos')
plt.title('Histograma Quantidade de Adjetivos no Texto')
plt.legend()

plt.show()
```



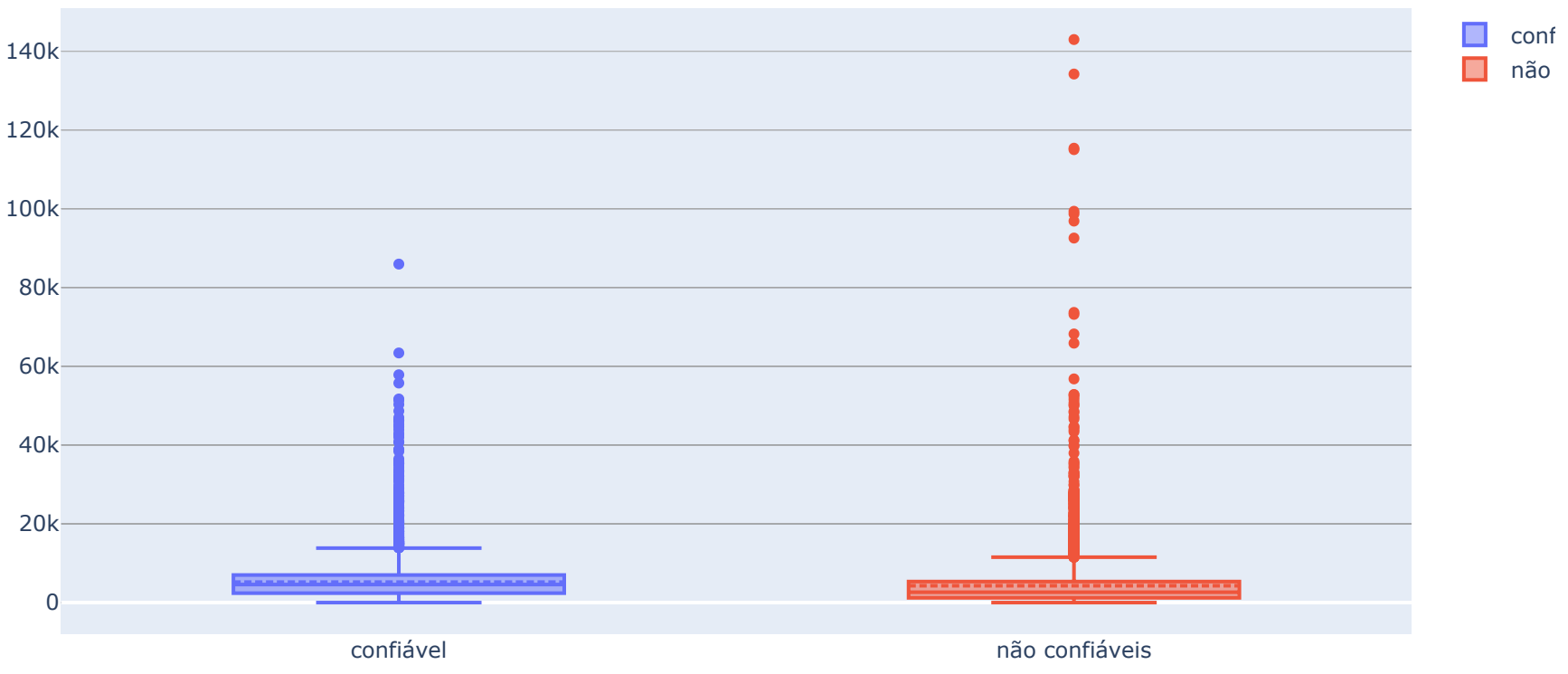


Boxplot Quantidade de Palavras

In [63]:
```
confiavel = go.Box(y=reliable.title_comprimento, name = 'confiável', boxmean=True)
nao_confiavel = go.Box(y=unreliable.title_comprimento, name = 'não confiáveis', boxmean=True)
data = [confiavel, nao_confiavel]
layout = go.Layout(title = "Boxplot Quantidade de Palavras dos Títulos")
fig = go.Figure(data=data,layout=layout)
py.iplot(fig)
```



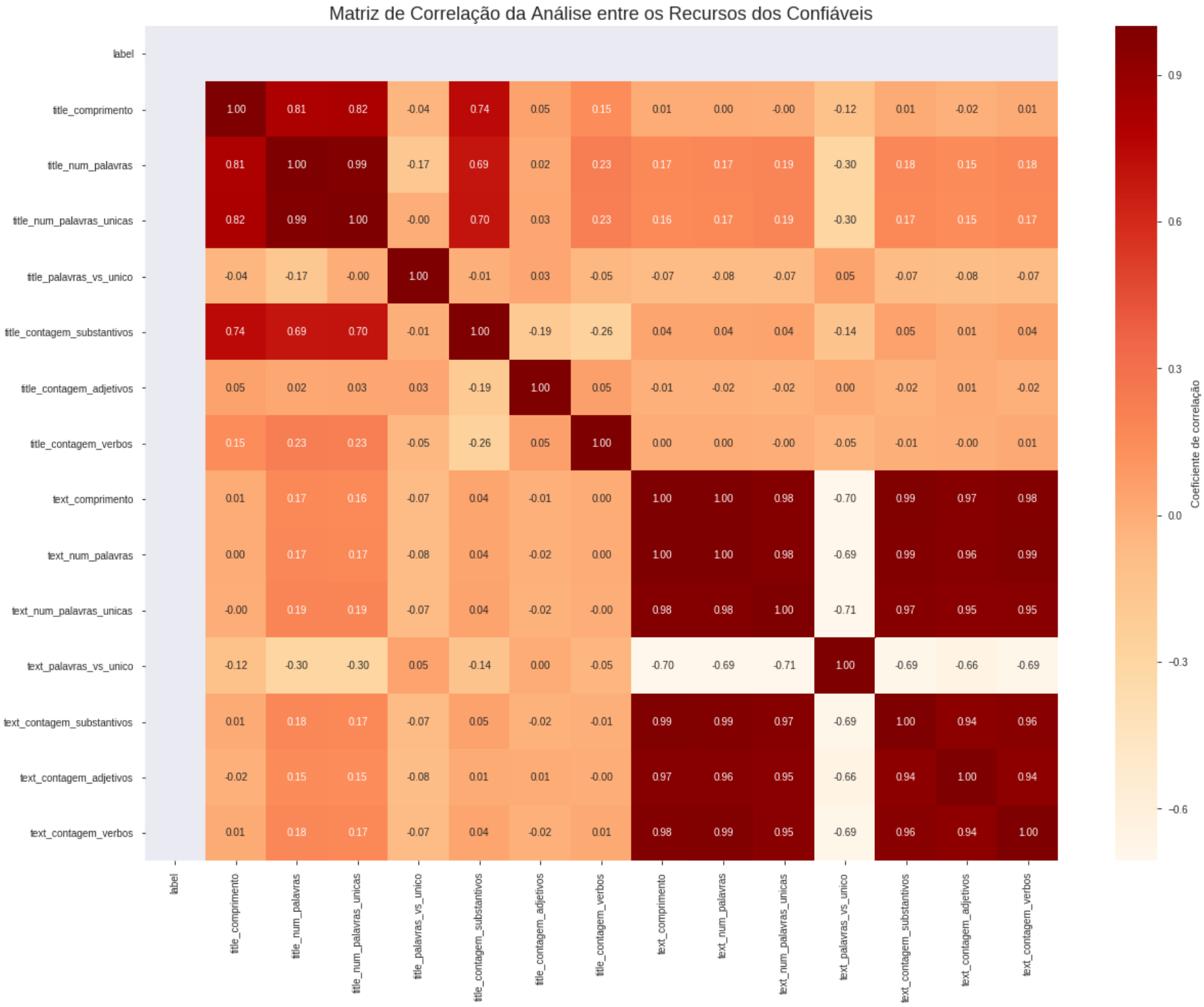
In [64]:
```
confiavel = go.Box(y=reliable.text_comprimento, name = 'confiável', boxmean=True)
nao_confiavel = go.Box(y=unreliable.text_comprimento, name = 'não confiáveis', boxmean=True)
data = [confiavel, nao_confiavel]
layout = go.Layout(title = "Boxplot da Quantidade de Palavras dos Textos")
fig = go.Figure(data=data,layout=layout)
py.iplot(fig)
```



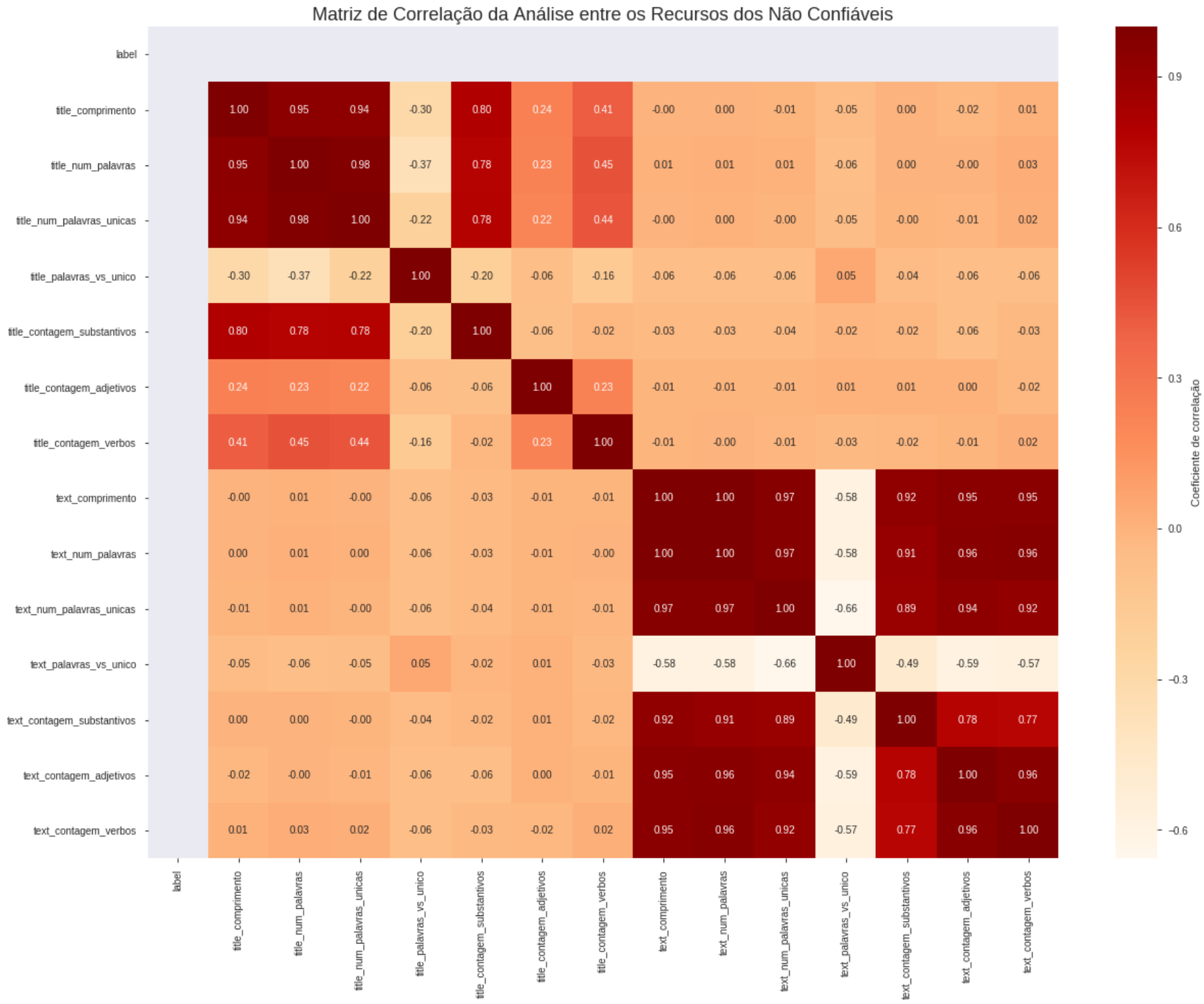
Matriz de correlação

In [65]:
```
py.init_notebook_mode(connected=True)
pd.options.mode.chained_assignment = None
np.random.seed(13)

color = sns.color_palette()
%matplotlib inline
```

In [66]:
```
corr_reliable = reliable.drop(columns=['id', 'title', 'author', 'text', 'title_token','text_token','title_substantivos','title_adjetivos','title_verbos','text_substantivos',
                                       'text_adjetivos','text_verbos'])

f, ax = plt.subplots(figsize= [20,15])
sns.heatmap(corr_reliable.corr(), annot=True, fmt=".2f", ax=ax,
            cbar_kws={'label': 'Coeficiente de correlação'}, cmap='OrRd')
ax.set_title("Matriz de Correlação da Análise entre os Recursos dos Confiáveis", fontsize=18)
plt.show()
```



In [67]:
```
corr_unreliable = unreliable.drop(columns=['id', 'title', 'author', 'text', 'title_token','text_token','title_substantivos','title_adjetivos','title_verbos','text_substantivos',
                                'text_adjetivos','text_verbos'])

f, ax = plt.subplots(figsize= [20,15])
sns.heatmap(corr_unreliable.corr(), annot=True, fmt=".2f", ax=ax,
            cbar_kws={'label': 'Coeficiente de correlação'}, cmap='OrRd')
ax.set_title("Matriz de Correlação da Análise entre os Recursos dos Não Confiáveis", fontsize=18)
plt.show()
```



In [68]:
```
train_data.columns
```

Out[68]:
```
Index(['id', 'title', 'author', 'text', 'label', 'title_token',
       'title_token_distint', 'title_comprimento', 'title_num_palavras',
       'title_num_palavras_unicas', 'title_palavras_vs_unico',
       'title_substantivos', 'title_adjetivos', 'title_verbos',
       'title_contagem_substantivos', 'title_contagem_adjetivos',
       'title_contagem_verbos', 'text_token', 'text_token_distint',
       'text_comprimento', 'text_num_palavras', 'text_num_palavras_unicas',
       'text_palavras_vs_unico', 'text_substantivos', 'text_adjetivos',
       'text_verbos', 'text_contagem_substantivos', 'text_contagem_adjetivos',
       'text_contagem_verbos'],
      dtype='object')
```

Aprendizado de Máquina

Preparação do Ambiente

In [69]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
```

In [70]:
```
path_train = '../input/train-refatorado-new/train_refatorado.csv'

# arquivo disponivel em: https://drive.google.com/drive/folders/1LqNzxY8l0EgznlCD-g873VD-ys1BRN2-?fbclid=IwAR3PPLu4hgNdKfQDJzrLGwV6L42Vm3xBrrcquuCOR4ySS97bVvU46JDaR2s

train_data = pd.read_csv(path_train, encoding='utf-8')
```

Classificação das Variáveis

In [71]:
```
table = [['id','Norminal Qualitativo'],
         ['title','Norminal Qualitativo'],
         ['author','Norminal Qualitativo'],
         ['text','Norminal Qualitativo'],
         ['label','Quantitativo Discreto'],
         ['title_token','Norminal Qualitativo'],
         ['title_token_distint','Norminal Qualitativo'],
         ['title_comprimento',"Quantitativo Discreto"],
         ['title_num_palavras',"Quantitativo Discreto"],
         ['title_num_palavras_unicas',"Quantitativo Discreto"],
         ['title_palavras_vs_unico','Quantitativo Discreto'],
         ['title_substantivos','Norminal Qualitativo'],
         ['title_adjetivos','Norminal Qualitativo'],
         ['title_verbos','Norminal Qualitativo'],
         ['title_contagem_substantivos',"Quantitativo Discreto"],
         ['title_contagem_adjetivos',"Quantitativo Discreto"],
         ['title_contagem_verbos',"Quantitativo Discreto"],
         ['text_token','Norminal Qualitativo'],
         ['text_token_distint','Norminal Qualitativo'],
         ['text_comprimento',"Quantitativo Discreto"],
         ['text_num_palavras',"Quantitativo Discreto"],
         ['text_num_palavras_unicas',"Quantitativo Discreto"],
         ['text_palavras_vs_unico','Quantitativo Discreto'],
         ['text_substantivos','Norminal Qualitativo'],
         ['text_adjetivos','Norminal Qualitativo'],
         ['text_verbos','Norminal Qualitativo'],
         ['text_contagem_substantivos',"Quantitativo Discreto"],
         ['text_contagem_adjetivos',"Quantitativo Discreto"],
         ['text_contagem_verbos',"Quantitativo Discreto"]]

filing = pd.DataFrame(table, columns=["Variável", "Classificação"])
filing
```

Out[71]:

	Variável	Classificação
0	id	Norminal Qualitativo
1	title	Norminal Qualitativo
2	author	Norminal Qualitativo
3	text	Norminal Qualitativo
4	label	Quantitativo Discreto
5	title_token	Norminal Qualitativo
6	title_token_distint	Norminal Qualitativo
7	title_comprimento	Quantitativo Discreto
8	title_num_palavras	Quantitativo Discreto
9	title_num_palavras_unicas	Quantitativo Discreto
10	title_palavras_vs_unico	Quantitativo Discreto
11	title_substantivos	Norminal Qualitativo
12	title_adjetivos	Norminal Qualitativo
13	title_verbos	Norminal Qualitativo
14	title_contagem_substantivos	Quantitativo Discreto
15	title_contagem_adjetivos	Quantitativo Discreto
16	title_contagem_verbos	Quantitativo Discreto
17	text_token	Norminal Qualitativo
18	text_token_distint	Norminal Qualitativo
19	text_comprimento	Quantitativo Discreto
20	text_num_palavras	Quantitativo Discreto
21	text_num_palavras_unicas	Quantitativo Discreto
22	text_palavras_vs_unico	Quantitativo Discreto
23	text_substantivos	Norminal Qualitativo
24	text_adjetivos	Norminal Qualitativo
25	text_verbos	Norminal Qualitativo
26	text_contagem_substantivos	Quantitativo Discreto
27	text_contagem_adjetivos	Quantitativo Discreto
28	text_contagem_verbos	Quantitativo Discreto

Dicionário dos Dados

- **id:** ID exclusivo para um artigo de notícias
- **title:** título de uma notícia
- **author:** autor da notícia
- **text:** texto do artigo
- **label:** rótulo que marca se a notícia é potencialmente não confiável
  - 1: não confiável
  - 0: confiável
- **title_token:** palavras dentro do título
- **title_token_distint:** palavras distintas dentro do título
- **title_comprimento:** comprimento do título
- **title_num_palavras:** número de palavras dentro do título
- **title_num_palavras_unicas:** número de palavras distintas dentro do título
- **title_palavras_vs_unico:** número de palavras / palavras distintas dentro do título
- **title_substantivos:** substantivos dentro do título
- **title_adjetivos:** adjetivos dentro do título
- **title_verbos:** verbos dentro do título
- **title_contagem_substantivos:** número de substantivos dentro do título
- **title_contagem_adjetivos:** número de adjetivos dentro do título
- **title_contagem_verbos:** número de verbos dentro do título
- **text_token:** palavras dentro do texto
- **text_token_distint:** palavras distintas dentro do texto
- **text_comprimento:** comprimento do texto
- **text_num_palavras:** número de palavras dentro do texto
- **text_num_palavras_unicas:** número de palavras distintas dentro do texto
- **text_palavras_vs_unico:** número de palavras / palavras distintas dentro do texto
- **text_substantivos:** substantivos dentro do texto
- **text_adjetivos:** adjetivos dentro do texto
- **text_verbos:** verbos dentro do texto
- **text_contagem_substantivos:** número de substantivos dentro do texto
- **text_contagem_adjetivos:** número de adjetivos dentro do texto
- **text_contagem_verbos:** número de verbos dentro do texto

Perfil do Conjunto de Dados

In [72]:
```
train_data.head()
```

Out[72]:

	id	title	author	text	label	title_token	title_comprimento	title_num_exclamação	title_num_questao	title_num_
0	16625	Donald Trump Calls A Black Supporter A Paid Th...	Sarah Jones	By Sarah Jones on Fri, Oct 28th, 2016 at 8:34 ...	1	['donald', 'trump', 'calls', 'a', 'black', 'su...	79	0	0	
1	1877	SIXTY KILLED IN NEW US-SAUDI WAR CRIME IN YEMEN	Iron Sheik	Home › WORLD NEWS › SIXTY KILLED IN NEW US-SAU...	1	['sixty', 'killed', 'in', 'new', 'us', 'saudi'...	47	0	0	
2	18296	How Electoral College Cheats Democracy	Consortiumnews.com	How Electoral College Cheats Democracy Novembe...	1	['how', 'electoral', 'college', 'cheats', 'dem...	38	0	0	
3	6031	Why Obama Will Win	Jared Taylor	Why Obama Will Win Jared Taylor, American Rena...	1	['why', 'obama', 'will', 'win']	18	0	0	
4	6033	Obamacare Architect LAUGHS About Skyrocketing ...	admin	Daily Caller October 27, 2016 \nEzekiel Emanue...	1	['obamacare', 'architect', 'laughs', 'about', ...	62	0	0	

5 rows × 71 columns

In [73]:
```
train_data['title_author_text'] = train_data['title'] + ' ' + train_data['author'] + ' ' + train_data['text']
train_data['len_title_author_text'] = [len(str(x)) for x in train_data['title_author_text']]
```

In [74]:
```
detail = train_data['len_title_author_text'].describe()
print(detail)
```

```
count     15848.000000
mean       4829.362128
std        5335.062015
min          23.000000
25%        1836.000000
50%        3658.000000
75%        6550.000000
max      143053.000000
Name: len_title_author_text, dtype: float64
```

Aprendizado de Máquina com Deep Learning

In [75]:
```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from keras.layers import Dropout
from keras.layers.embeddings import Embedding
from keras.preprocessing import sequence
from keras.preprocessing.text import Tokenizer

# Corrigir seed aleatório para qualidade de reprodução
# Este método é chamado quando o RandomState é inicializado.
np.random.seed(7)
```

In [76]:
```
## Separando feature e target
train_features = train_data['title_author_text']
train_targets = train_data['label']

## Realizando split do treino e teste
X_train, X_test, y_train, y_test = train_test_split(train_features, train_targets, test_size=0.2, random_state=42)

print('Dados de Treino - Feature: {}'.format(len(X_train)))
print('Dados de Treino - Label: {}'.format(len(y_train)))
print(' ')
print('Dados de Teste - Feature: {}'.format(len(X_test)))
print('Dados de Teste - Label: {}'.format(len(y_test)))
```

```
Dados de Treino - Feature: 12678
Dados de Treino - Label: 12678

Dados de Teste - Feature: 3170
Dados de Teste - Label: 3170
```

In [77]:
```
## Tokenização
num_token = 7000
token = Tokenizer(num_words = num_token, filters = '!"#$%&()*+,-./:;<=>?@[\\]^_`{|}~\t\n')
token.fit_on_texts(X_train.astype(str))
```

In [78]:
```
## Tamanho da sequencia de palavras
max_review_length = 7000

x_train_token = token.texts_to_sequences(X_train.astype(str))
x_test_token = token.texts_to_sequences(X_test.astype(str))

X_train_seq = sequence.pad_sequences(x_train_token, maxlen=max_review_length)
X_test_seq = sequence.pad_sequences(x_test_token, maxlen=max_review_length)
```

In [79]:
```
## Parametros Keras
embedding_vector_length = 32
dropout = 0.3
batch_size = 64
epochs = 3

## Modelo LSTM
model_lstm = Sequential()
model_lstm.add(Embedding(input_dim=num_token, output_dim=embedding_vector_length, input_length=max_review_length))
model_lstm.add(LSTM(128,return_sequences=True))
model_lstm.add(Dropout(dropout))
model_lstm.add(LSTM(64))
model_lstm.add(Dropout(dropout))
model_lstm.add(Dense(64, activation='relu'))
model_lstm.add(Dropout(dropout))
model_lstm.add(Dense(1, activation='sigmoid'))
model_lstm.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'] )
print(model_lstm.summary())

## Treinando
history = model_lstm.fit( X_train_seq, y_train,
                         nb_epoch=epochs, batch_size=batch_size,
                         validation_data=(X_test_seq,y_test))
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_2 (Embedding)      (None, 7000, 32)          224000
_________________________________________________________________
lstm_3 (LSTM)                (None, 7000, 128)         82432
_________________________________________________________________
dropout_4 (Dropout)          (None, 7000, 128)         0
_________________________________________________________________
lstm_4 (LSTM)                (None, 64)                49408
_________________________________________________________________
dropout_5 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_3 (Dense)              (None, 64)                4160
_________________________________________________________________
dropout_6 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 65
=================================================================
Total params: 360,065
Trainable params: 360,065
Non-trainable params: 0
_________________________________________________________________
None

/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:23: UserWarning:

The `nb_epoch` argument in `fit` has been renamed `epochs`.


Train on 12678 samples, validate on 3170 samples
Epoch 1/3
12678/12678 [==============================] - 4293s 339ms/step - loss: 0.4125 - accuracy: 0.8140 - val_loss: 0.2678
- val_accuracy: 0.8871
Epoch 2/3
12678/12678 [==============================] - 4494s 355ms/step - loss: 0.1863 - accuracy: 0.9322 - val_loss: 0.2181
- val_accuracy: 0.9205
Epoch 3/3
12678/12678 [==============================] - 4590s 362ms/step - loss: 0.2590 - accuracy: 0.9024 - val_loss: 0.2178
- val_accuracy: 0.9341
```

In [80]:
```
#Serializar modelo para JSON
model_json = model_lstm.to_json()
with open("model_lstm.json", "w") as json_file:
    json_file.write(model_json)
```

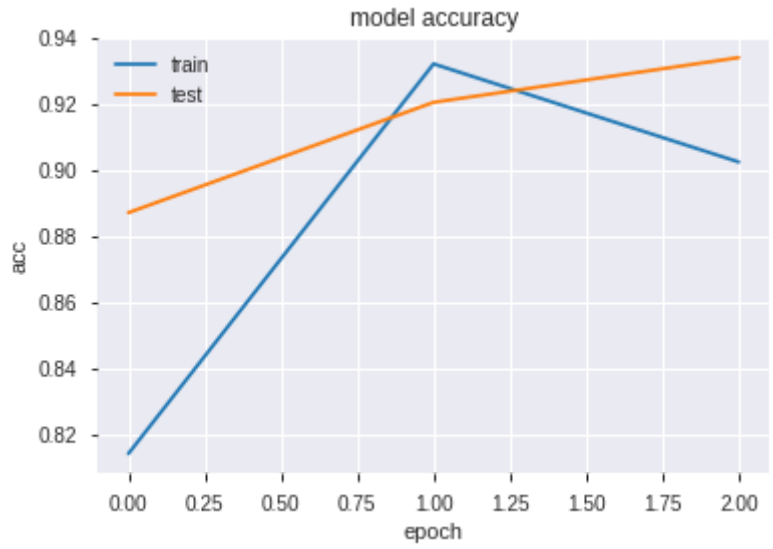
In [81]:
```
#Serializar pesos em HDF5
model_lstm.save_weights("model_ltsm.h5")
print("Modelo Salvo")
```

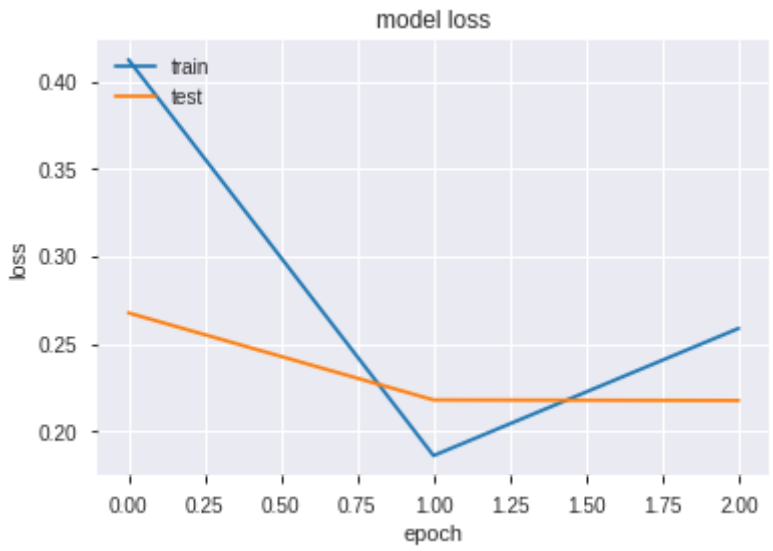
```
Modelo Salvo
```

In [82]:
```
#Verificação dos Históricos
print(history.history.keys())
```

```
dict_keys(['val_loss', 'val_accuracy', 'loss', 'accuracy'])
```

In [84]:
```
#Verificação Resultados modo Gráfico
import matplotlib.pyplot as plt
from matplotlib.legend_handler import HandlerLine2D

# resumir o histórico para precisão (accuracy)
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('acc')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

# resumir o histórico de perda (loss)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```





In [85]:
```
# Resultado do Treinamento
scores = model_lstm.evaluate(X_test_seq, y_test, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
```

```
Accuracy: 93.41%
```

Testando o Modelo

Após a conclusão do modelo de predição, foi criado um pequeno conjunto de dados para testes, composto de 20 notícias, sendo 10 notícias consideradas verdades extraídas de sites confiáveis como CNN, The New York Times, Aljazeera, Bloomberg, Reuters, e 10 notícias de sites não conhecidos, ou pouco confiáveis como, Viral Cord, The Onion, Real News Right Now, Empire News, National Report. Estas notícias foram retiradas do site chamado FACTITIOUS, uma plataforma de jogos, onde o desafio é identificar se a notícia falsa ou não.

A estrutura do conjunto de dados para teste segue abaixo:

id	title	author	text	label	journal
0-20	notícia	escritor	matéria	fake ou não	local

Preparação do Ambiente de Teste

In [586]:
```
## Recebendo Conjunto de Dados
path_test = 'resources/datasets/test.csv'

test_data = pd.read_csv(path_test, header=0, encoding = 'unicode_escape',sep=';')
```

Perfil do Conjunto de Dados

In [587]:
```
test_data.head()
```

Out[587]:

	id	title	author	text	label	journal
0	0	Bernie Sanders had a heart attack	Deanna Hackney and Caroline Kelly	(CNN)Democratic presidential candidate Sen. Be...	0	cnn
1	1	Alabama Requires Birth Certificate To Use Publ...	Viral Cords	Alabama is the first country to enact insane b...	1	Viral Cords
2	2	Iranian Hackers Target Trump Campaign as Threa...	Nicole Perlroth and David E. Sanger	SAN FRANCISCO The 2020 presidential election...	0	the new york times
3	3	Buckingham Palace Guards Impressed By First La...	the onion	LONDONSaying the first ladys skills were of ...	1	the onion
4	4	Los Angeles Tap Water Contains 18% Xanax and 7...	Viral Cords	In new Environmental Protection Agency (EPA) f...	1	Viral Cords

In [588]:
```
## criando novos recursos
test_data ['title_author_text'] = test_data ['title'] + ' ' + test_data ['author'] + ' ' + test_data ['text']
test_data ['len_title_author_text'] = [len(str(x)) for x in test_data ['title_author_text']]
```

In [589]:
```
detail = test_data['len_title_author_text'].describe()
print(detail)
```

```
count      20.000000
mean     3441.100000
std      2768.796126
min       908.000000
25%      1437.500000
50%      2155.500000
75%      4816.500000
max      9677.000000
Name: len_title_author_text, dtype: float64
```

In [590]:
```
## Separação das features X e Y
X_test = test_data['title_author_text']
Y_test = test_data['label']
```

In [591]:
```
## Tokenizando o texto
num_token = 7000
token = Tokenizer(num_words = num_token, filters = '!"#$%&()*+,-./:;<=>?@[\\]^_`{|}~\t\n')
token.fit_on_texts(X_test.astype(str))
```

In [592]:
```
## Sequenciando os tokens
max_review_length = 7000

X_test_token = token.texts_to_sequences(X_test.astype(str))
X_test_seq = sequence.pad_sequences(X_test_token, maxlen=max_review_length)
```

Utilizando Modelo Treinado

In [593]:
```
from keras.models import model_from_json

json_file = open('resources/model_lstm.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
loaded_model = model_from_json(loaded_model_json)

loaded_model.load_weights("resources/model_ltsm.h5")
print("Modelo carregado do Arquivo")

loaded_model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
Modelo carregado do Arquivo
```

Predição

In [594]:
```
scores = loaded_model.evaluate(X_test_seq, Y_test, verbose=0)
print("%s: %.2f%%" % (loaded_model.metrics_names[1], scores[1]*100))
```

```
acc: 55.00%
```

In [595]:
```
predictions = loaded_model.predict_classes(X_test_seq)
```

In [596]:
```
print(predictions.shape)
```

```
(20, 1)
```

In [597]:
```
rounded = [np.round(x) for x in predictions]
rounded
```

Out[597]:
```
[array([1]),
 array([1]),
 array([0]),
 array([1]),
 array([1]),
 array([1]),
 array([1]),
 array([0]),
 array([0]),
 array([0]),
 array([1]),
 array([1]),
 array([1]),
 array([1]),
 array([1]),
 array([1]),
 array([1]),
 array([0]),
 array([1]),
 array([1]),
 array([1])]
```

Os dados têm a seguinte informação:

- 0 Notícia Verdadeira
- 1 Notícia Falsa

Resultados

Este trabalho possibilitou entender a forma como as notícias falsas estão em nossos dias e os problemas que elas estão gerando a sociedade. Com isso, pôde-se perceber a necessidade de projetos científicos que considerem a criação de uma ferramenta que possibilitasse a detecção de fake news.

Para se atingir o objetivo, definiram-se duas premissas. A primeira, de como seria o conjunto de dados, demandou a pesquisa de um conjunto de dados consistente e com notícias rotuladas como verdadeiras ou falsas. Percebeu-se uma escassez para encontrar um conjunto de dados que fosse de fonte confiável.

Após encontrado as amostras de dados, deu-se início ao segundo objetivo: desenvolver um aprendizado de máquina capaz de identificar uma notícia falsa.

Como já detalhado no capítulo de análise exploratória, tem-se de maneira geral, um padrão utilizado nas notícias classificadas como fake news. Nas análises gráficas apontaram uma disparidade do uso linguístico em comparação com a notícia verdadeira. Geralmente a notícia falsa mostrou conter textos menores e com mais pronomes pessoais.

Embora a análise gráfica não influenciou no treinamento do aprendizado de máquina, foi importante entender o padrão que consiste em uma notícia falsa, podendo ser útil em uma futura pesquisa.

O treinamento do modelo apresentou resultado satisfatório para este início de pesquisa. Conseguimos criar uma máquina capaz de acertar 55% dos resultados, entretanto o modelo conseguiu identificar 80% das vezes uma notícia falsa, visto que o conjunto de teste que utilizamos é composto por 10 notícias verdadeiras e 10 notícias falsas, porém, o modelo apresenta problema para identificar notícia verdadeira.

Trabalhos Futuros

Ficou para trabalhos futuros a melhoria desse modelo e o desenvolvimento de uma extensão para navegadores web.

Referências

[1] BLOCH, March, Réflexions d’un historien sur les fausses nouvelles de la guerre. Disponível em: https://fr.wikisource.org/wiki/R%C3%A9flexions_d%E2%80%99un_historien_sur_les_fausses_nouvelles_de_la_guerre. Acesso em: 01 out. 2019.

[2] LUCENDO, Guillermo Altares. A longa história das notícias falsas, 18 jun 2018. Disponível em: https://brasil.elpais.com/brasil/2018/06/08/cultura/1528467298_389944.html. Acesso em: 01 out. 2019.

[3] David M. J. Lazer, Matthew A. Baum, Yochai Benkler, Adam J. Berinsky, Kelly M. Greenhill, Filippo Menczer, Miriam J.Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, Michael Schudson, Steven A. Sloman, Cass R. Sunstein, Emily A. Thorson, Duncan J. Watts and Jonathan L. Zittrain. The science of fake news. Disponível em: https://scholar.harvard.edu/files/mbaum/files/science_of_fake_news.pdf. Acesso em: 01 out. 2019.

[4] S. Vosoughi et al., Science 359, 1146 (2018).The spread of true and false news online. Disponível em: https://science.sciencemag.org/content/359/6380/1146. Acesso em: 01 out. 2019.

[5] DAVIDSON, Thomas et al. Automated Hate Speech Detection and the Problem of Offensive Language. 2013. 4 f. Department of Applied Mathematics (Qatar Computing Research Institute) - Cornell University, 3Department of Information Science, Ithaca, NY, USA, 2013.

[6] RUBIN, Victoria L. et al. Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News. 2016. 11 p. Language and Information Technology Research Lab (Faculty of Information and Media Studies) - University of Western Ontario, London, Ontario, CANADA, 2016.

[8] LIWICKI, Marcus et al. A Novel Approach to On-Line Handwriting Recognition Based on Bidirectional Long Short-Term Memory Networks. [S. l.], 2019. Disponível em: https://mediatum.ub.tum.de/doc/1289961/file.pdf. Acesso em: 9 jul. 2019.

[7]HAYKIN, S. Neural Networks: A Comprehensive Foundation. 2nd. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998. ISBN 0132733501.Acesso em: 9 ago. 2019.

[9] WASON, R. Deep learning: Evolution and expansion. Cognitive Systems Research, v. 52, p. 701 – 708, 2018. ISSN 1389-0417.Acesso em: 9 ago. 2019.

[10] MICHAEL, David, NELSON, Quirino. USO DE REDES NEURAIS RECORRENTES PARA PREVISÃO DE SÉRIES TEMPORAIS FINANCEIRAS. Disponível em: https://www.dcc.ufmg.br/pos/cursos/defesas/1999M.PDF. Acesso em: 9 jul. 2019.

[11] COELHO, Iasmin. ESTUDO SOBRE A APLICAÇÃO DE REDES NEURAIS RECORRENTES PARA PREVISÃO DA GERAÇÃO EÓLICA E DO PREÇO DE LIQUIDAÇÃO DAS DIFERENÇAS. Disponível em: https://repositorio.ufsc.br/bitstream/handle/123456789/192503/TCC%20-%20Iasmin%20Coelho.pdf?sequence=1&isAllowed=y. Acesso em: 9 jul. 2019.

[12] COUTINHO, Lucas.APLICAÇÃO DE REDES NEURAIS LSTM PARA A PREVISÃO DE CURTO PRAZO DE VAZÃO DO RIO PARAÍBA DO SUL. Disponível em: http://www.ufjf.br/engcomputacional/files/2018/09/Monografia_LucasVassalli.pdf. Acesso em: 9 jul. 2019.

[13] SHAHEBAZ, Mohammad. Hand Crafted Feature Engineering For Insincerity. Disponível em: <https://www.kaggle.com/shaz13/feature-engineering-for-nlp-classification/notebook#Tagging-Parts-Of-Speech-And-More-Feature-Engineering. Acesso em: 14 set. 2019.

[14] UTK Machine Learning Club. Fake News - Build a system to identify unreliable news articles. 2017. Disponível em: https://www.kaggle.com/c/fake-news/data. Acesso em: 28 jul. 2019.

[15] AQUINO, Bruno. 5-fold LSTM Attention (fully commented). Disponível em: https://www.kaggle.com/braquino/5-fold-lstm-attention-fully-commented-0-694/. Acesso em: 03 out. 2019.

[16] AU Game Lab, the JoLT program. factitious project. Disponível em: <http://factitious.augamestudio.com/#/. Acesso em: 04 out. 2019.

[17] SUHANKO, Djames. Como salvar um model treinado com Keras. Disponível em: <https://www.dobitaobyte.com.br/como-salvar-um-model-treinado-com-keras/. Acesso em: 04 out. 2019.

[18] PEREZ-ROSAS, Verônica et al. Automatic Detection of Fake News. 2016. 10 f. Department of Psychology (Computer Science and Engineering,) - University of Michigan, [S.l.], 2016.

[19] BROWNLEE, Jason. Dropout Regularization in Deep Learning Models With Keras. [S. l.], 20 jun. 2016. Disponível em: https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/. Acesso em: 9 jul. 2019.

[20] Frei , Lukas. Speed Up Your Exploratory Data Analysis With Pandas-Profiling. [S. l.], 25 apr. 2019. Disponível em: https://towardsdatascience.com/speed-up-your-exploratory-data-analysis-with-pandas-profiling-88b33dc53625. Acesso em: 9 jul. 2019.

[21] BROWNLEE, Jason. Display Deep Learning Model Training History in Keras. Disponível em: <https://machinelearningmastery.com/display-deep-learning-model-training-history-in-keras/. Acesso em: 02 out. 2019.

[22] TEIXEIRA, Fabrício. Análise heurística: como fazer e os benefícios para o projeto. Disponível em: https://brasil.uxdesign.cc/an%C3%A1lise-heur%C3%ADstica-o-que-%C3%A9-como-fazer-e-os-benef%C3%ADcios-para-o-projeto-161f3d94436b. Acesso em: 02 out. 2019.

[23] WIKIPEDIA. Histograma. Disponível em: https://pt.wikipedia.org/wiki/Histograma. Acesso em: 02 out. 2019.