

UNIVERSIDADE PAULISTA

MATHEUS FELIPE DOS ANJOS

**ANÁLISE DE SENTIMENTOS EM TEXTOS CURTOS DIRIGIDOS À ELEIÇÃO
PRESIDENCIAL DO BRASIL EM 2018.**

Limeira
2018

UNIVERSIDADE PAULISTA

MATHEUS FELIPE DOS ANJOS

**ANÁLISE DE SENTIMENTOS EM TEXTOS CURTOS DIRIGIDOS À ELEIÇÃO
PRESIDENCIAL DO BRASIL EM 2018.**

Trabalho de conclusão de curso apresentado à banca examinadora da Universidade Paulista, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação sob a orientação do professor Me. Amaury Bosso André.

Limeira

2018

Matheus Felipe dos Anjos

**ANÁLISE DE SENTIMENTOS EM TEXTOS CURTOS DIRIGIDOS À ELEIÇÃO
PRESIDENCIAL DO BRASIL EM 2018.**

Trabalho de conclusão de curso apresentado à banca examinadora da Universidade Paulista, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação sob a orientação do professor Me. Amaury Bosso André.

Aprovada em 03 de dezembro de 2018.

BANCA EXAMINADORA

Prof. Dr.

Prof. Me.

Prof. Esp.

Dedico este trabalho aos meus amigos, professores e principalmente aos meus pais por todo o incentivo e apoio ao longo desses anos de graduação, os quais favoreceram para meu amadurecimento pessoal, profissional e acadêmico.

AGRADECIMENTOS

Agradeço a todos os professores da graduação, em especial aos docentes Sérgio Nunes, Marcos Gialdi, Mateus Locci e Amaury André pelas horas de conversas que puderam resultar no amadurecimento e conclusão desse trabalho.

Ao meu orientador Amaury, por todo tempo que se dedicou a me ajudar durante o processo de realização deste trabalho, além de ter me proporcionado uma excelente oportunidade profissional na área.

Aos meus colegas de classe que sempre se mostraram curiosos e interessados quanto ao meu trabalho e sempre se mostraram dispostos a elogiar, criticar e sugerir.

Também gostaria de agradecer aos meus amigos pelo apoio ao longo de minha vida e principalmente durante a graduação.

Por fim, agradeço ao meu pai e minha mãe que são os responsáveis por tudo isso se tornar realidade. Que me apoiaram nos momentos mais difíceis e fizeram de tudo para que eu obtivesse o título de bacharel em ciência da computação.

*"Se eu vi mais longe, foi por estar sobre
ombros de gigantes."
(Isaac Newton)*

LISTA DE ILUSTRAÇÕES

Figura 1 - Ilustração de um neurônio biológico.....	11
Figura 2 - Funcionamento de um simples neurônio computacional	12
Figura 3 - Camadas de uma rede neural.....	13
Figura 4 - Representação das etapas de uma rede neural convolucional.....	14
Figura 5 - Representação de uma imagem após o uso de filtro.	15
Figura 6 - Representação de <i>pooling</i>	15
Figura 7 - Rede neural recorrente	16
Figura 8 - Funcionamento interno de neurônio em uma rede LSTM	17
Figura 9 - Exemplos de <i>stopwords</i> em português	19
Figura 10 - Diagrama representando rede neural de um canal.....	22
Figura 11 - Diagrama representando rede neural de três canais.	23
Figura 12 - Representação de cada modelo e seu melhor valor de <i>threshold</i>	25
Figura 13 - Taxa de comentários positivos e negativos para os candidatos no primeiro turno	28
Figura 14 - Mapa do Brasil representando a proporção entre <i>tweets</i> dirigidos aos candidatos no segundo turno por estado	30
Figura 15 - Taxa de comentários positivos e negativos do candidato Bolsonaro	30
Figura 16 - Taxa de comentários positivos e negativos do candidato Haddad.	31

LISTA DE TABELAS

Tabela 1 - Representação de palavras vetorizadas	20
Tabela 2 - Métricas de cada modelo por <i>threshold</i>	24
Tabela 3 – Palavras chaves para capturar <i>tweets</i> de cada candidato	26
Tabela 4 – Candidatos e suas proporções de <i>tweets</i> exclusivos	26
Tabela 5 – Regiões e suas proporções da amostra	29

LISTA DE ABREVIATURAS E SIGLAS

CNN	<i>Convolutional Neural Network</i>
CNTK	<i>Microsoft Cognitive Toolkit</i>
FNN	<i>Feedforward Neural Network</i>
LSTM	<i>Long Short-Term Memory</i>
MDB	Movimento Democrático Brasileiro
MLP	<i>Multilayer Perceptron</i>
Novo	Partido Novo
Patri	Partido Patriota
PDT	Partido Democrático Trabalhista
PLN	Processamento de Linguagem Natural
Pode	Podemos
PSDB	Partido da Social Democracia Brasileira
PSL	Partido Social Liberal
PSOL	Partido Socialismo e Liberdade
PT	Partido dos Trabalhadores
Rede	Rede Sustentabilidade
RNA	Rede Neural Artificial
RNN	<i>Recurrent Neural Network</i>
SBT	Sistema Brasileiro de Televisão
SVM	<i>Support Vector Machine</i>

RESUMO

A ascensão de manifestações realizadas em redes sociais, principalmente no *Facebook* e *Twitter* gera uma quantidade colossal de dados. Apenas pequenas porções desses dados são analisados para gerar informações de viés político que podem auxiliar eleitores e principalmente equipes de assessores a entenderem melhor o colégio eleitoral de determinada região ou estado durante as eleições. A análise de sentimentos vem com o propósito de classificar uma sentença textual entre os mais diversos sentimentos, o qual nesse trabalho será apenas em positivo ou negativo. Através desse propósito, discorrendo especialmente sobre a ciência política é proposto modelos complexos de redes neurais computacionais que são capazes de gerar informações relevantes quanto aos discursos dos eleitores nas redes sociais referente ao meio político durante as eleições do principal cargo executivo na república brasileira. Para que esses resultados sejam relevantes e assertivos, serão utilizadas técnicas de processamento de linguagem natural juntamente com uma gama de arquiteturas de redes neurais disponíveis, dos quais a partir deles serão desenvolvidos dois modelos distintos. Desses, o que apresentar melhores resultados para análise de sentimento em textos, será escolhido como o principal modelo a ser utilizado durante o decorrer do trabalho e através desse apresentar os resultados alcançados nos *tweets* capturados durante os debates televisivos entre os candidatos.

Palavras-chave: Redes neurais. Processamento de linguagem natural. Eleições. Análise de sentimentos.

ABSTRACT

The rise of manifestations carried out on social networks, especially on Facebook and Twitter generates a colossal amount of data. Only small portions of this data are analyzed to generate political bias information that can assist voters and especially advisory teams to better understand the electoral college of a particular region or state during the elections. The sentiment analysis comes with the purpose of classifying a textual sentence among the most diverse sentiments, which in this work will be only positive or negative. Through this purpose, discussing especially on political science, it is proposed complex neural networks computational models that are capable of generating relevant information about the voters' discourses in social networks regarding the political environment during the elections of the main executive position in the Brazilian republic. In order for these results to be relevant and assertive, natural language processing techniques will be used along with a range of available neural network architectures, from which two distinct models will be developed. Of these, the one that presents better results for sentiment analysis in texts, will be chosen as the main model to be used during the course of the work and through this to present results achieved in the tweets captured during the televising debates between the candidates.

Keywords: Neural networks. Natural language processing. Elections. Sentiment analysis.

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Objetivos	8
1.1.1	Objetivo geral.....	9
1.1.2	Objetivos específicos.....	9
1.2	Justificativa	9
1.3	Metodologia	10
1.4	Estrutura do documento	10
2	FUNDAMENTAÇÃO TEÓRICA	11
2.1	Redes neurais artificiais	11
2.1.1	Rede neural perceptrons multicamadas	12
2.1.2	Rede neural convolucional	14
2.1.3	Rede neural recorrente.....	16
2.1.3.1	Rede neural <i>Long Short Term-memory</i>	17
2.2	Processamento de linguagem natural para <i>deep learning</i>	18
2.2.1	Pré processamento.....	18
2.2.2	Processamento em redes neurais	19
2.2.3	Camada <i>Embedding</i>	20
3	DESENVOLVIMENTO DA APLICAÇÃO	21
3.1	Base de dados	21
3.2	Modelos de redes neurais	22
3.2.1	Resultados.....	23
3.3	Extração dos <i>tweets</i>	26
4	RESULTADOS	28
4.1	Regiões	29
5	CONSIDERAÇÕES FINAIS	32
	REFERÊNCIAS BIBLIOGRÁFICAS	33

1 INTRODUÇÃO

A utilização de aprendizado de máquina tem sido efetiva em diversas áreas, como: economia, ciências sociais, negócios, mecânica, agricultura, saúde e política. Esse último, o qual será abordado ao decorrer de toda essa dissertação, já foi bastante utilizado fazendo-se o uso de algoritmos tradicionais de *machine learning* em língua portuguesa (CARVALHO; NAGANO; BARROS, 2017). Porém, a proposta desse documento será a utilização de técnicas de *deep learning* para a realização de análise de sentimentos (AIN; *et al*, 2017) durante os debates realizados na eleição presidencial no Brasil em 2018.

A análise de sentimentos é amplamente utilizada para o mapeamento das opiniões dos usuários, realizando a classificação sentimental de um texto, podendo ser positivo ou negativo. Com a aplicação desenvolvida nesse projeto é possível que equipes de assessores e partidos políticos consigam identificar fragilidades na campanha, aceitação e rejeição dos candidatos.

Técnicas de *deep learning* para processamentos de textos estão em evidências a poucos anos e ainda têm muito o que ser descoberto e aprimorado na área, ainda mais quando tratamos dessas técnicas para a língua portuguesa, que é uma das principais propostas desse documento.

Ao contrário de alguns trabalhos já propostos para as eleições dos Estados Unidos da América em 2016 (SALUNKHE; SURNAR; SONAWANE, 2017), a intenção da aplicação a ser desenvolvida não tem como principal requisito prever quem será o próximo presidente, mas sim, encontrar correlações de intenção de voto e rejeição entre as pesquisas, resultados das eleições e a análise de sentimentos nas manifestações do público em rede social, mais especificamente no *Twitter*.

1.1 Objetivos

O desenvolvimento desse trabalho tem basicamente dois objetivos: tecnológico – contribuindo diretamente com o estudo de inteligência artificial, comprovar a acurácia das redes neurais no processo de classificação textual em

língua portuguesa; sociopolítico – realizando análises de sentimentos das manifestações online dos eleitores brasileiros durante toda a corrida eleitoral.

1.1.1 Objetivo geral

O objetivo principal desse trabalho tem como finalidade o estudo de teorias e desenvolvimento prático de um sistema inteligente capaz de realizar a classificação de comentários disseminados através das redes sociais.

Por meio dessas informações recolhidas e devidamente tratadas para o treinamento do algoritmo, será realizada uma classificação com os novos dados, para a avaliação do aprendizado realizado pelo sistema – uma espécie de termômetro – das eleições presidenciais na República Federativa do Brasil no ano de 2018.

1.1.2 Objetivos específicos

- a) Desenvolver e mensurar a acuracidade em duas variações de redes neurais para classificação textual.
- b) Comparar os resultados das eleições presidenciais com as divulgadas pelos meios de pesquisas e principalmente com os frutos gerados pelos estudos dessa proposta.
- c) Avaliar a confiança do algoritmo referente a classificação com novas informações.
- d) Apresentar de maneira gráfica as conclusões contraídas com as análises de sentimentos.

1.2 Justificativa

Cada vez mais os brasileiros estão se interessando por política, valorizando ainda mais os seus votos, e se manifestando nas redes sociais sobre algum assunto em pauta no parlamento ou até mesmo sobre algum político. Visualizando essa crescente politização em território nacional esse trabalho vem com um propósito de analisar as manifestações políticas dos eleitores no *Twitter*.

1.3 Metodologia

Através de dados extraídos do *Twitter*, será realizado um tratamento com os textos obtidos, por meio de técnicas de processamento de linguagem natural. Conseqüentemente o processamento das palavras por meio de vetorização das mesmas, que tem como finalidade ser a camada de entrada em uma rede neural.

Posteriormente será realizada a construção de redes neurais com várias camadas para que o aprendizado possa ser iniciado. Então, a rede neural em si será avaliada com métricas estatísticas. A base de dados a ser utilizada no decorrer desse estudo é única e foi capturada de forma automática com alguns critérios, o que será explicado futuramente.

Após todo esse processo, novos textos serão introduzidos para percorrer todo aprendizado gerado pelo algoritmo. Então o novo texto – não utilizado para treinamento – será classificado de acordo com o aprendizado.

1.4 Estrutura do documento

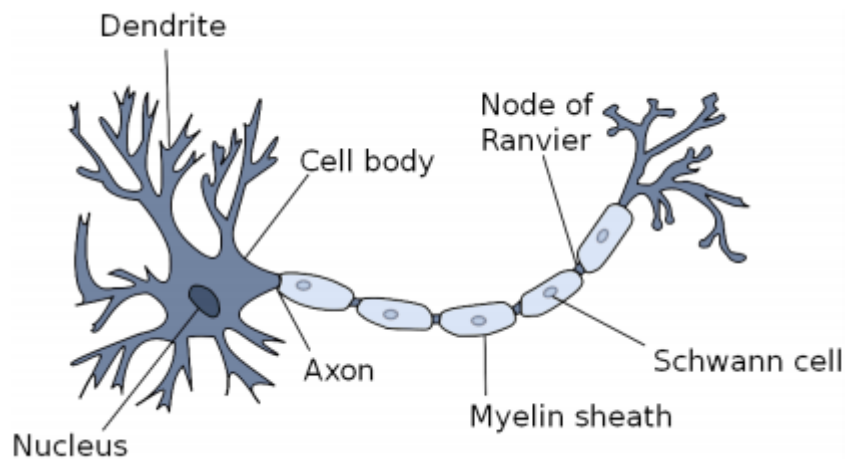
O documento será estruturado basicamente em três módulos: a parte teórica, que irá abranger todos os fundamentos utilizados para o desenvolvimento da segunda, que será o desenvolvimento prático da aplicação. Conseqüentemente as últimas seções do trabalho apresentarão os resultados alcançados e as considerações finais do mesmo.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Redes neurais artificiais

De acordo com Kevin Gurney (1997), Redes Neurais Artificiais (RNA) são modelos computacionais capazes de simular o funcionamento de um cérebro animal inteligente através de modelos matemáticos e cognitivos.

Figura 1 - Ilustração de um neurônio biológico.



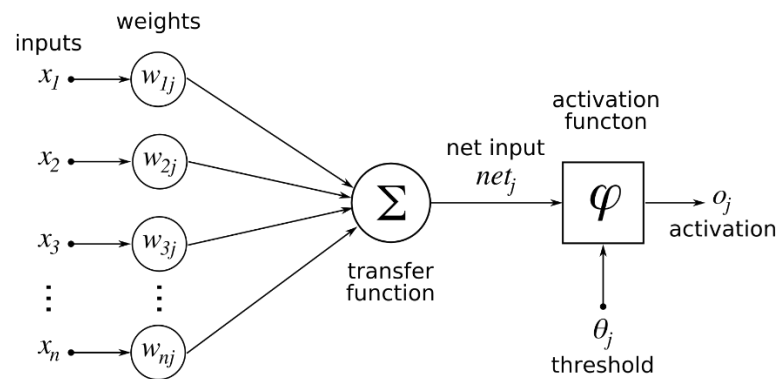
Fonte: *A Brief Introduction to Neural Networks* (2007).

Ainda segundo Kevin Gurney, o termo “rede neural” pode ser referenciado como sendo um simples nó com apenas um neurônio ou uma extensa rede com múltiplos neurônios e diversas camadas.

O conceito foi introduzido no início dos anos 40 pelo neuroanatomista e psiquiatra Warren McCullo e pelo cientista cognitivo Walter Pitts. Eles propuseram um simples modelo de neurônio computacional conhecido como MCP (McCulloch-Pitts).

Segundo McCulloch e Pitts (1943) o sistema nervoso animal é uma rede de neurônios que contém conexões entre si, também conhecidos como sinapses. A conexão desses neurônios alimentados entre si é capaz de gerar informações relevantes para o aprendizado e conseqüentemente a realização de tarefas.

Figura 2 - Funcionamento de um simples neurônio computacional.



Fonte: *KDNuggets* (2016).

O conceito por trás do primeiro modelo de rede neural proposto por McCulloch e Pitts, ilustrado na figura 2, consiste em um número de entradas (x) – exemplos de treinamento – que são multiplicadas por seus respectivos pesos (w), que normalmente são iniciados com valores aleatórios e ajustados ao decorrer do processo de aprendizagem. É então realizada uma média ponderada entre a soma da multiplicação dos pesos e dos valores de entrada, os quais proverão os devidos valores para a função de ativação, que tem como objetivo transformar o valor de entrada em intervalos definidos, como por exemplo entre 0 e 1 (função sigmoide). O *perceptron* – nome dado à neurônios computacionais – será ativado apenas se o valor de entrada do mesmo alcançar o seu valor de ativação (*threshold*).

2.1.1 Rede neural *perceptrons* multicamadas

Anos depois foi apresentado o que hoje chamamos de uma Rede Neural *Perceptrons* Multicamadas, ou MLP (ROSENBLATT, 1958), que basicamente aplica algumas modificações ao modelo original apresentado em 1943 pela dupla McCulloch e Pitts. Esse modelo possibilita a utilização de mais camadas e consequentemente mais neurônios. Esse conceito é o mais utilizado e o mais aceito até hoje, possibilitando a resolução de problemas e execução de tarefas não lineares, ou seja, uma estrutura que apresente múltiplos caminhos distintos e complexos.

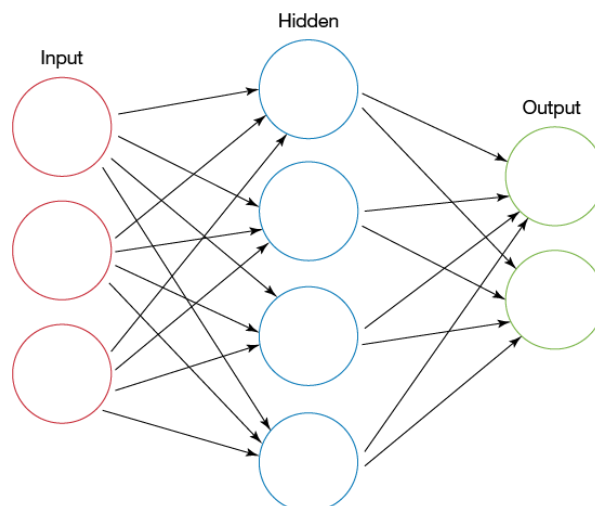
Um sistema de inteligência artificial baseado em *deep learning*, consiste basicamente em 3 etapas, influenciado pelo modelo proposto por Rosenblatt: camada de entrada, camada oculta e a camada de saída.

A primeira receberá os dados já devidamente pré processados que serão utilizados como base de treinamento do modelo, no qual será realizado o aprendizado.

Na seguinte ocorre as atualizações dos pesos entre as conexões das camadas. A regulagem dos pesos depende de diversos fatores, tais como funções de perda e otimizadores utilizados durante o treinamento do algoritmo, que conjunto ao algoritmo de retropropagação, otimizará os pesos para que os valores da camada seguinte possam chegar o mais próximo possível da classe ou valor desejado como saída.

Em sua última etapa, a RNA retorna um valor, ou valores, de saída que são consequência de todo o aprendizado nas camadas anteriores. Esse mesmo valor é influenciado pela ativação utilizada na última camada de uma rede neural.

Figura 3 - Camadas de uma rede neural.



Fonte: IBM (2018).

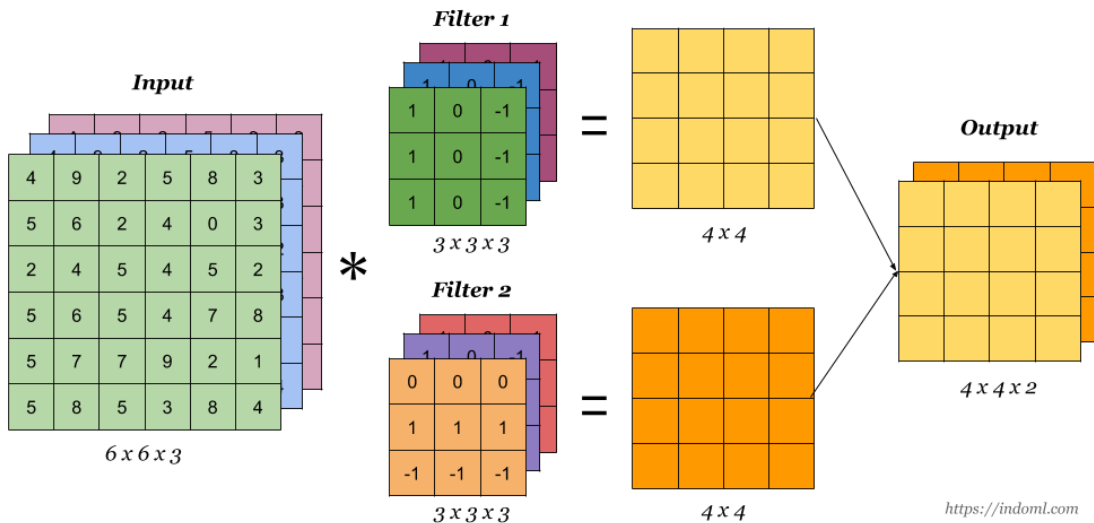
Como dito anteriormente, as redes neurais atuais trabalham com o algoritmo de retropropagação. Com o mesmo, calculamos o gradiente da função de perda – do valor previsto com o valor desejado como saída – na última camada e assim aplicamos recursivamente, de trás para frente, diversos ajustes de pesos entre as conexões (MAKIN, 2006).

Dentre as principais conquistas das redes neurais, podemos citar o reconhecimento de imagens e objetos, utilizando-se redes convolucionais e recorrentes, que são extremamente utilizadas em previsões de séries temporais e também em reconhecimento e geração de textos (GREFF, 2017), que será a principal rede neural a ser utilizada ao longo desse trabalho.

2.1.2 Rede neural convolucional

As redes neurais convolucionais (LeCun *et al*, 1989), também conhecidas do inglês como CNN, têm alcançado ótimos resultados, principalmente tratando-se de reconhecimento de imagens e detecção de objetos. Porém nos últimos anos tem-se mostrado capaz de ser aplicada em demais finalidades, auxiliando as redes neurais recorrentes (ZHOU; *et al*, 2015).

Figura 4 - Representação das etapas de uma rede neural convolucional.



Fonte: IndoML (2018).

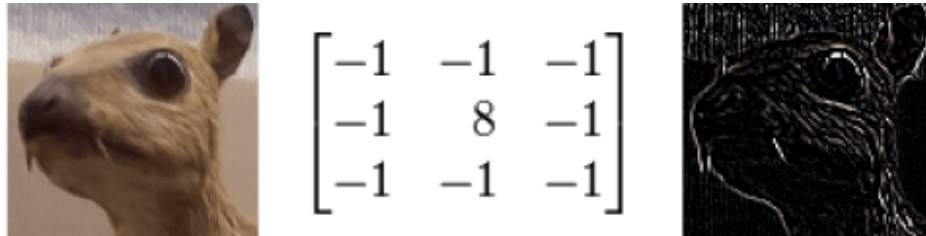
Como o próprio nome sugere, em uma CNN são aplicadas diversas convoluções em todos os valores dos vetores ou matrizes que foram passados como entrada em uma rede.

Convolução é uma operação extremamente utilizada para processamento de sinais (SOVIERZOSKI, 2010) que opera com duas funções para gerar uma terceira. Nas redes convolucionais que utilizamos para processamento de texto e imagem, essa terceira função dá origem ao mapa de características, decorrente de diversas convoluções realizadas durante as camadas.

Inicialmente os filtros utilizados para as operações de convoluções são gerados de forma aleatória, assim como acontece com os pesos em uma rede neural clássica. Com o decorrer do treinamento e o algoritmo de retropropagação, a rede

neural aprende quais os melhores filtros a serem utilizados para a classificação ou regressão de novos valores (ZHANG, 2016).

Figura 5 - Representação de uma imagem após o uso de filtro.



Fonte: Tim Dettmers (2015)

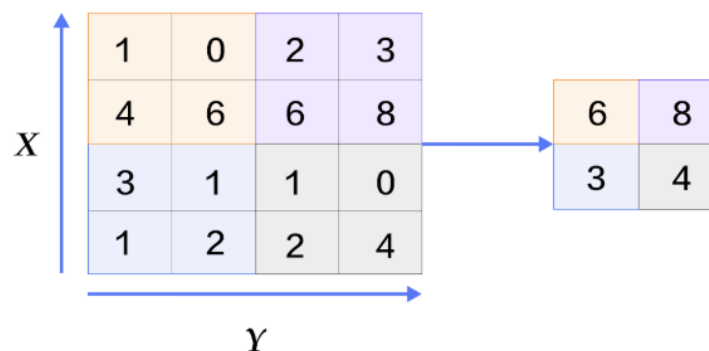
Na figura 5 foi aplicada uma convolução com núcleo (ou *kernel*), que tem como objetivo identificar traços relevantes em uma imagem.

Filtros semelhantes a esse podem ser utilizados não somente em imagens, mas em dados estruturados e textos, gerando novos valores para o conjunto de treino, e assim, generalizando o aprendizado para novas entradas.

Comumente, após o processo de convolução, os novos valores são aplicados a técnica que chamamos de *pooling*, que tem como objetivo reduzir o tamanho de uma matriz ou vetor, mantendo apenas os valores mais relevantes.

Todos esses processos citados anteriormente, utilizando-se convoluções, filtros e a técnica de *pooling*, são essenciais para gerar os mapas de características (ou *features maps*). Utilizando-se desses conceitos e técnicas é possível obter melhores resultados, além de um algoritmo e treinamento mais otimizado, reduzindo tempo e custo computacional, levando em conta que os parâmetros serão reduzidos ao decorrer desse processo (THOMA *et al*, 2017).

Figura 6 - Representação de *pooling*.



Fonte: PadlePaddle (2018).

A utilização de CNN em textos tem se mostrado promissora e com o auxílio de outras redes neurais, vem alcançando ótimos resultados em classificações textuais, como análise de sentimentos.

2.1.3 Rede neural recorrente

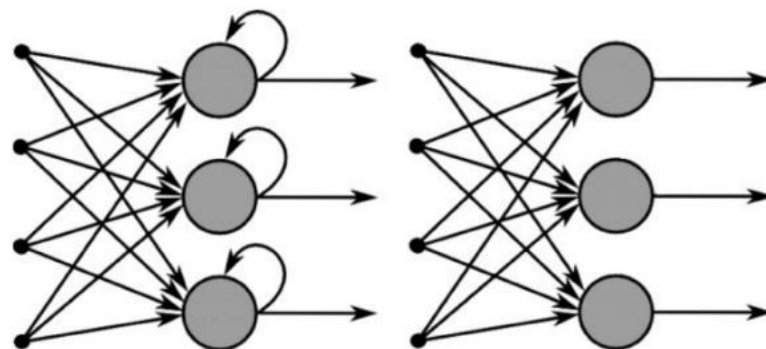
As redes neurais recorrentes (RUMELHART *et al.*, 1986), conhecidas no inglês como RNN, são extremamente aproveitadas quando trabalhamos com dados temporais.

Como exemplos de aplicações nas quais são recomendadas o uso de RNN estão reconhecimento de áudio, texto, imagens, previsão do tempo, entre outros.

Tudo isso é possível porquê os neurônios de uma rede neural desse tipo trabalham com o conceito de recorrência, pois o mesmo neurônio se retroalimenta com seus próprios valores gerados na saída.

Em uma rede neural comum, como MLP, os dados fluem em apenas uma direção, nunca passando duas vezes pelo mesmo neurônio – ou nó – diferente de uma rede neural recorrente. Em uma RNN, os valores na camada oculta, fluem em direção à camada de saída, mas também os próprios neurônios são alimentados com os valores de saída dos mesmos, gerando a sensação de que esses modelos computacionais possuem memória.

Figura 7 - Rede neural recorrente.



Fonte: Towards Data Science (2018).

O conceito apresentado por Rumelhart, na segunda metade dos anos 80, foi visto por bons olhos perante a ciência computacional, mas por um tempo a ideia referente a esse tipo de rede neural não evoluiu, pelo fato de que uma espécie de redes neurais dessa complexidade não conseguiria ser executada com êxito e da

maneira desejada, pelo déficit computacional na época – o que nos dias atuais não são mais um dos grandes problemas da computação.

Apesar dos avanços científicos revelados pelas RNNs, elas sofriam com o que chamamos de Memória de Curto Termo (*Short Term Memory*), ou seja, tratando-se de dados temporais e sequenciais, o modelo computacional de certo modo acaba “esquecendo” os valores relevantes que antecedem a instância a ser processada no presente.

2.1.3.1 Rede neural *Long Short Term-memory*

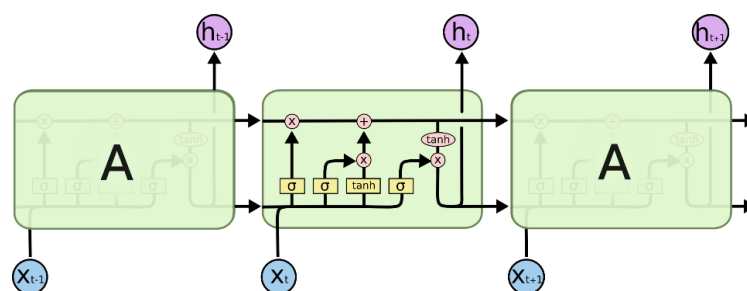
Para contornar os problemas referentes a curta memória dos neurônios em uma arquitetura de rede neural recorrente, foi apresentado a rede neural com Memória de Longo e Curto Termo (*Long Short Term-Memory*), conhecida simplesmente como LSTM (HOCHREITER e SCHMIDHUBER, 1997).

A grande diferença das redes neurais recorrentes comuns para uma LSTM é que em seu *kernel* há unidades de memórias, que conseguem acumular informações (através das funções de ativação), relevantes dentro de sua unidade, mantendo informações de longo prazo, que era o principal problema da primeira RNN apresentada.

Na área de processamento de linguagem natural, ou simplesmente NLP, a LSTM tem apresentado ótimos resultados nos últimos anos para análise de sentimentos, classificação textual e geração de textos.

A transição de informações em cada nó acontece de maneira semelhante entre todas as variações de redes recorrente, porém no interior de seus núcleos há pequenas modificações que fazem uma grande diferença de resultados, como demonstrado na imagem abaixo.

Figura 8 - Funcionamento interno de um neurônio em uma rede LSTM.



Fonte: Colah's Blog (2015).

No diagrama da figura 8, os círculos dentro da unidade que estão em rosa, representam as operações realizadas ao juntar as duas linhas que carregam um vetor que já passou pela função de ativação – representados por um retângulo amarelo. Desse modo alguns valores são apreendidos no interior do neurônio e só são modificados quando alcançados os valores nas funções de ativação.

Como dito por Hochreiter e Schmidhuber, com os algoritmos de retropropagação através do tempo (WILLIAMS e ZIPSER, 1992), os pesos de uma rede neural recorrente comum podem crescer demasiadamente em problemas temporais mais complexos ou simplesmente desaparecer, decorrente dos problemas de longos termos já citados anteriormente. A LSTM é apresentada originalmente como um remédio para essa adversidade. A estrutura celular de uma unidade LSTM é capaz de aprender termos e intervalos distantes, até mesmo em casos em que haja ruído ou entradas incompreensíveis pelo modelo. Esse êxito é consequência de um método gradiente de otimização que não permite o modelo “explodir”, ou “sumir” com seus pesos (CHEN, 2018) e conseqüentemente não realizar o aprendizado desejado dos dados.

2.2 Processamento de linguagem natural para *deep learning*

O processamento de linguagem natural em aplicações envolvendo redes neurais acontece de maneira diferente, comparados aos algoritmos mais comuns de aprendizado de máquina para tal finalidade, como: árvores de decisão, *Naive Bayes* e SVM (BERGSMA *et al*, 2005). Tudo isso, pelo fato de que uma rede neural não consegue realizar o processamento de textos puros, sejam palavras, sílabas ou letras de forma individual.

Para que isso aconteça, temos de recorrer a diversas técnicas de pré-processamento para transformar textos em vetores (MAAS; *et al*, 2011), assim como em dados estruturados ou tabulares.

2.2.1 Pré processamento

Um dos primeiros passos a serem realizados para processamento de textos utilizando *deep learning* é o que chamamos de geração de *tokens*. Processo no qual

as palavras de uma sentença são separadas, semelhante ao processo de análise léxica em compiladores, assim cada palavra pode ser analisada de modo individual para a realização das etapas seguintes.

Também é realizado o processo de remoção de qualquer tipo de pontuação e acentuação nos textos, além de um outro passo extremamente importante, muito utilizado em sistemas de buscas, que é a remoção das chamadas *stopwords*.

Figura 9 - Exemplos de stopwords em português

a ao aos aquela aquelas aquele aqueles aqui aquilo as até ai com como da das de dela delas dele deles depois do dos dá e ela elas ele eles em entre era eram essa essas esse esses esta estamos estas estava estavam este esteja estejam estejamos estes esteve estive estivemos estiver estivera estiveram estiverem estivermos estivesse estivessem estivéramos estivéssemos estou está estávamos estão eu foi fomos for fora foram forem formos fosse fossem fui fôramos fôssemos haja hajam hajamos havemos houve houvermos houvera houveram houverei houverem houveremos houveria houveriam houvermos houverá houverão houveríamos houvesse houvessem houvéramos houvéssemos há hã isso isto já lhe lhes lá mais mas me mesmo meu meus minha minhas muito na nas nem no nos nossa nossas nosso nossos num numa não nós o os ou para pela pelas pelo pelos por pra qual quando que quem se seja sejam sejam sem serei seremos seria seriam será serão seríamos seu seus somos sou sua suas são só também te tem temos tenha tenham tenhamos tenho terei teremos teria teriam terá terão teríamos teu teus teve tinha tinham tive tivemos tiver tivera tiveram tiverem tivermos tivesse tivessem tivéramos tivéssemos tu tua tuas tá têm tínhamos um uma vai você vocês vos vou à às é éramos

Fonte: Universidade Federal do ABC

As *stopwords* são palavras das quais não fazem sentido algum quando utilizadas de formas isoladas e não representam pesos importantes para a análise de sentimentos.

2.2.2 Processamento em redes neurais

Como citado anteriormente, redes neurais não são capazes de processar textos de forma pura. Uma das formas de promover essa realização e a utilizada para a realização do projeto proposto nesse documento é transformando os *tokens* gerados em vetores que se referenciam à uma espécie de dicionário (GAAL e GHAHRAMANI, 2016).

Tratando-se de redes neurais e textos, devemos tratar os textos como uma imagem ou dados estruturados. Pensando dessa forma, ao transformarmos cada sentença em um vetor, cada um terá um tamanho diferente.

Essa estrutura não é aceita por neurônios artificiais. Para isso os vetores com tamanhos menores são preenchidos com valores iguais a zero, até ficarem do tamanho da maior instância do conjunto.

2.2.3 Camada *Embedding*

A entrada de dados em uma rede neural que processa texto primeiro será exercida pela camada *Embedding*, como proposto por Gal e Ghahmarani, que transforma cada valor inteiro dos *tokens* em outros vetores de valores de decimais, que então representarão a semântica de cada palavra, que também são ajustados ao decorrer dos treinos dos modelos.

Tabela 1 - Representação de palavras vetorizadas.

candidato	-0,40	-0,18	0,06	0,16
política	0,15	0,07	-0,02	0,22
debate	0,30	0,14	-0,05	0,02
eleitor	0,56	0,25	-0,09	0,11
partido	0,12	0,05	-0,02	-0,89
presidente	-0,78	-0,35	0,12	0,23
discurso	-0,09	-0,04	0,01	0,67
televisão	0,44	0,20	-0,07	0,87

Fonte: Gerado pelo próprio autor no Microsoft Excel.

São esses os valores, das palavras vetorizadas, como representados exemplarmente na tabela 1, que serão os dados aceitos na primeira camada de uma rede neural.

Existem outras diversas formas de realizar a vetorização de palavras, com conjuntos pré-treinados para *deep learning*, porém, apenas em língua estrangeira (CHAKRAVARTI, NAVRATIL e SANTOS, 2017), e não em língua portuguesa, a qual é o foco nesse trabalho.

3 DESENVOLVIMENTO DA APLICAÇÃO

O desenvolvimento da aplicação proposta foi feito utilizando diversas ferramentas abertas e disponíveis para a comunidade, como Keras, NLTK, *Sci-Kit Learn* e a API do *Twitter*, ambas disponíveis para a linguagem de programação *Python*.

Keras é uma API de alto nível capaz de desenvolver redes neurais com apenas algumas linhas de código para testes experimentais, sendo possível ser utilizado com Tensorflow, CNTK ou Theano em seu *backend*.

A biblioteca *Natural Language Toolkit* (NLTK) é voltada para o desenvolvimento de aplicações que trabalham com linguagem humana e processamento de linguagem natural.

Sci-Kit Learn é uma biblioteca que disponibiliza diversos algoritmos de aprendizado de máquinas prontos para serem utilizados, além de oferecer diversas funções para tratamento de dados.

O recurso disponibilizado pelo *Twitter* para desenvolvedores, possui funções disponíveis para capturar textos por filtros textuais, país e língua. Para realizar a integração com a API, foi utilizada a biblioteca *Tweepy*.

3.1 Base de dados

Encontrar uma base de dados concreta, íntegra e confiável em língua portuguesa para o desenvolvimento desse projeto foi de extrema dificuldade. Por essas razões, como forma de precingir esses problemas, foi elaborada uma base de dados classificados manualmente pelo próprio autor.

A base de dados foi desenvolvida utilizando a API do *Twitter* para capturar os textos. Possui 1043 exemplos, desses 493 com sentimentos positivos e 550 negativos. Como palavras chaves foram consideradas os nomes dos candidatos e suas possíveis variações dos que disputaram o primeiro turno das eleições presidenciais. Desse modo, essa base de dados já está inserida no contexto político.

Por se tratar de uma base de dados desbalanceada, foi utilizado uma técnica de *undersampling* (KSIENIEWICZ, 2018) para que as classes fossem balanceadas e conseqüentemente as métricas utilizadas se tornem mais confiáveis. Basicamente foi

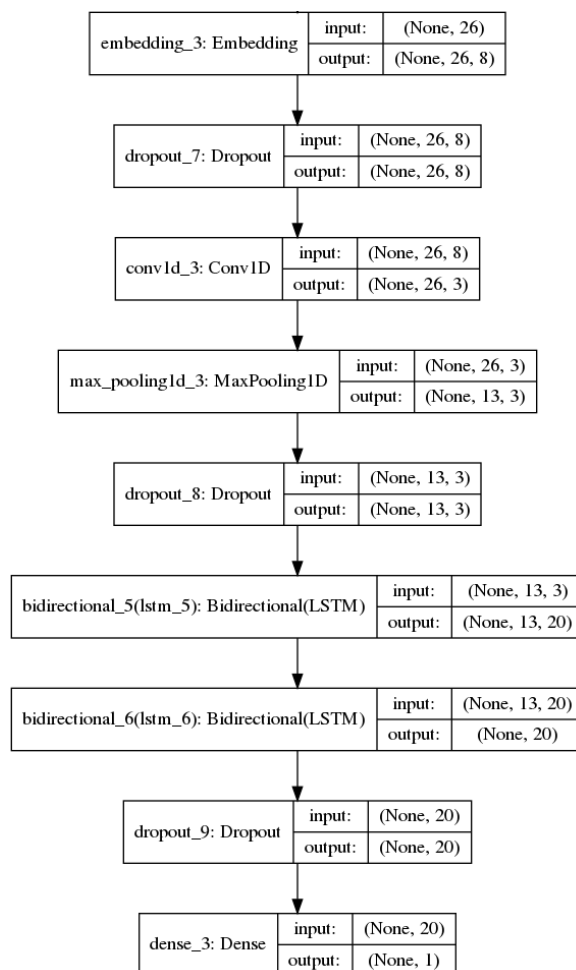
realizado o balanceamento entre as duas classes (positivo e negativo), no qual foi pego como referência a menor classe – no caso a positiva – e conseqüentemente eliminando aleatoriamente algumas instâncias negativas dessa classe até que a proporção entre elas fosse de um para um.

3.2 Modelos de redes neurais

Foi proposto duas arquiteturas distintas de redes neurais, ambas utilizando os conceitos já apresentadas no capítulo primário anterior, como: redes neurais convolucionais, recorrentes e *perceptron* multicamadas.

A grande diferença entre as duas redes neurais desenvolvidas ficam por conta do modo com o qual os dados são inseridos nas mesmas.

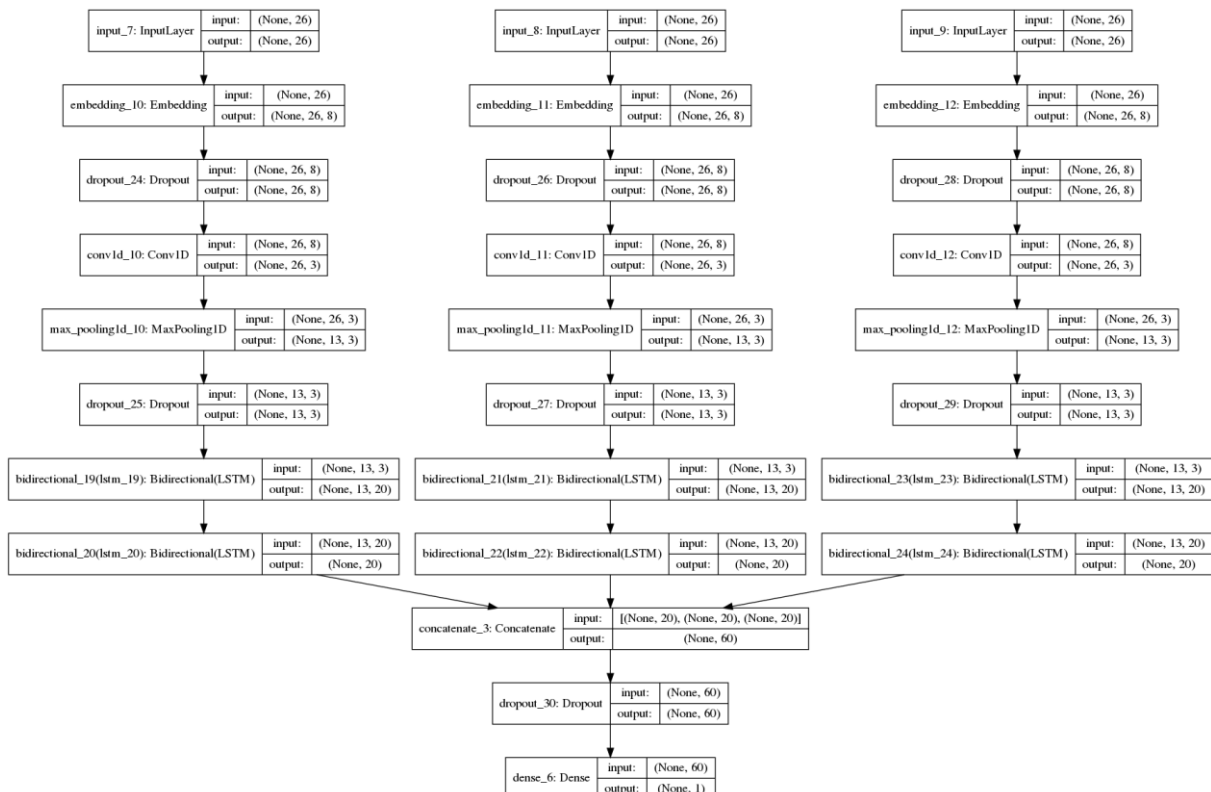
Figura 10 - Diagrama representando rede neural de um canal.



Fonte: Gerado pelo próprio autor no Keras

O primeiro modelo proposto tem como entrada o texto – já devidamente pré processado – uma única vez, já o segundo modelo possibilita a entrada do mesmo texto três vezes, em três canais diferentes. Durante o processamento na camada convolucional, cada canal processará a entrada com um *kernel* diferente de tamanhos 2, 3 e 4 cada, enquanto o modelo de canal único realiza o processo de convolução com um kernel de tamanho 3.

Figura 11 - Diagrama representando rede neural de três canais.



Fonte: Gerado pelo próprio autor no Keras

Como representados em ambos os diagramas dos modelos nas figuras 10 e 11, as entradas dos dados foram alimentadas com uma matriz de 26x8, ou seja, os textos – já vetorizados – foram todos transformados para ter um comprimento de 26 e cada *token* foi representado por um vetor de tamanho 8. Em seguida passaram por camadas convolucionais e LSTM bidirecionais, que além de receber os valores em uma sequencia padrão, também o recebem de modo invertido (trás para frente). Para o modelo de três canais, ao fim de cada canal os três são concatenados e por fim através de apenas um neurônio com a função de ativação sigmoide, tem-se como saída o valor de confiança.

Entre algumas camadas foi aplicado *dropout*, que é uma forma de ajudar na generalização dos dados (GAAL e GHAHRAMANI, 2016). Durante o treinamento, algumas conexões entre os neurônios são ignoradas, diminuindo o número de parâmetros.

3.2.1 Resultados

Para a comprovação de um modelo que generalize bem os dados que foram expostos, foram realizados três treinamentos diferentes para cada proposta. Em cada treinamento os conjuntos de treinos e de testes foram alterados aleatoriamente através da técnica estatística que chamamos de validação cruzada (ZHANG e YANG, 2015).

A validação cruzada é um método utilizado para o aprendizado de máquina supervisionado em que é medido a qualidade do modelo proposto através das diversas validações realizadas e então é calculado a média dentre elas, além do desvio padrão.

Outro teste realizado para encontrar possíveis melhorias para os modelos foi a busca pelo melhor valor para a função de ativação (Lepora, 2016), ou seja, o melhor valor para o nível de confiança de saída do modelo. Foi realizado uma busca entre os valores 0,4 e 0,6 em intervalos de 0,05.

Para cada *threshold* foram calculadas as métricas F1, recall, precisão, acurácia e desvio padrão da acurácia levando em consideração a média entre as validações cruzadas.

Tabela 2 - Métricas de cada modelo por *threshold*.

Threshold	Média de F1	Média de Recall	Média de Precisão	Média de Acurácia	Desvio de Acurácia
Canal Triplo	0,6763	0,7178	0,6649	0,6707	0,0455
0,40	0,7150	0,8721	0,6069	0,6515	0,0394
0,45	0,7102	0,8182	0,6279	0,6667	0,0231
0,50	0,6930	0,7306	0,6607	0,6785	0,0329
0,55	0,6941	0,6801	0,7120	0,7037	0,0458
0,60	0,5691	0,4882	0,7173	0,6532	0,0805

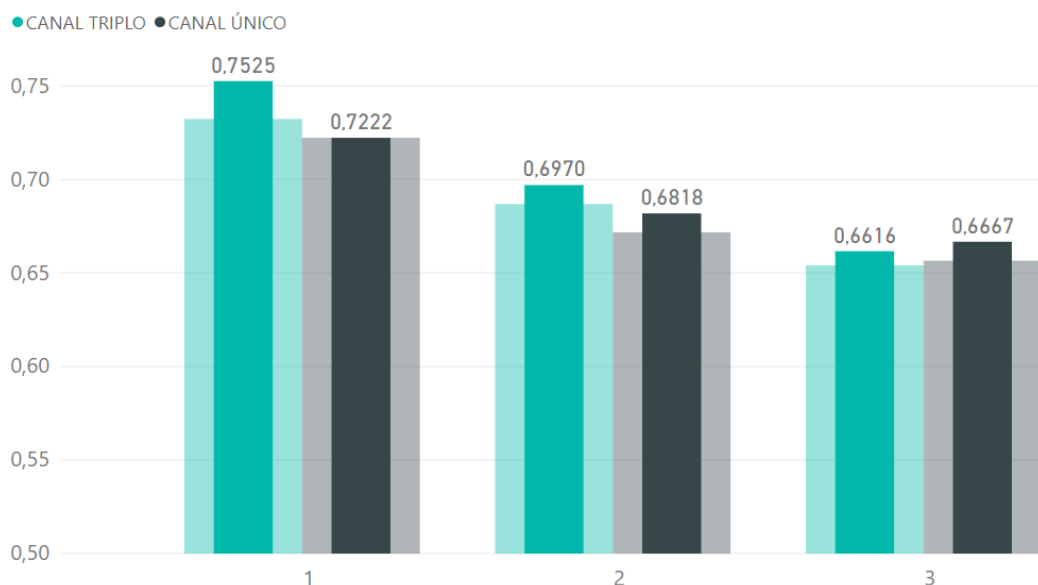
Canal Único	0,6329	0,7024	0,6104	0,6471	0,0573
0,40	0,7035	0,9125	0,5755	0,6145	0,0526
0,45	0,7215	0,8855	0,6110	0,6582	0,0239
0,50	0,7171	0,7879	0,6587	0,6902	0,0287
0,55	0,6333	0,5892	0,7410	0,6768	0,0401
0,60	0,3892	0,3367	0,4658	0,5960	0,0862

Fonte: Gerado pelo próprio autor no Microsoft Excel.

De modo geral, o modelo proposto com múltiplos canais obteve melhores resultados em todas as métricas, levando em consideração a média geral entre todos os valores de ativação considerados. O *threshold* com valor de 0,55 foi o que obteve a melhor acurácia ($70,37\% \pm 4,58$) – que é a principal métrica a ser considerada – e por isso foi o valor de ativação escolhido para realizar as predições futuras da aplicação proposta.

O modelo com apenas um canal teve seu melhor desempenho com o valor de ativação de 0,5 alcançando uma acurácia de $69,02\% \pm 2,87$. Apesar do modelo triplo ter alcançado uma acurácia superior, o com arquitetura mais simples obteve um desvio padrão mais baixo.

Figura 12 - Representação de cada modelo e seu melhor valor de threshold.



Fonte: Gerado pelo próprio autor no Microsoft Power BI.

3.3 Extração dos tweets

Os *tweets* utilizados para essa aplicação foram capturados durante os debates televisivos promovidos pelas emissoras de televisão no Brasil. Ao total foram monitorados cinco debates, das emissoras: Bandeirantes, RedeTV, SBT, Record e Globo.

Utilizando a API do *Twitter* foi escolhido alguns nomes, apelidos e palavras que poderiam remeter aos candidatos presentes nos debates. Esses nomes, demonstrados na tabela abaixo, foram escolhidos como as palavras chaves de *tweets* a serem capturados.

Tabela 3 – Palavras chaves para capturar *tweets* de cada candidato.

Candidato	Partido	Palavras remetentes
Álvaro Dias	Pode	alvaro dias, álvaro dias, partido podemos
Cabo Daciolo	Patri	cabo daciolo, daciolo, partido patriota,
Ciro Gomes	PDT	ciro, ciro gomes, cirão, cirão da massa, pdt
Fernando Haddad	PT	haddad,hadad,pt, partido dos trabalhadores
Geraldo Alckmin	PSDB	alckmin, geraldo alckmin, psdb
Guilherme Boulos	PSOL	boulos, guilherme boulos, psol
Henrique Meirelles	MDB	henrique meireles, henrique meirelles, meirelles, meireles, mdb
Jair Bolsonaro	PSL	jair bolsonaro, bolsonaro, bolsomito, psl,
João Amoêdo	Novo	amoedo, amoêdo, joão amoedo, joao amoêdo, joão amoêdo, partido novo
Marina Silva	Rede	marina, marina silva, rede, partido verde

Fonte: Gerado pelo próprio autor no Microsoft Excel.

De acordo com o artigo 46 da Lei Eleitoral no Brasil, os veículos de comunicação, seja na televisão, rádio e internet, são obrigados a convidarem todos os candidatos para os debates no qual seus respectivos partidos atendam ao requisito mínimo de cinco parlamentares no Congresso Nacional. Porém, os candidatos que forem convidados não são obrigados a participar. Os demais candidatos que não atendem ao requisito, podem ou não serem convidados, dependendo do critério de cada emissora, como, por exemplo, uma porcentagem mínima em números de intenções de votos em determinado candidato em pesquisas realizadas por alguns institutos de pesquisas no Brasil.

Todos os tweets capturados foram analisados computacionalmente para que fosse mantido apenas os tweets que fazem menção à apenas um candidato. Desse modo, textos que façam referência a dois candidatos ou mais foram descartados, garantindo assim, para a próxima etapa, uma análise de sentimentos mais íntegra e coesa para todos os candidatos. Após o descarte desses, remanesceram um total de 642.933 *tweets*.

Tabela 4 – Candidatos e suas proporções de *tweets* exclusivos.

Candidato	Tweets(qtd)	Proporção do total
Álvaro Dias	31.880	4,96%
Cabo Daciolo	114.197	17,76%
Ciro Gomes	95.855	14,91%
Fernando Haddad	57.798	8,99%
Geraldo Alckmin	30.364	4,72%
Guilherme Boulos	78.537	12,22%
Henrique Meirelles	39.124	6,09%
Jair Bolsonaro	134.442	20,91%
João Amoêdo	8.776	1,36%
Marina Silva	51.960	8,08%

Fonte: Gerado pelo próprio autor no Microsoft Excel.

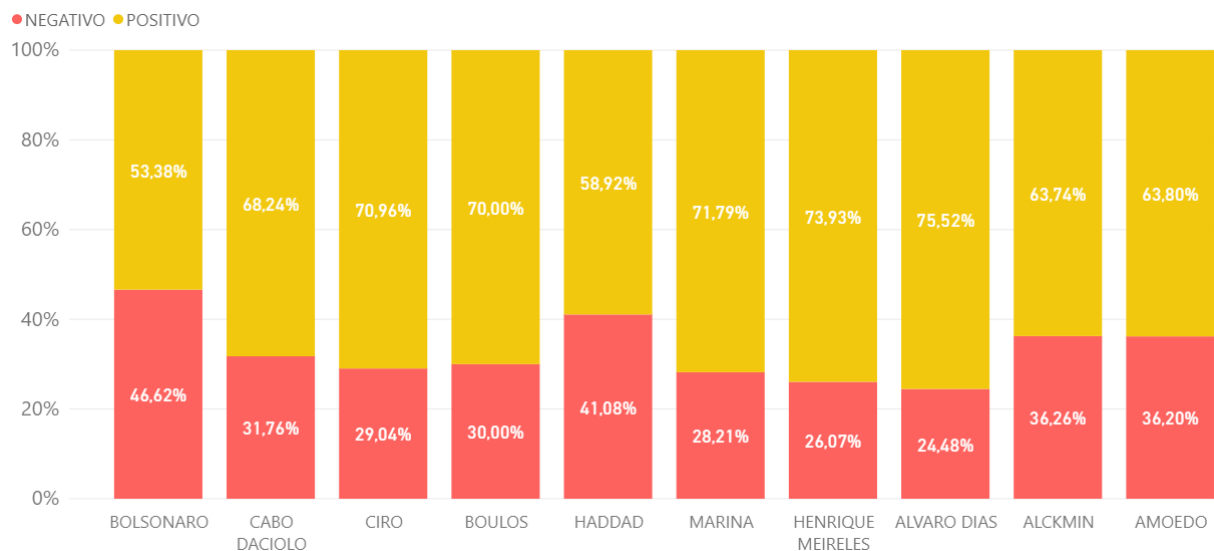
Como só houve debates no primeiro turno da eleição, não foi capturado *tweets* para o segundo. Para os resultados apresentados no capítulo seguinte serão considerados apenas os comentários no primeiro turno.

4 RESULTADOS

O modelo escolhido durante o desenvolvimento prático do capítulo anterior foi utilizado para realizar as classificações dos textos remanescentes após os descartes de acordo com os critérios já estabelecidos no subcapítulo anterior.

Ao decorrer desse capítulo será feito uma análise dos resultados geridos pela aplicação desenvolvida, além de realizar comparações com os resultados finais das eleições e pesquisas realizadas pelos institutos de pesquisa no Brasil.

Figura 13 - Taxa de comentários positivos e negativos para os candidatos no primeiro turno.



Fonte: Gerado pelo próprio autor no Microsoft Power BI.

De acordo com a pesquisa do instituto Datafolha, às vésperas das eleições presidenciais no primeiro turno, os candidatos Jair Bolsonaro e Fernando Haddad estavam com as maiores taxas de rejeição, registradas em 44% e 41% respectivamente, com uma margem de erro de 2%.

O modelo desenvolvido chegou a números próximos, como demonstrado na figura acima. O candidato Bolsonaro chegou à taxa de 44,62% de comentários classificados como negativos e o candidato Fernando Haddad à 41,08%.

Curiosamente, ambos os candidatos, que sempre apresentaram altas taxas de rejeições, nas pesquisas e na aplicação, foram os que remanesceram para o segundo turno das eleições.

No Brasil as eleições para cargos executivos acontecem em turnos. Um segundo turno só é invocado quando no primeiro turno nenhum dos candidatos ao cargo atinge a maioria absoluta dos votos. Caso um candidato alcançar a maioria absoluta dos votos válidos no primeiro turno, o mesmo será eleito.

O modelo desenvolvido conseguiu refletir parcialmente o que aconteceu nas eleições e nas pesquisas de intenção de votos, visando os candidatos que foram para o segundo turno (Bolsonaro e Haddad). Se fossemos considerar os resultados alcançados nesse trabalho, os candidatos Henrique Meirelles e Álvaro Dias seriam os remanescentes para um suposto segundo turno, com 73,93% e 75,52% de taxa de comentários positivos, respectivamente.

Como Jair Bolsonaro e Fernando Haddad foram os candidatos que chegaram ao segundo turno das eleições, será feita uma análise mais detalhada com pequenas amostras entre os *tweets* capturados que os usuários deixam habilitados a sua localização em modo público.

4.1 Regiões

Dentre os 192.240 *tweets* que referenciam a Bolsonaro e Haddad, apenas de 3.679 é possível acessar a localização do usuário. E é dessa amostra que será apresentado os resultados por regiões do país.

Tabela 5 – Regiões e suas proporções da amostra.

Região	Tweets(qtd)	Proporção da amostra
Centro Oeste	270	7,34%
Nordeste	773	21,01%
Norte	226	6,14%
Sudeste	1998	54,31%
Sul	640	17,40%

Fonte: Gerado pelo próprio autor no Microsoft Excel.

Considerando a quantidade de *tweets* dessa amostra para cada candidato, Bolsonaro lidera em todos os estados mesmo nos que ele teve uma taxa de comentários positivos menor que de seu opositor.

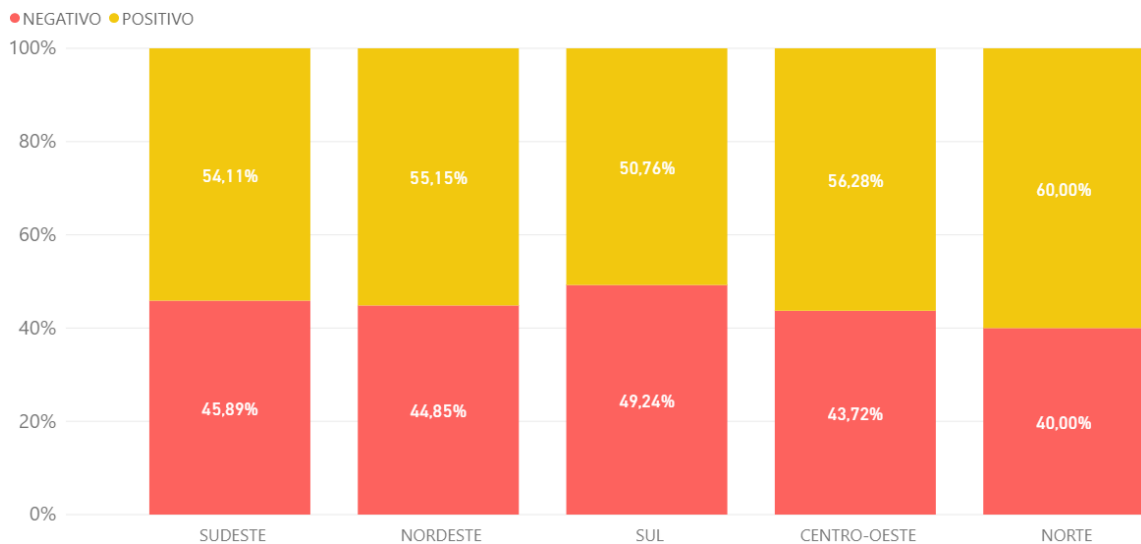
Figura 14 - Mapa do Brasil representando a proporção entre tweets dirigidos aos candidatos no segundo turno por estado.



Fonte: Gerado pelo próprio autor no Microsoft Power BI.

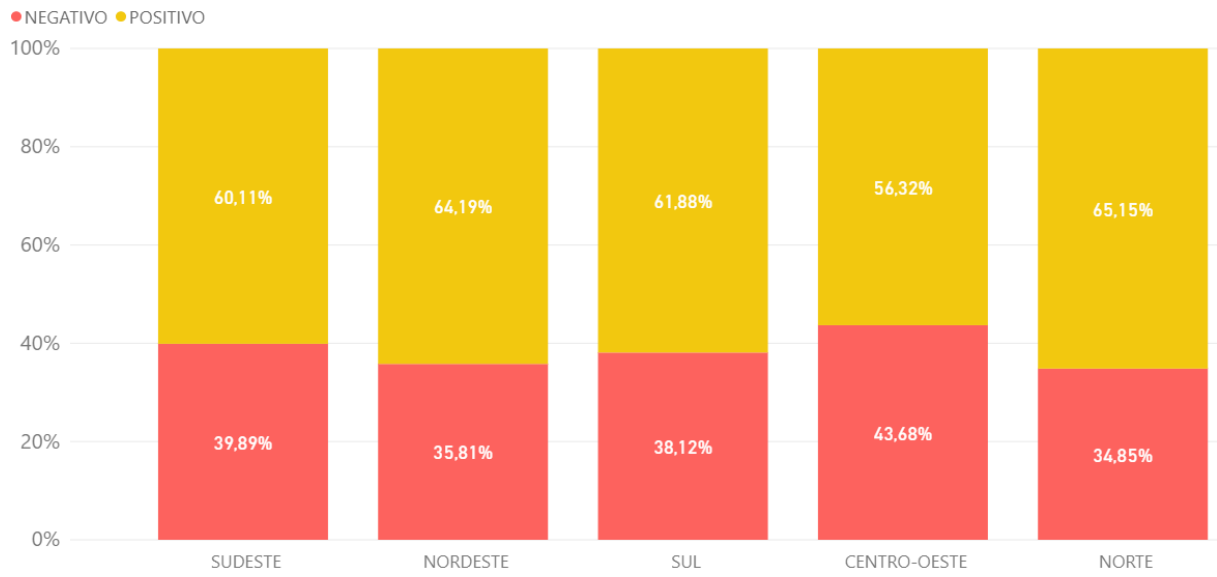
Isso acaba demonstrando sua alta popularidade em toda o território nacional, seja positivamente ou negativamente. Pois, apesar de ter sido eleito o presidente da república para 2019 com 55,13% dos votos válidos, ele também sempre liderou as pesquisas de rejeição. Além disso, um dos principais resultados desse trabalho foi também ter chego a números semelhantes, no qual o mesmo também ficou na primeira colocação em taxa de comentários negativos.

Figura 15 - Taxa de comentários positivos e negativos do candidato Bolsonaro.



Fonte: Gerado pelo próprio autor no Microsoft Power BI.

Figura 16 - Taxa de comentários positivos e negativos do candidato Haddad.



Ao considerarmos as amostras de cada região, Fernando Haddad lidera a taxa de comentários positivos em todas, o que não foi refletido nas urnas. Bolsonaro por outro lado lidera em todas as regiões as taxas de comentários negativos, porém, foi quem ganhou nas votações oficiais.

5 CONSIDERAÇÕES FINAIS

As opiniões disseminadas pelos usuários nas redes sociais, mesmo que não tenham sido refletidas fielmente o resultado final nas urnas, são de extrema importância e valor para as equipes políticas.

Deve-se levar em consideração que nem todos os usuários possuem redes sociais e que dessa quantia apenas alguns se expressam politicamente em público, ainda mais se considerarmos a rede social utilizada, o Twitter. Esse trabalho desenvolvido pode ser ampliado futuramente para incorporar diversas outras redes sociais fazendo o uso do mesmo conceito.

O trabalho em si conseguiu concluir seus objetivos estabelecidos inicialmente mesmo com a principal adversidade encontrada que foi encontrar uma base de dados extensa e confiável. Além das colaborações para os setores da computação e políticos, o projeto construiu uma base de dados extensa e contextualizada na política, que está pronta para ser classificada para um aprendizado supervisionado no futuro, visando ainda mais confiança nos dados e métricas, além de uma generalização superior quanto ao aprendizado.

Os modelos de redes neurais propostos obtiverem bons resultados e generalização frente a base de dados escassa e ainda demonstrou confiança quanto a utilização de redes neurais híbridas para a análise de sentimentos.

Considerando o trabalho e os resultados obtidos, as redes sociais conseguem representar parcialmente a opinião dos eleitores brasileiros e não devem ser substituídas pelas pesquisas de intenções de votos realizadas pelos institutos de pesquisas. Porém, mostrou-se ser uma ferramenta essencial para entender o comportamento de uma amostra dos eleitores digitais em época eleições.

REFERÊNCIAS BIBLIOGRÁFICAS

AIN, Q. T. Sentiment Analysis Using Deep Learning. **International Journal of Advanced Computer Science and Applications**, v. 8, n. 6, p. 424-433, 2017.

BERGSMA, S.; ETAL. **Machine Learning Approaches to Sentiment Classification**. Department of Computing Science, University of Alberta. [S.l.], p. 12-26. 2005.

CARVALHO, C. M.; NAGANO, H.; BARROS, A.. A Comparative Study for Sentiment Analysis on Election, Uberlândia, Proceedings of Symposium in Information and Human Language Technology Outubro 2017. 103-111.

CHAKRAVARTI, R.; NAVRATIL, J.; DOS SANTOS, C.. Improved Answer Selection with Pre-Trained Word Embeddings. **Cornell University Library**, 2017.

CHEN, G. **A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation**. Department of Computer Science and Engineering, WASEDA. Buffalo, p. 4-6. 2018.

GAAL, Y.; GHAMRANI, Z. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. **Neural Information Processing Systems**, 2016. Disponível em: <<https://arxiv.org/abs/1512.05287>>.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. Disponível em: <<http://www.deeplearningbook.org/>>.

GREFF, K.; ETAL. LSTM: A Search Space Odyssey. **Transactions on Neural Networks and Learning System**, Outubro 2017. Disponível em: <<https://arxiv.org/pdf/1503.04069.pdf>>.

GURNEY, K. **An introduction to neural networks**. Londres: UCL Press Limited, 1997.

HOCHREITER, S.; SCHMIDHUBER,. Long Short-Term Memory. **Neural Computation** , Munique, v. 9, n. 8, p. 1735-1780, 1997.

KSIENIEWICZ, P. Undersampled Majority Class Ensemble for highly imbalanced binary classification. **Proceedings of Machine Learning Research**, 94, 2018. 82-94.

LECUN, Y. Backpropagation Applied to Handwritten Zip Code Recognition. **Neural Computation**, Holmdel, v. 1, p. 541-551, Setembro 1989.

LEPORA, N. Threshold Learning for Optimal Decision Making. **Neural Information Processing Systems**, 2016.

MAAS, A.; ETAL. Learning Word Vectors for Sentiment Analysis. **Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies**, Junho 2011. 142-150.

MAKIN, J. G. Backpropagation. **Cornell University Library**, Fevereiro 2006. Disponível em: <<http://www.cs.cornell.edu/courses/cs5740/2016sp/resources/backprop.pdf>>.

MCCULLOCH, W.; PITTS, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. **Bulletin of Mathematical Biophysics**, v. 5, p. 115-133, 1943.

MENA-CHALCO, J. **Normalização de texto: Palavras e stopwords**. Universidade Federal do ABC. [S.l.], p. 30.

PAK, A.; PAROUBEK, P. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. **Proceedings of the Seventh conference on International Language Resources and Evaluation**, 2010. 1320-1326.

ROSENBLATT, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. **Psychological Review**, v. 65, n. 6, p. 386-408, Abril 1958.

RUMELHART, D.; HINTON, G.; WILLIAMS, R. Learning Representations by Back-propagating Errors. **Nature**, São Diego, v. 323, p. 533-536, Outubro 1986.

SALUNKHE, P.; SURNAR, ; SONAWANE,. A Review: Prediction of Election Using Twitter. **International Journal of Advanced Research in Computer Engineering & Technology**, 6, 5 Maio 2017. 723-725.

SOVIERZOSKI, M. A. Convolução de sinais: definição, propriedades e ferramentas. **Revista Ilha Digital - Instituto Federal Santa Catarina**, Florianópolis, 2, 2010. 81-95.

THOMA, M.; ETAL. **Analysis and Optimization of Convolutional Neural Network Architectures**. Karlsruhe Institute of Technology. [S.l.], p. 5-8. 2017.

YIN, W. et al. Comparative Study of CNN and RNN for Natural Language Processing. **Cornell University Library**, Ithaca, 7 Fevereiro 2017. Disponível em: <<https://arxiv.org/abs/1702.01923>>.

ZHANG, Y.; YANG, Y. Cross-Validation for Selecting a Model Selection Procedure. **Journal of Econometrics**, 187, 2015. 95-112.

ZHANG, Z. Derivation of Backpropagation in Convolutional Neural Network (CNN). **University of Tennessee**, p. 7, Outubro 2016. Disponível em: <<https://pdfs.semanticscholar.org/5d79/11c93ddcb34cac088d99bd0cae9124e5dcd1.pdf>>.

ZHOU, C.; ETAL. A C-LSTM Neural Network for Text Classification. **Cornell University Library**, Novembro 2015.