



**CARLOS LEVI DA SILVA ALBUQUERQUE**

**ANÁLISE DE SENTIMENTO SOBRE COMENTÁRIOS EM SITES DE E-COMMERCE  
NO IDIOMA PORTUGUÊS/BR**

**FORTALEZA**

**2022**

CARLOS LEVI DA SILVA ALBUQUERQUE

ANÁLISE DE SENTIMENTO SOBRE COMENTÁRIOS EM SITES DE E-COMMERCE NO  
IDIOMA PORTUGUÊS/BR

Trabalho de Conclusão de Curso (TCC) apresentado ao curso de Sistemas de Informação do Centro Universitário Christus, como requisito parcial para obtenção do grau de bacharel em Sistemas de Informação.

Orientador: Prof. Me. Felipe Timbó Brito

FORTALEZA

2022

Dados Internacionais de Catalogação na Publicação  
Centro Universitário Christus - Unichristus  
Gerada automaticamente pelo Sistema de Elaboração de Ficha Catalográfica do  
Centro Universitário Christus - Unichristus, com dados fornecidos pelo(a) autor(a)

Sa Silva Albuquerque, Carlos Levi.  
ANÁLISE DE SENTIMENTO SOBRE COMENTÁRIOS EM  
SITES DE E-COMMERCE NO IDIOMA PORTUGUÊS/BR / Carlos  
Levi Silva Albuquerque. - 2022.  
43 f. : il. color.

Trabalho de Conclusão de Curso (Graduação) - Centro  
Universitário Christus - Unichristus, Curso de Sistemas de  
Informação, Fortaleza, 2022.  
Orientação: Prof. Me. Felipe Timbó Brito.

1. Análise de sentimento. 2. E-commerce. 3. Processamento de  
linguagem natural. I. Título.

CDD 005

CARLOS LEVI DA SILVA ALBUQUERQUE

ANÁLISE DE SENTIMENTO SOBRE COMENTÁRIOS EM SITES DE E-COMMERCE NO  
IDIOMA PORTUGUÊS/BR

Trabalho de Conclusão de Curso (TCC) apresentado ao curso de Sistemas de Informação do Centro Universitário Christus, como requisito parcial para obtenção do grau de bacharel em Sistemas de Informação.

Aprovada em:

BANCA EXAMINADORA

---

Prof. Me. Felipe Timbó Brito (Orientador)  
Centro Universitário Christus (Unichristus)

---

Prof. Dr. Daniel Nascimento Teixeira  
Centro Universitário Christus (Unichristus)

---

Prof. Me. Iago Castro Chaves  
Centro Universitário Christus (Unichristus)

## **AGRADECIMENTOS**

Primeiramente agradeço a Deus pela vida que Ele me concedeu e por sempre mostrar o caminho certo.

Ao professor Me Felipe Timbó Brito, por ter sido meu orientador e ter desempenhado tal função com grande atenção e dedicação, o qual tornou-se essencial para que o projeto fosse concluído.

Aos meus pais que sempre estiveram ao meu lado me apoiando ao longo de toda a minha trajetória. Também aos meus irmãos pela amizade e atenção dedicadas quando sempre precisei.

Por fim, agradeço os professores pelas correções e ensinamentos, as quais me permitiram apresentar um melhor desempenho no meu processo de formação profissional ao longo do curso. Também por todos os conselhos, pela ajuda e pela paciência com a qual guiaram o meu aprendizado.

"Que nada nos defina, que nada nos sujeite. Que a liberdade seja a nossa própria substância."

(Simone de Beauvoir)

## RESUMO

O mundo tem observado a popularização do *e-commerce* na última década. Um dos motivos para isso é a comodidade que estas plataformas fornecem, permitindo que consumidores realizem compras sem a necessidade de saírem de suas residências. Nesse contexto, usuários de plataformas de *e-commerce* são constantemente incentivados a compartilharem suas opiniões e sentimentos de forma online, uma vez que as avaliações de produtos podem influenciar nas decisões de compra por parte dos consumidores e, ao mesmo tempo, ajudar as empresas a avaliarem os seus processos. A análise manual, ou sem o auxílio de técnicas específicas, para estas finalidades podem acarretar desvantagens competitiva, visto a grande quantidade e diversidade de comentários existentes em plataformas de *e-commerce*. Dessa forma, este trabalho tem como objetivo desenvolver um modelo de processamento de linguagem natural para análise de sentimento sobre comentários de plataformas de *e-commerce*, de maneira automática e no idioma português/BR. Os resultados obtidos através de uma análise experimental mostram que o modelo proposto fornece uma acurácia acima de 85% para todos os conjuntos de dados testados.

**Palavras-chave:** *E-Commerce*. Análise de sentimento. Processamento de Linguagem Natural. Português/BR.

## ABSTRACT

The world has observed the popularization of e-commerce in the last decade. One of the reasons for this is the convenience that these platforms provide, allowing clients to make purchases without having to leave their homes. In this context, users of e-commerce platforms are constantly encouraged to share their opinions and feelings online, since product reviews can influence purchase decisions by consumers and, at the same time, help companies evaluate their processes. Manual analysis, or without the aid of specific techniques, for these purposes can result in a competitive disadvantage, given the large number and diversity of comments existing on e-commerce platforms. Thus, this work aims to develop a natural language processing model for sentiment analysis in comments on e-commerce platforms, automatically and in Portuguese/BR. Experimental analysis results show that the proposed model provides an accuracy above 85% for all tested datasets.

**Keywords:** E-Commerce. Sentiment analysis. Natural Language Processing. Portuguese/BR.

## LISTA DE FIGURAS

Figura 1 – Processo de análise de sentimentos em <i>reviews</i> de produtos. . . . .	16
Figura 2 – Processo de aprendizado de máquina supervisionado. . . . .	17
Figura 3 – Rede neural recorrente. . . . .	18
Figura 4 – Rede neural recorrente expandido. . . . .	19
Figura 5 – Arquitetura original de um LSTM. . . . .	21
Figura 6 – Arquitetura original de um LSTM. . . . .	21
Figura 7 – Exemplo de uma matriz de confusão de duas classes. . . . .	23
Figura 8 – Arquitetura do sistema proposto por (JABBAR <i>et al.</i> , 2019). . . . .	26
Figura 9 – Arquitetura do sistema proposto por (CHAVES <i>et al.</i> , 2022). . . . .	27
Figura 10 – Arquitetura do sistema proposto por (KHAN <i>et al.</i> , 2022). . . . .	28
Figura 11 – Fluxo de pré-processamento. . . . .	30
Figura 12 – Fluxo do pré-processamento. . . . .	34
Figura 13 – Matriz de confusão da base de dados B2W. . . . .	37
Figura 14 – Matriz de confusão da base de dados Buscape. . . . .	38
Figura 15 – Matriz de confusão da base de dados Olist. . . . .	39
Figura 16 – Matriz de confusão da base de dados Concat. . . . .	40

## LISTA DE TABELAS

Tabela 1 – Distribuição dos dados por base de dados. . . . .	30
Tabela 2 – Acurácia dos três bases de dados analisadas . . . . .	37
Tabela 3 – Métricas B2W . . . . .	37
Tabela 4 – Métricas Buscape . . . . .	38
Tabela 5 – Métricas Olist . . . . .	39
Tabela 6 – Métricas <i>Concat</i> . . . . .	40

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	12
<b>1.1</b>	<b>Contextualização e delimitação do tema</b>	12
<b>1.2</b>	<b>Problematização</b>	12
<b>1.3</b>	<b>Pressupostos</b>	13
<b>1.4</b>	<b>Objetivos</b>	13
1.4.1	<i>Objetivo geral</i>	13
1.4.2	<i>Objetivos específicos</i>	13
<b>1.5</b>	<b>Justificativa</b>	14
<b>1.6</b>	<b>Estrutura do trabalho</b>	14
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	15
<b>2.1</b>	<b>Análise de sentimentos</b>	15
<b>2.2</b>	<b>Aprendizado de máquina supervisionado</b>	16
<b>2.3</b>	<b>NLP</b>	17
<b>2.4</b>	<b>Redes neurais recorrentes</b>	18
<b>2.5</b>	<b>LSTM</b>	20
<b>2.6</b>	<b>Função de ativação</b>	22
<b>2.7</b>	<i>Loss function</i>	22
<b>2.8</b>	<b>Otimizadores</b>	23
<b>2.9</b>	<b>Métricas de avaliação</b>	23
<b>3</b>	<b>TRABALHOS CORRELATOS</b>	26
<b>4</b>	<b>METODOLOGIA</b>	29
<b>4.1</b>	<b>Dados utilizados</b>	29
<b>4.2</b>	<b>Pré-processamento</b>	30
4.2.1	<i>Remoção de elementos NaN</i>	30
4.2.2	<i>Transformação do texto em minúsculo</i>	31
4.2.3	<i>Remoção de pontuação</i>	31
4.2.4	<i>Remoção de stopwords e números</i>	31
4.2.5	<i>Lematização</i>	32
4.2.6	<i>Remoção de acentuação</i>	32
4.2.7	<i>Tokenização</i>	33

<b>4.3</b>	<b>Construção do modelo</b>	33
<i>4.3.1</i>	<i>Camadas do modelo</i>	34
<b>5</b>	<b>RESULTADOS</b>	36
<b>5.1</b>	<b>Especificações do ambiente</b>	36
<b>5.2</b>	<b>Apresentação de métricas e análise de resultados</b>	36
<i>5.2.1</i>	<i>Resultados obtidos</i>	37
<i>5.2.1.1</i>	<i>B2W</i>	37
<i>5.2.1.2</i>	<i>Buscape</i>	38
<i>5.2.1.3</i>	<i>Olist</i>	39
<i>5.2.1.4</i>	<i>Concat</i>	40
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	41
	<b>REFERÊNCIAS</b>	42

# 1 INTRODUÇÃO

O conceito de *e-commerce* está relacionado às noções de economia da Internet e economia digital. Todos esses conceitos dizem respeito ao uso das novas tecnologias de informação e comunicação para atividades econômicas, mas com focos diferentes. A economia da Internet refere-se às atividades econômicas que geram receita a partir da Internet ou de produtos e serviços relacionados à Internet (COSTA, 2003). De forma direta, *e-commerce* ou comércio eletrônico, também conhecido como *e-business*, refere-se à transação de bens e serviços por meio de comunicações eletrônicas. Embora o público em geral tenha se familiarizado com o comércio eletrônico apenas na última década, ele existe há mais de 30 anos (TIAN; STEWART, 2006).

## 1.1 Contextualização e delimitação do tema

Plataformas de *e-commerce* permitem que os clientes possam optar por comprar produtos de forma *on-line* sem o uso de certas experiências sensoriais, como tocar, cheirar ou provar. Além disso, usuários dessas plataformas são constantemente incentivados a compartilhar suas opiniões e sentimentos para diversas finalidades. Tal fato pode influenciar nas decisões de compra por parte dos consumidores, ajudar as empresas a avaliarem de seus processos ou mesmo auxiliar na implantação estratégias de *marketing online* (ULLAL *et al.*, 2021).

Grandes empresas de *e-commerce*, tais como a *Amazon*, oferecem diversas formas para que os consumidores compartilhem suas opiniões a respeito de produtos ou serviços adquiridos (SISTA *et al.*, 2021). Para analisar esses comentários, pode-se utilizar a técnica de análise de sentimentos, que consiste no processo de analisar um texto digital para determinar se o tom emocional da mensagem é positivo ou negativo. Essa análise tem o intuito de identificar como os usuários dessas plataformas se sentem em relação aos serviços fornecidos.

## 1.2 Problematização

Com a proliferação dos *reviews* sobre uma determinada marca ou serviço, a imagem das empresas está cada vez mais exposta nas redes. Por esse motivo, elas estão cada vez mais interessadas em automatizar o processo de entendimento sobre o que está sendo falado acerca delas. Além disso, geralmente *reviews* são produzidos em grandes quantidades, o que dificulta a análise de sentimentos de forma manual ou sem o auxílio de técnicas específicas

para essa finalidade. Dessa forma, é importante analisar, de maneira automática, quais opiniões e sentimentos estão sendo expressos em sites de vendas *online*, com o intuito de entender as expectativas e interesses dos consumidores.

### 1.3 Pressupostos

NLP (*Natural Language Processing*), *Data Mining*, *Text Analysis*, *Machine Learning*, e *Deep Learning*, são subáreas da ciência da computação e inteligência artificial que são utilizadas com o intuito de analisar os sentimentos a partir do *feedback* compartilhado pelos usuários (XING *et al.*, 2019; ZHANG *et al.*, 2019). Assim, este trabalho tem como pressuposto que a utilização dessas técnicas pode facilitar a análise de sentimentos de maneira automática, especificamente em sites de vendas online e serviços de e-commerce.

### 1.4 Objetivos

O objetivo geral e os objetivos específicos deste trabalhos são descritos a seguir.

#### 1.4.1 *Objetivo geral*

Este trabalho tem como objetivo principal desenvolver um modelo de NLP (*Natural Language Processing*) para analisar de forma automática sentimentos contidos em comentários de usuários em plataformas de *e-commerce*.

#### 1.4.2 *Objetivos específicos*

Em particular, os seguintes objetivos específicos são relacionados:

1. Realizar o pré-processamento de conjuntos de comentários e *reviews* em plataformas de vendas *online*, com o intuito de preparar os dados como entrada para o modelo que será desenvolvido.
2. Construir um modelo de NLP (*Natural language processing*) utilizando LSTM (*Long short-term memory*) para identificar sentimentos positivos ou negativos de maneira automática em comentários e *reviews* de plataformas de *e-commerce*.
3. Analisar, de forma experimental, métricas de desempenho obtidas após o treinamento do modelo com base nos dados analisados.

## 1.5 Justificativa

Com o contínuo e rápido desenvolvimento da internet, lojas *online* se tornaram uma das principais formas do consumidor comprar e consumir produtos e serviços. Analisar o sentimento em um grande número de *reviews* de usuários nestas plataformas pode ajudar a melhorar a satisfação destes usuários (YANG *et al.*, 2020). Além disso, analisar a tendência de sentimento a partir das avaliações dos consumidores pode não apenas fornecer uma referência para outros consumidores, mas também ajudar as empresas em plataformas de comércio eletrônico a melhorar a qualidade do serviço e a satisfação do consumidor (YANG *et al.*, 2020). Dito isso, este trabalho se mostra relevante pelo fato de contribuir com os estudos na área de análise de sentimento, em particular sobre sites de vendas e plataformas de *e-commerce* no idioma português/BR.

## 1.6 Estrutura do trabalho

Os capítulos seguintes estão divididos da seguinte forma: o capítulo 2 apresenta o referencial teórico base para o desenvolvimento deste trabalho. Já no capítulo 3 é apresentado um conjunto de trabalhos correlatos, onde são descritas abordagens similares a este trabalho. No capítulo 4 é apresentada a metodologia adotada para o desenvolvimento da técnica proposta de análise de sentimento sobre dados de vendas e *e-commerce*, visando descrever de forma detalhada os processos realizados. O capítulo 5 consiste em apresentar os resultados obtidos a partir da técnica proposta em três bases de dados distintas.

## 2 REFERENCIAL TEÓRICO

Nesta seção serão apresentados os conceitos e definições base para a elaboração deste trabalho.

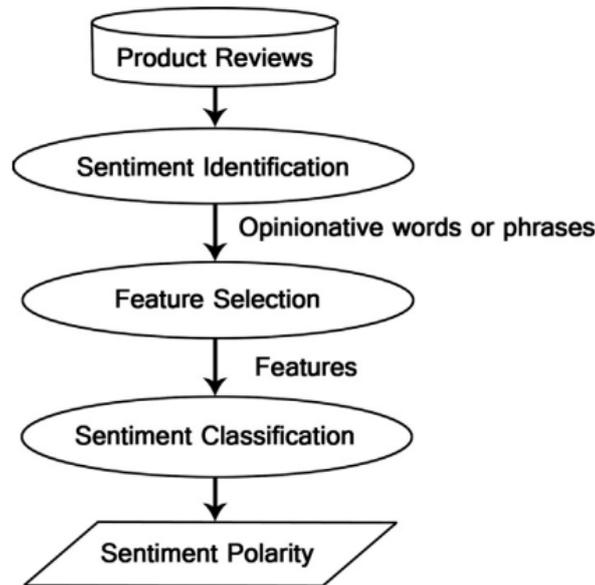
### 2.1 Análise de sentimentos

As opiniões e sentimentos são essenciais para quase todas as atividades humanas e são os principais influenciadores de nossos comportamentos. Nossas crenças, percepções da realidade e as escolhas que fazemos, são, em um grau considerável, condicionados há como os outros veem e avaliam o mundo. Por isso, quando precisamos tomar uma certa decisão, muitas vezes procuramos as opiniões dos outros. Isso não é verdade apenas para indivíduos, mas também para as organizações. Nesse contexto, a análise de sentimentos é o campo de estudo que analisa as opiniões, sentimentos, avaliações, apreciações, atitudes e emoções em relação a entidades, como produtos, serviços, organizações, indivíduos, eventos e seus atributos (LIU, 2012).

Embora a linguística e o processamento de linguagem natural (NLP) tenham uma longa história, pouca pesquisa foi feita sobre as opiniões e sentimentos das pessoas antes do ano 2000. Desde então, este campo se tornou uma área de pesquisa muito ativa. Há várias razões para isso. Em primeiro lugar, tem-se uma ampla gama de aplicações, quase que em todos os domínios. A indústria em torno da análise de sentimento também floresceu devido à proliferação de aplicações comerciais. Isso fornece uma boa motivação para a pesquisa. Em segundo lugar, este campo oferece muitos problemas de pesquisa desafiadores, que nunca foram estudados (LIU, 2012).

No contexto de aprendizado de máquina, análise de sentimentos pode ser considerada como um processo de classificação (MEDHAT *et al.*, 2014) conforme ilustrado na Figura 1, onde é proposta a aplicação da análise de sentimentos em *reviews* e comentários em plataformas de *e-commerce*.

Figura 1 – Processo de análise de sentimentos em *reviews* de produtos.



Fonte: (MEDHAT *et al.*, 2014)

Análise de sentimentos não é aplicada somente para o contexto de produtos, mas também em outras áreas como bolsas de valores (YU *et al.*, 2013; HAGENAU *et al.*, 2013), notícias (XU *et al.*, 2012), debates políticos (MAKS; VOSSEN, 2012), entre outros.

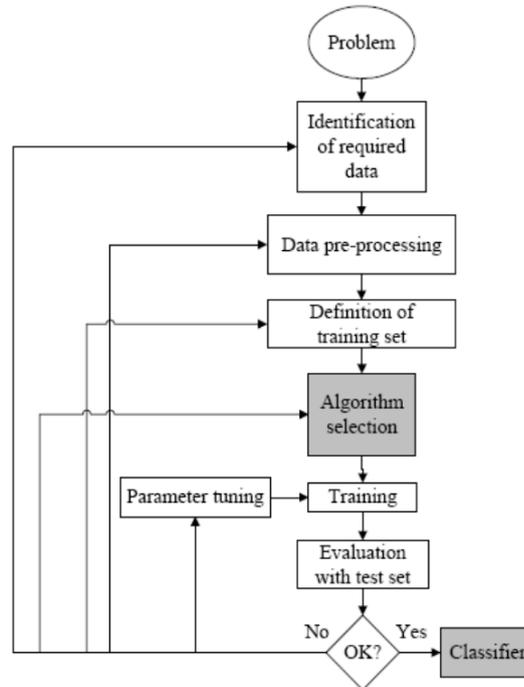
## 2.2 Aprendizado de máquina supervisionado

Aprendizado de máquina é um subcampo da inteligência artificial que estuda tipos específicos de algoritmos que são capazes de melhorar com base em experiências (MITCHELL; MITCHELL, 1997). Para esses algoritmos, o tipo de *feedback* disponível para o aprendizado geralmente é o fator mais importante na determinação da natureza do problema de aprendizado. Nesse contexto, o aprendizado de máquina geralmente se distingue em três casos: aprendizado supervisionado, não supervisionado e por reforço. Mais especificamente, o problema de aprendizado supervisionado envolve aprender uma função a partir de exemplos de suas entradas e saídas. Além disso, os dados de treinamento devem estar rotulados de alguma forma. Um exemplo de dado rotulado é o rótulo categórico de *spam* em uma mensagem SMS (*Short Message Service*). Outro exemplo é o valor quantitativo para o número de curtidas de um *tweet* (RUSSELL *et al.*, 2003).

O aprendizado supervisionado é o que a maioria das pessoas pensam quando se fala sobre aprendizado de máquina, no qual o modelo só pode melhorar se puder medir a diferença entre a saída esperada (o rótulo) e suas previsões (LANE *et al.*, 2019), conforme exemplificado

na Figura 2. Um exemplo clássico de aprendizado supervisionado é o problema de classificação, no qual um algoritmo deve aprender uma função que mapeia os objetos de um conjunto em uma de suas classes (OSISANWO *et al.*, 2017).

Figura 2 – Processo de aprendizado de máquina supervisionado.



Fonte: (OSISANWO *et al.*, 2017)

### 2.3 NLP

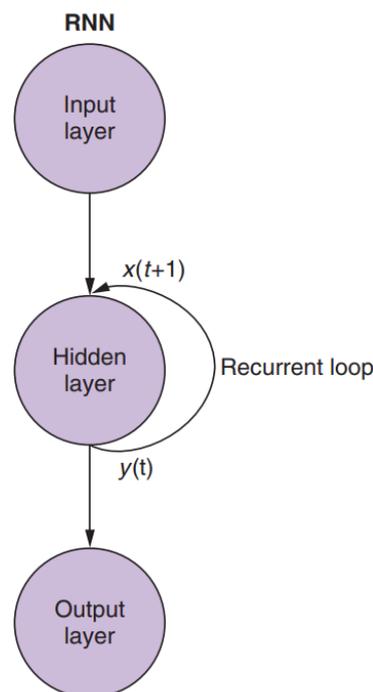
NLP (*Natural Language Processing*) consiste em um conjunto de técnicas computacionais utilizadas para automatizar as análises e representações da linguagem humana (JOHRI *et al.*, 2021). Processar a linguagem natural para extrair informações úteis pode ser dispendioso. Contudo, técnicas computacionais foram desenvolvidas para facilitar esses processos. Estas técnicas, por mais que não realizem algumas tarefas mais práticas de NLP (como conversação e compreensão de leitura) de forma tão precisa quanto humanos, em alguns casos elas podem superar tanto a precisão quanto a velocidade em comparação com humanos. Por exemplo, o reconhecimento de sarcasmo em uma mensagem isolada do *Twitter* pode ser feito com mais precisão por uma máquina do que por um ser humano. Embora humanos ainda sejam melhores em reconhecer sentimentos dentro de um diálogo permanente, devido à nossa capacidade em reter informações sobre o contexto de uma declaração, as máquinas estão ficando cada vez melhores em manter o contexto (LANE *et al.*, 2019).

## 2.4 Redes neurais recorrentes

Redes neurais representam uma grande classe de modelos computacionais que tem como base a estrutura biológica dos módulos cerebrais. Em uma rede neural artificial, vários neurônios abstratos (também chamados de unidades ou elementos de processamento) são interconectados por conexões sinápticas (ou *links*) igualmente abstraídas, que permitem que as ativações se propaguem pela rede (LUKOŠEVIČIUS; JAEGER, 2009).

Dentre as redes neurais artificiais, as redes neurais recorrentes (RNNs) diferem das outras na capacidade de processar dados temporais (dados em sequência) como sentenças, onde a rede é capaz de lembrar as palavras passadas dentro de uma frase. Como é possível observar na Figura 3, um único neurônio recorrente na camada oculta adiciona um *loop* recorrente para “reciclar” a saída da camada oculta no tempo ( $t$ ). A saída no tempo ( $t$ ) é adicionada à próxima entrada no tempo ( $t+1$ ). Esta nova entrada é processada pela rede no intervalo de tempo ( $t+1$ ) para criar a saída para essa camada oculta no tempo ( $t+1$ ). Essa saída em ( $t+1$ ) é então reciclada novamente para a entrada no passo de tempo ( $t+2$ ), e assim por diante (LANE *et al.*, 2019).

Figura 3 – Rede neural recorrente.

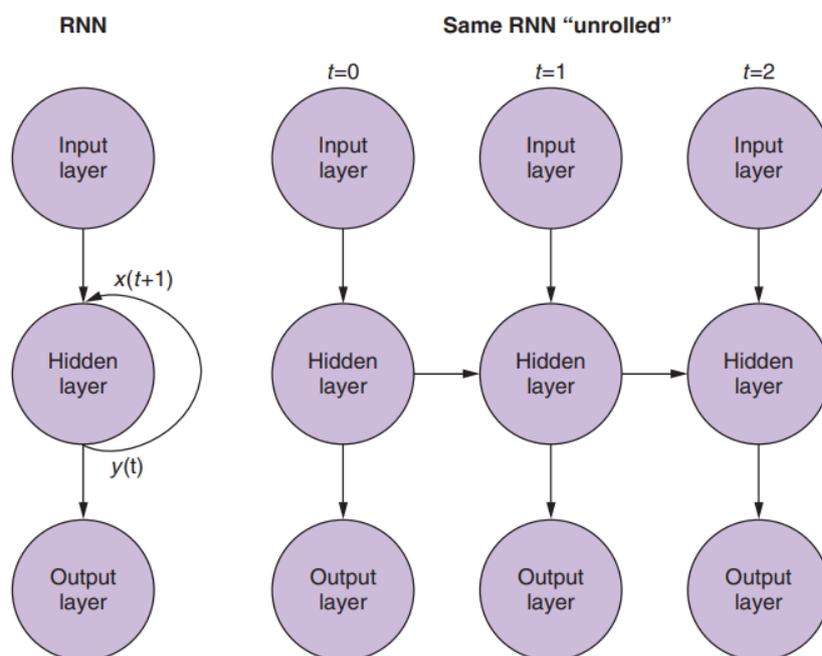


Fonte: (LANE *et al.*, 2019)

Embora a ideia de afetar o estado ao longo do tempo possa ser um pouco confusa no começo, o conceito básico é simples. Para cada entrada alimentada em uma rede *feedforward*

regular (onde todas as camadas processam os dados de entrada em uma direção uniforme), a saída da rede no intervalo de tempo ( $t$ ) é obtida e fornecida como uma entrada adicional, junto com o próximo dado sendo alimentado na rede no intervalo de tempo ( $t+1$ ). Um exemplo de rede recorrente é ilustrado na Figura 3. Os círculos são camadas de rede *feedforward* inteiras compostas por um ou mais neurônios. A saída da camada oculta emerge da rede normalmente, mas também é reservada para ser passada de volta como uma entrada para si mesma juntamente com a entrada normal do próximo intervalo de tempo. Este *feedback* é representado com um arco da saída de uma camada de volta para sua própria entrada. Uma maneira mais comum de visualizar esse processo é expandindo a rede. A Figura 4 mostra a rede invertida com dois desdobramentos da variável de tempo ( $t$ ), mostrando camadas para ( $t+1$ ) e ( $t+2$ ). Cada passo de tempo é representado por uma coluna de neurônios na versão expandida da mesma rede neural. A rede à direita é a versão futura da rede à esquerda. A saída de uma camada oculta em uma etapa de tempo ( $t$ ) é realimentada na camada oculta junto com os dados de entrada para a próxima etapa de tempo ( $t+1$ ) à direita. Este diagrama mostra duas iterações deste desdobramento, isto é, três colunas de neurônios para ( $t=0$ ), ( $t=1$ ) e ( $t=2$ ). Todos os caminhos verticais nesta visualização são clones ou visualizações dos mesmos neurônios. Eles são a única rede representada em uma linha do tempo.

Figura 4 – Rede neural recorrente expandido.



Fonte: (LANE *et al.*, 2019)

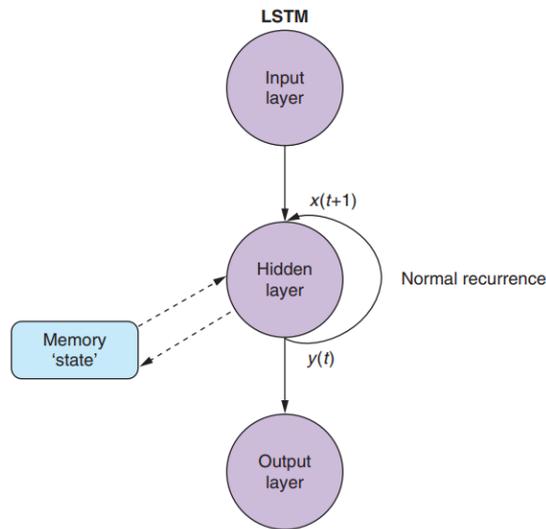
## 2.5 LSTM

Por mais que as redes neurais recorrentes convencionais forneçam vários benefícios para relacionamentos de modelagem em dados de sequência, eles sofrem de uma principal deficiência: o efeito de um *token*, que representa uma unidade de informação a ser analisada pelo modelo, é quase completamente perdido quando dois *tokens* foram analisados. Qualquer efeito que o primeiro nó tem sobre o terceiro nó (dois intervalos de tempo após o primeiro intervalo de tempo) será completamente sobreposto por novos dados introduzidos no passo de tempo interveniente. Isso é importante para a estrutura básica da rede, mas evita o caso comum na linguagem humana de que os *tokens* podem estar profundamente inter-relacionados mesmo quando estão distantes em uma frase.

Os LSTMs (*Long short-term memory*) introduzem o conceito de um estado para cada camada na rede recorrente, como pode ser observado na figura 5. Este estado atua como sua memória. Pode-se pensar nisso como adicionar atributos a uma classe na programação orientada a objetos. Os atributos do estado da memória são atualizados a cada treinamento exemplo.

Nos LSTMs, as regras que regem as informações armazenadas no estado (memória) são redes neurais treinadas. Eles podem ser treinados para aprender o que lembrar, enquanto ao mesmo tempo o resto da rede recorrente aprende a prever o rótulo de destino. Com a introdução de uma memória e estado, pode-se começar a aprender dependências que se estendem não apenas a um ou dois *tokens* de distância, mas ao longo da totalidade de cada amostra de dados. Com essas dependências de longo prazo em mãos, pode-se começar a visualizar além das próprias palavras e até algo mais profundo sobre a linguagem. Com os LSTMs, padrões que os humanos aceitam e processam em um nível subconsciente começam a estar disponíveis para o seu modelo. Com esses padrões, não se pode apenas prever classificações de amostra com mais precisão, mas também começar a gerar novos textos utilizando esses padrões. O estado da arte neste campo ainda está longe de ser perfeito, mas os resultados futuros, mesmo em seus exemplos de demonstração, devem ser impressionantes (LANE *et al.*, 2019).

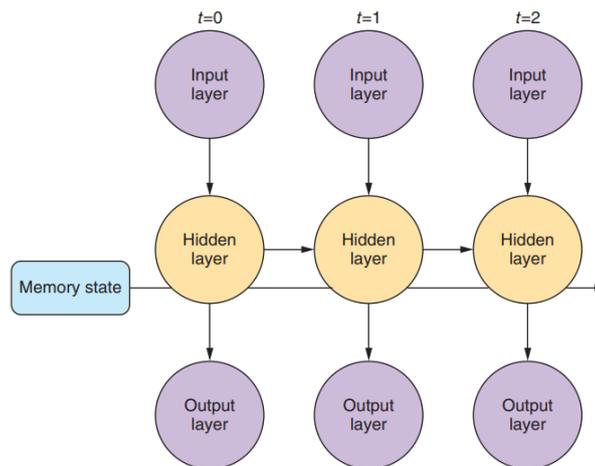
Figura 5 – Arquitetura original de um LSTM.



Fonte: (LANE *et al.*, 2019)

O estado da memória é afetado pela entrada e afeta a saída da camada, assim como em uma rede recorrente normal. Mas esse estado de memória persiste em todas as etapas de tempo da série temporal (sua sentença ou documento). Assim, cada entrada pode ter um efeito sobre o estado da memória, bem como um efeito na saída da camada oculta. O interessante do estado de memória é que ele aprende o que lembrar ao mesmo tempo que aprende a reproduzir a saída, usando retro propagação padrão.

Figura 6 – Arquitetura original de um LSTM.



Fonte: (LANE *et al.*, 2019)

Observa-se na Figura 6 a semelhança com uma rede neural recorrente normal. No entanto, além da saída de ativação que alimenta a próxima etapa de tempo da camada, adiciona-se um estado de memória que também passa por etapas de tempo da rede. A cada iteração de passo

de tempo, a unidade recorrente oculta tem acesso à unidade de memória. A adição dessa unidade de memória e os mecanismos que interagem com ela tornam isso um pouco diferente de uma camada de rede neural tradicional. No entanto, é interessante perceber que é possível projetar um conjunto de camadas de rede neural recorrentes tradicionais (um gráfico computacional) que realiza todos os cálculos existentes em uma camada LSTM. De forma simples, uma camada LSTM é apenas uma rede neural recorrente altamente especializada.

## 2.6 Função de ativação

Funções de ativação são muito importantes no contexto de redes neurais, pois ajudam no aprendizado e na compreensão de mapeamentos não lineares e complicados entre as entradas e as saídas correspondentes. A função de ativação decide se um neurônio deve ser ativado ou não, calculando a soma ponderada e adicionando viés a ela. O objetivo da função de ativação é introduzir não linearidade na saída de um neurônio.

As funções de ativação são especialmente usadas em redes neurais artificiais para transformar um sinal de entrada em um sinal de saída que, por sua vez, é alimentado como entrada para a próxima camada na pilha. Em uma rede neural artificial, calculamos a soma dos produtos das entradas e seus pesos correspondentes e, finalmente, aplicamos uma função de ativação a ela para obter a saída dessa camada específica e fornecê-la como entrada para a próxima camada (SHARMA *et al.*, 2017).

## 2.7 Loss function

Uma *Loss function* quantifica o descompasso entre as respostas corretas que a rede deve produzir e os valores das saídas reais ( $y$ ) para a entrada ( $x$ ) correspondente na rede. Ela também é conhecida como função de custo ou função de perda. A *Loss function* nos diz com que frequência nossa rede responde o valor errado e o quão erradas essas respostas estavam. A Equação descrita a seguir é um exemplo de função de custo, calculando o erro entre o valor real e a previsão do modelo.

$$err(x) = |y - f(x)|$$

## 2.8 Otimizadores

Otimizadores são algoritmos utilizados para alterar diversos atributos da rede neural, como pesos e taxa de aprendizado. Essas alterações são realizadas em todas as épocas durante o processo de treinamento do modelo, com o intuito de reduzir o valor retornado pela *Loss Function*, de forma a melhorar os resultados obtidos pelo modelo (TATO; NKAMBOU, 2018).

## 2.9 Métricas de avaliação

Em aprendizado de máquina, uma matriz de confusão é um layout de tabela específico que permite visualizar o desempenho de um modelo, tipicamente de um aprendizado supervisionado. Cada linha da matriz representa as instâncias de uma classe real enquanto cada coluna representa as instâncias de uma classe predita, ou vice-versa. Isto é, uma matriz de confusão é uma tabela comparativa dos valores que um algoritmo trouxe como predição em relação aos valores reais. A Figura 7 ilustra uma matriz de confusão para duas classes.

Figura 7 – Exemplo de uma matriz de confusão de duas classes.

		Valor Predito	
		Negativo (0)	Positivo (1)
Valor Real	Negativo (0)	VN	FP
	Positivo (1)	FN	VP

Fonte: Elaborada pelo autor.

Os quadrantes dessa matriz são descritos a seguir:

- Verdadeiro positivo (VP): ocorre quando, no conjunto real, a classe que estamos buscando foi prevista corretamente. Por exemplo, quando o sentimento é positivo e o modelo previu corretamente que o sentimento do texto analisado é positivo.
- Verdadeiro Negativo (VN): ocorre quando, no conjunto real, a classe que não estamos buscando foi prevista corretamente. Por exemplo: o sentimento não é positivo, e o modelo previu corretamente que o sentimento do texto analisado não é positivo.

- Falso Positivo (FP): ocorre quando, no conjunto real, a classe que estamos buscando foi prevista incorretamente. Por exemplo: o sentimento é positivo, mas o modelo previu que o texto analisado contém sentimento negativo.
- Falso Negativo (FN): ocorre quando, no conjunto real, a classe que não estamos buscando prever foi prevista incorretamente. Por exemplo, o sentimento é negativo, mas o modelo previu que o texto analisado contém sentimento positivo.

No contexto de aprendizado de máquina, pode-se utilizar algumas métricas de desempenho com o intuito de avaliar a qualidade de um determinado modelo. As métricas mais utilizados para isso são: acurácia, precisão, revocação e F1.

Acurácia é uma métrica de avaliação comum para problemas de classificação. É o número de previsões corretas feitas como uma proporção de todas as previsões feitas. A precisão é comumente utilizada para avaliar o desempenho do modelo. No entanto, existem algumas desvantagens que devem ser consideradas antes de usar a precisão de forma liberal. Uma dessas desvantagens lida com conjuntos de dados desequilibrados, onde uma classe, verdadeira ou falsa, é mais comum que a outra, fazendo com que o modelo classifique as observações com base nesse desequilíbrio. A fórmula da acurácia, com base na matriz de confusão, é descrita a seguir:

$$acuracia = \frac{VP + VN}{VP + FN + VN + FP}$$

Precisão é a medida de verdadeiros positivos sobre o número total de positivos previstos pelo seu modelo. O que essa métrica permite calcular é a taxa em que suas previsões positivas são realmente positivas. A precisão é definida como:

$$precisao = \frac{VP}{VP + FP}$$

Revocação é a medida do seu verdadeiro positivo sobre a contagem de resultados positivos reais. Usando a fórmula descrita abaixo, pode-se avaliar quão bem nosso modelo é capaz de identificar o resultado real verdadeiro.

$$revocacao = \frac{VP}{VP + FN}$$

O F1 é a média harmônica entre precisão e a revocação. Essa pontuação pode ser usada como uma métrica geral que incorpora precisão e revocação. A razão pela qual usamos a

média harmônica em oposição à média regular é que a média harmônica pune valores que estão mais distantes.

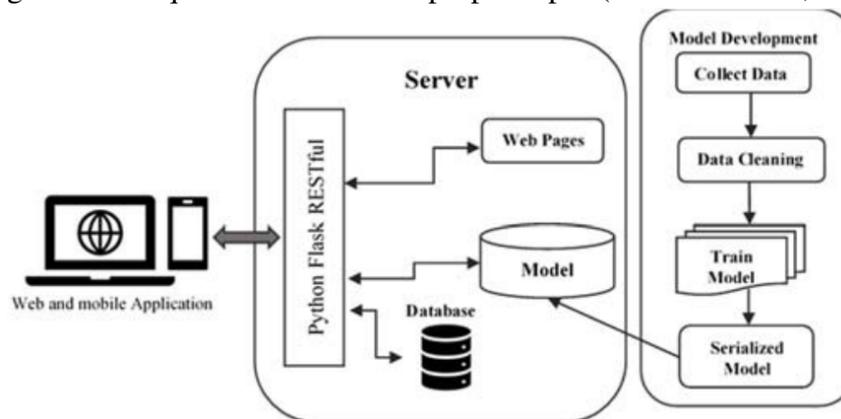
$$f1 = \frac{\textit{precisao} * \textit{revocacao}}{\textit{precisao} + \textit{revocacao}}$$

### 3 TRABALHOS CORRELATOS

Existem vários trabalhos na área de NLP, que se propõem a utilizar aprendizado de máquina para análise de linguagem natural. Os trabalhos que têm foco em análise de sentimentos são citados a seguir.

O estudo realizado por (JABBAR *et al.*, 2019) utiliza uma técnica de *machine learning* denominada *Support Vector Machine (SVM)* que foi utilizada para desenvolver um modelo de NLP (*Natural language processing*) em uma aplicação de *e-commerce*. Este trabalho está ilustrado na Figura 8. O objetivo principal deste trabalho é apresentar um processo de análise de sentimentos aplicado em tempo real em uma aplicação de *e-commerce* com o intuito de melhorar a experiência do usuário.

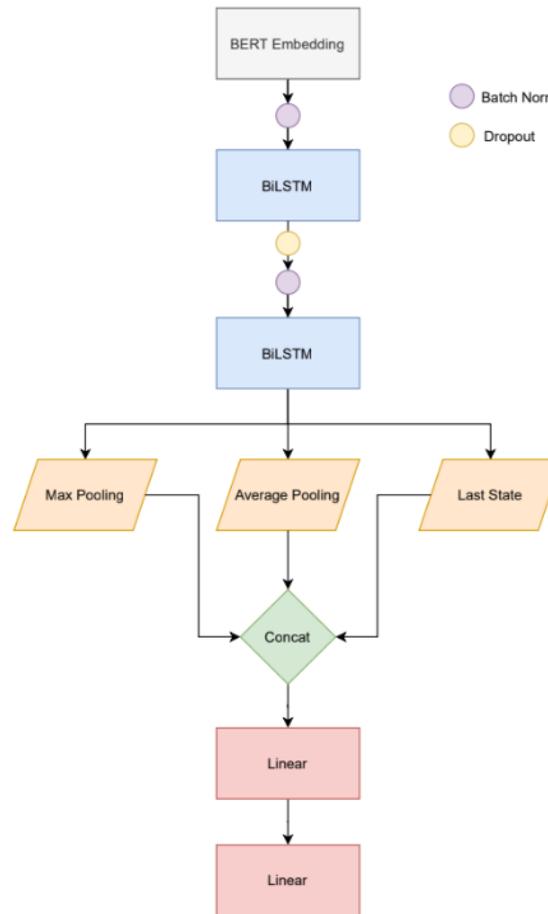
Figura 8 – Arquitetura do sistema proposto por (JABBAR *et al.*, 2019).



Fonte: (JABBAR *et al.*, 2019)

Já o trabalho em (CHAVES *et al.*, 2022) propõe uma nova abordagem denominada BPA. Essa nova abordagem foi desenvolvida baseada em redes neurais BiLSTM, operações de agrupamento e mecanismo de atenção, que é capaz de classificar automaticamente os níveis de polaridade de um texto. A arquitetura do sistema proposto pode ser visualizada na Figura 9. Esta nova abordagem foi avaliada utilizando cinco conjuntos de dados distintos em três diferentes idiomas: inglês, espanhol e português. Os melhores resultados alcançados pela abordagem BPA atingiram uma precisão de 0,901, 0,865 e 0,923 para inglês, espanhol e português, respectivamente.

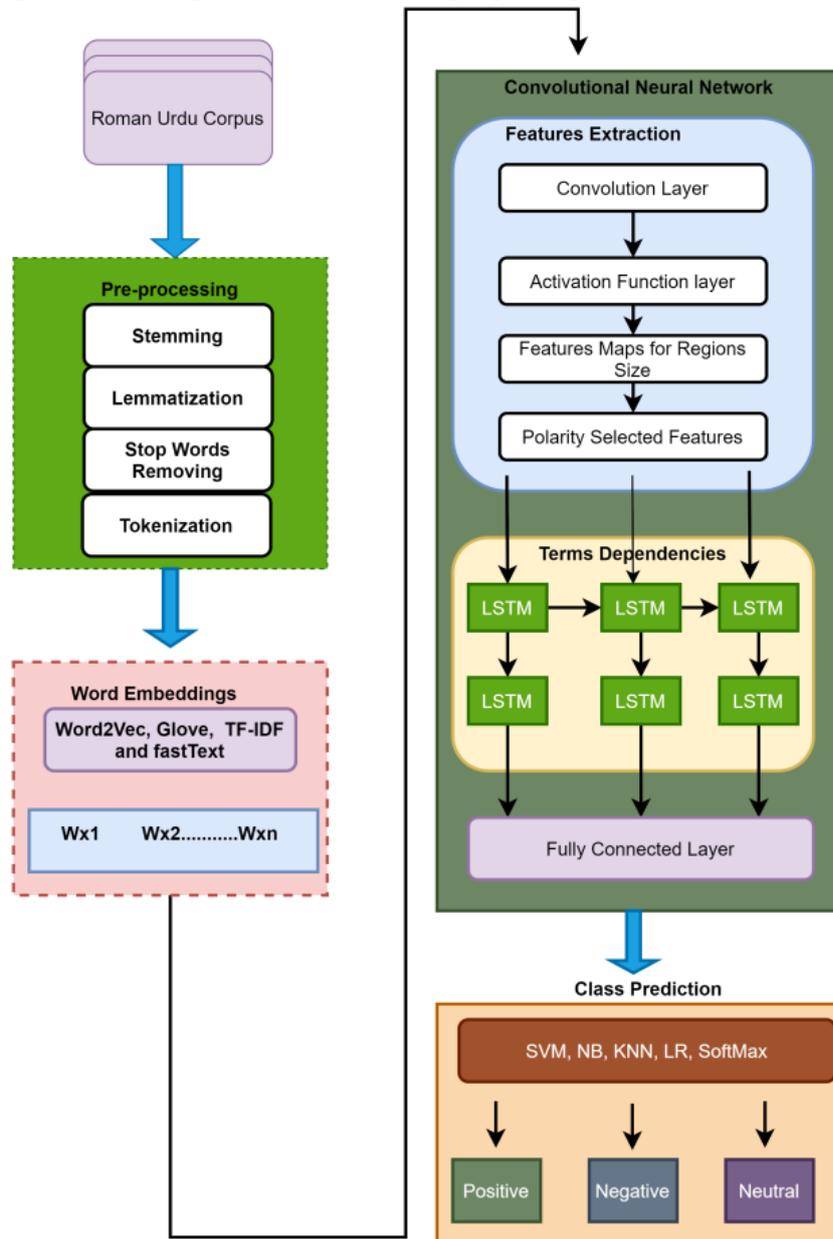
Figura 9 – Arquitetura do sistema proposto por (CHAVES *et al.*, 2022).



Fonte: (CHAVES *et al.*, 2022)

Por fim, no estudo em (KHAN *et al.*, 2022) é proposta uma nova arquitetura de *deep learning* para análise dos dialetos urdu romano e inglês SA com base em duas camadas: LSTM e um modelo CNN. Para obter a classificação final, os mapas de *features* aprendidas pela CNN e LSTM são inseridos em vários classificadores de aprendizado de máquina, como, SVM (*Support Vector Machine*), NB (*Naive Bayes*), LR (*Logistic Regression*), RF (*Random forest*), KNN (*K-nearest neighbors*), e softmax, conforme apresentado na Figura 10. Nesse estudo, vários testes extensivos foram aplicados em quatro bases de dados distintas e apresentaram resultados bastante satisfatórios em relação a classificação de sentimentos em textos escritos em urdu romano e inglês. Os resultados mostram uma precisão de 0,904, 0,841, 0,740 e 0,748 em relação aos conjuntos de dados MDPI, RUSA, RUSA-19, e UCL, respectivamente.

Figura 10 – Arquitetura do sistema proposto por (KHAN *et al.*, 2022).



Fonte: (KHAN *et al.*, 2022)

## 4 METODOLOGIA

Este trabalho é de caráter experimental, pois atua diretamente sobre as variáveis relacionadas ao objeto de estudo. Essas variáveis são os dados e a forma como esses dados são tratados, afetando diretamente os resultados obtidos. Sua atividade principal consiste em identificar sentimentos positivos e negativos a partir da análise de comentários e *reviews* em plataformas de *e-commerce*.

### 4.1 Dados utilizados

Para esse trabalho foram selecionadas cinco bases de dados reais, sendo três delas utilizadas conforme estavam disponíveis originalmente, sendo elas (B2W, Buscape e *Olist*), e uma construída a partir da integração de todas as bases. Esses dados são compostos por *reviews* e comentários coletados de plataformas de vendas *online*, sendo estes: B2W Digital, Buscapé, *Olist* e duas bases de dados diferentes da UTLCorpus. A rotulação utilizada foi a mesma para todas as bases de dados. Para isso, foi utilizado o valor do *rating*, que pode variar de uma a cinco estrelas. Foi considerado *reviews* que continham quatro e cinco estrelas como *reviews* positivos (1), *reviews* que tinham uma avaliação de uma e duas estrelas foram consideradas como negativo (0), *reviews* com três estrelas foram desconsiderados (DIAS, 2022). Para as bases de dados utilizadas, os dados foram divididos da seguinte forma: 70% dos dados foram utilizados para o treinamento do modelo, os 30% restantes foram divididos em duas partes, onde 15% foram destinados para testes e os últimos 15% restantes foram utilizados para a avaliação do modelo treinado. Também é importante citar que as três bases de dados (B2W, Buscape e *Olist*) possuem um problema de balanceamento, onde a quantidade de classes positivas (1) excede em muito a quantidade de classes negativas (0). Tendo isso em vista, foi criada uma quarta base de dados chamada de "Concat". Esta base de dados foi montada realizando uma junção das cinco bases de dados previamente citadas. Depois foi feito um *under sample* (técnica que consiste em manter todos os dados da classe com menor frequência e diminuir a quantidade dos que estão na classe de maior frequência) com o intuito de igualar a quantidade das classes. Este pré-processamento foi considerado para analisar o modelo em uma base de dados balanceada.

A Tabela 1 a seguir descreve de forma detalhada as quantidades e as distribuições dos dados utilizados.

Tabela 1 – Distribuição dos dados por base de dados.

	<b>Total</b>	<b>Classe 0</b>	<b>Classe 1</b>	<b>Treino</b>	<b>Teste</b>	<b>Validação</b>
<b>B2W</b>	116.058	35.757	80.300	81.240	17.409	17.409
<b>Buscape</b>	73.627	6.810	66.817	51.538	11.044	11.044
<b>Olist</b>	38.079	11.408	26.671	26.655	5.712	5.712
<b>Utlc_apps</b>	1.039.535	218.114	750.744	678.200	145.329	145.329
<b>Utlc_movies</b>	1.487.449	137.539	1.052.003	832.679	178.432	178.431
<b>Concat</b>	819.258	409.629	409.629	573.480	122.889	122.889

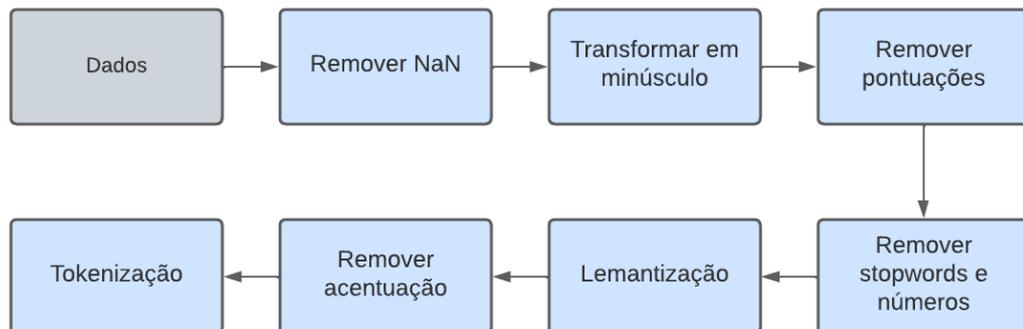
Fonte: Elaborada pelo autor.

## 4.2 Pré-processamento

O pré-processamento consiste em uma coleção de técnicas nas quais os textos analisados são pré-processados com o intuito de eliminar formatos especiais comumente encontrado em textos, como formatos de números, preposições, artigos, pronomes, entre outros (KAN-NAN *et al.*, 2014). Essa etapa é uma das primeiras e mais importantes etapas em um processo de análise de sentimentos, podendo afetar diretamente o resultado e a qualidade das análises (CAMACHO-COLLADOS; PILEHVAR, 2017).

Para o pré-processamento foi utilizada a biblioteca *pandas*, uma biblioteca de *software* criada para a linguagem *Python*, utilizada para manipulação e análise de dados. Em particular, essa biblioteca oferece estruturas e operações para manipular tabelas numéricas e séries temporais. A seguir são listadas as técnicas de pré-processamento utilizadas neste trabalho.

Figura 11 – Fluxo de pré-processamento.



Fonte: Elaborada pelo autor.

### 4.2.1 Remoção de elementos NaN

Durante a análise dos conjuntos de dados escolhidos, foi notado que alguns elementos continham o elemento NaN (*not a number*), uma representação para informar a ausência de

valor. Esse tipo de dado não contém nenhum valor significativo para nossa análise, por isso, é necessário removê-los da nossa base de dados. A biblioteca *pandas* contém um método chamado *dropna* que foi utilizado para remover as linhas do conjunto de dados que continham elementos NaN.

#### 4.2.2 *Transformação do texto em minúsculo*

Esta etapa consiste em transformar todos os caracteres de um texto para o seu formato minúsculo (CAMACHO-COLLADOS; PILEHVAR, 2017). Devido a sua simplicidade, esse processo é uma prática bastante popular em módulos de bibliotecas de *deep learning* e em *word embedding packages* (PENNINGTON *et al.*, 2014; FARUQUI *et al.*, 2014). O Exemplo 1 mostra o funcionamento de um processo de transformação de texto em minúsculo.

**Exemplo 1** *Antes: ÓTIMO PRODUTO, atendeu as expectativas! Comprei 3 relógios!*

*Depois: ótimo produto, atendeu as expectativas! comprei 3 relógios!*

#### 4.2.3 *Remoção de pontuação*

Esta etapa consiste em remover toda pontuação contida nos textos. Esse processo é importante, pois a mesma palavra pode aparecer em contextos diferentes. Dependendo da pontuação utilizada, a mesma palavra pode ser considerada uma palavra diferente. Outro motivo para a remoção de pontuações é o fato de que pontuações não possuem um significado no processo de análise de sentimentos. O Exemplo 2 mostra o processo de remoção de pontuações do texto.

**Exemplo 2** *Antes: ótimo produto, atendeu as expectativas! comprei 3 relógios!*

*Depois: ótimo produto atendeu as expectativas comprei 3 relógios*

#### 4.2.4 *Remoção de stopwords e números*

Muitas palavras contidas em textos se repetem com muita frequência, mas são essencialmente sem sentido, pois são usadas para unir palavras em uma frase. Essas palavras são denominadas *stopwords*. É comumente entendido que *stopwords* não contribuem para o contexto ou conteúdo de textos. Devido à sua alta frequência de ocorrência, sua presença na mineração de texto apresenta um obstáculo na compreensão do conteúdo dos textos analisados (KANNAN *et*

al., 2014). Para a remoção de *stopwords* foi utilizado o NLTK (*Natural Language Toolkit*), que consiste em um conjunto de bibliotecas e programas em *Python* para processamento simbólico e estatístico de linguagem natural. NLTK foi escolhido pois fornece a possibilidade de filtrar *stopwords* de um texto em português. No Exemplo 3 é possível observar a remoção de números e de *stopwords*, nesse caso a palavra "as" e o número "3".

**Exemplo 3** *Antes: ótimo produto atendeu as expectativas comprei 3 relógios*

*Depois: ótimo produto atendeu expectativas comprei relógios*

#### 4.2.5 Lematização

O processo de lematização consiste em substituir uma palavra específica pela sua palavra de origem correspondente. A ideia principal por trás da lematização é reduzir a diferença entre palavras com a mesma origem, pois diferentes formas flexionadas da mesma palavra podem ocorrer com alguma frequência durante o treinamento (CAMACHO-COLLADOS; PILEHVAR, 2017). A lematização tem sido tradicionalmente uma técnica de pré-processamento padrão para sistemas de classificação de texto (MULLEN; COLLIER, 2004; TOMAN *et al.*, 2006; HASSAN *et al.*, 2007). Para a aplicação da lematização, foi utilizada a biblioteca *spaCy*, uma biblioteca de *software* de código aberto para processamento avançado de linguagem natural em *Python*. A biblioteca *spaCy* foi escolhida, pois fornece uma maneira fácil de aplicar lematização em textos escritos no idioma português brasileiro. O Exemplo 4 mostra o funcionamento de um processo de lematização.

**Exemplo 4** *Antes: ótimo produto atendeu expectativas comprei relógios*

*Depois: bom produto atender expectativa comprar relógio*

#### 4.2.6 Remoção de acentuação

Esta etapa é bastante importante considerando que os textos analisados são *reviews* coletados de sites de vendas online. Nesse cenário, os usuários tendem a não prestar muita atenção a acentuação, ou seja, a mesma palavra pode ser considerada uma completamente diferente caso a acentuação esteja errada, dificultando dessa forma o processo de análise. O Exemplo 5 mostra o funcionamento de um processo de remoção de acentuação do texto.

**Exemplo 5** *Antes: bom produto atender expectativa comprar relógio*

*Depois: bom produto atender expectativas comprar relógio*

#### 4.2.7 Tokenização

*Tokenização* é o processo de quebrar um fluxo de texto em palavras, frases, símbolos ou outros elementos significativos, denominados *tokens*. Para esse trabalho foi utilizado o tipo de *tokenização* por palavra, que consiste em dividir um determinado pedaço de texto em palavras individuais de acordo com um delimitador específico. Em outras palavras, o principal uso da *tokenização* é identificar as palavras-chave significativas em um determinado texto (KANNAN *et al.*, 2014). Para isso, foi utilizada a biblioteca *Keras*, uma biblioteca de rede neural de alto nível que é executada no *TensorFlow*. *Keras* foi escolhida pela facilidade que a biblioteca permite ao realizar o processo de *tokenização*. O Exemplo 6 mostra o funcionamento de um processo de *tokenização* do texto.

**Exemplo 6** *Antes: bom produto atender expectativas comprar relógio*

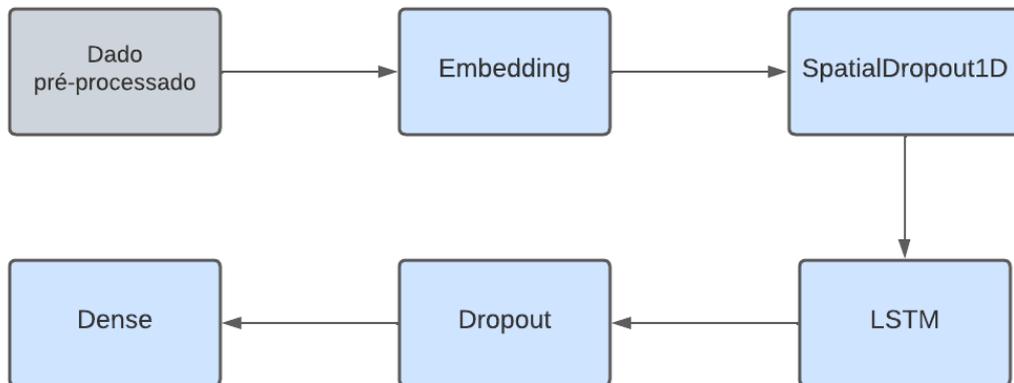
*Depois: ['bom','produto','atender','expectativas','comprar','relógio']*

### 4.3 Construção do modelo

Para este trabalho foi desenvolvido um modelo de aprendizado de máquina utilizando o *TensorFlow* para facilitar a criação do modelo de NLP de análise de sentimentos. O modelo foi criado a partir da classe *Sequential* do *Keras* que é responsável por ordenar a sequência de camadas presentes no modelo. Nesse trabalho as seguintes camadas foram utilizadas para construir o modelo utilizando a biblioteca *Keras*: *Embedding*, *SpatialDropout1D*, *LSTM*, *Dropout* e *Dense* utilizando função de ativação *Sigmoid*. O modelo foi compilado utilizando o *binary\_crossentropy* como *loss function* e otimizador Adam (*Adaptive Moment estimation*). Também é importante ressaltar que para esse trabalho o modelo foi treinado rodando um total de 100 épocas para todas as base de dados analisadas, valor esse onde o modelo começa a convergir para as três primeiras bases de dados analisadas (B2W, Buscape e Olist).

### 4.3.1 Camadas do modelo

Figura 12 – Fluxo do pré-processamento.



Fonte: Elaborada pelo autor.

- *Embedding*: do tipo *word embedding*, é a camada responsável por transformar inteiros positivos (índices) em vetores densos de tamanho fixo;
- *SpatialDropout1D*: executa a mesma função que *Dropout*, no entanto, ele descarta mapas de recursos 1D inteiros em vez de elementos individuais;
- *LSTM*: essa camada aprende dependências de longo prazo entre etapas de tempo, foram utilizados 50 neurônios nessa camada do modelo.
- *Dropout*: define aleatoriamente as unidades de entrada para 0 com uma frequência de taxa em cada etapa durante o tempo de treinamento, o que ajuda a evitar o *overfitting*.
- *Dense*: implementa a operação:  $output = activation(dot(input, kernel) + bias)$  onde ativação é a função de ativação por elemento passada como argumento de ativação (no modelo proposto a função utilizada foi *Sigmoid*).

É importante mencionar a função de ativação, *Loss function* e o otimizador utilizados na construção deste modelo. Para a função de ativação, *Sigmoid* foi a opção escolhida, por ser muito performática além de ser a função de ativação mais utilizada e presente na literatura (SHARMA *et al.*, 2017). A *Loss function* escolhida para este trabalho foi a *binary\_crossentropy*, pelo fato de ser a *Loss function* mais utilizada para problemas de classificação. A *Binary\_CrossEntropy* é uma função que mede o quão longe do valor verdadeiro da previsão está para cada uma das classes e então calcula a média dos erros de classe para obter a perda final (BRAHMBHATT; RAJAN, 2019). Por fim, o otimizador escolhido foi o Adam (*Adaptive Moment Estimation*), pelo fato de ser um dos algoritmos de otimização mais presentes na literatura.

tura (GOODFELLOW *et al.*, 2016; ISOLA *et al.*, 2017; XU *et al.*, 2015). O otimizador Adam, é um método para otimização estocástica eficiente que requer apenas gradientes de primeira ordem com pouca necessidade de memória. O método calcula as taxas individuais de aprendizado adaptativo para parâmetros diferentes das estimativas de primeiro e segundo momentos dos gradientes (KINGMA; BA, 2014).

## 5 RESULTADOS

Este capítulo detalha as especificações do ambiente de desenvolvimento e execução do modelo, além de apresentar e analisar os resultados obtidos a partir das quatro bases de dados nas quais o modelo foi treinado: B2W Digital, Buscapé, Olist e Concat.

### 5.1 Especificações do ambiente

Para este trabalho o ambiente de desenvolvimento e execução consistiu em uma máquina com as seguintes especificações: Windows 10 Home 64 bits, processador AMD Ryzen 5 3600 6-Core Processor 3.60 GHz, memória RAM de 16GB e armazenamento de dados em SSD. Para a implementação do modelo foi utilizada a linguagem de programação *Python* em conjunto com o *JupyterLab*, que consiste em um ambiente de desenvolvimento interativo baseado na web para *notebooks*. Para este projeto foram utilizadas as bibliotecas listadas a seguir:

- *TensorFlow*: utilizada para facilitar a criação e a implantação de modelos de aprendizado de máquina;
- *Matplotlib*: utilizada para visualização de dados em gráficos.
- *Pandas*: utilizada para manipulação e análise de dados.
- *Seaborn*: utilizada juntamente com a *Matplotlib* para exibição de figuras.
- *Nltk*: utilizada para filtrar *stopwords* de um texto em português.
- *Pickle*: utilizado para salvar os dados de treino, teste e validação.
- *Unidecode*: utilizado para remover acentos na etapa de pré-processamento;
- *Spacy*: biblioteca que fornece uma forma fácil de aplicar lematização em textos escritos no idioma português brasileiro.

### 5.2 Apresentação de métricas e análise de resultados

Para a apresentação dos resultados deste trabalho as seguintes métricas foram utilizadas para avaliar a qualidade do modelo proposto: acurácia, precisão, sensibilidade e F1. Também foram utilizadas matrizes de confusão para ajudar a medir o desempenho do modelo.

### 5.2.1 Resultados obtidos

A seguir são apresentados os resultados obtidos referente a cada base de dados analisadas neste trabalho: B2W, Buscape e Olist. É possível visualizar na Tabela abaixo contendo as acurácias obtidas após a análise de cada uma das bases de dados mencionadas acima.

Tabela 2 – Acurácia dos três bases de dados analisadas

	<b>b2w</b>	<b>buscape</b>	<b>olist</b>	<b>concat</b>
<b>acurácia</b>	93,29%	93,63%	90,70%	86,77%

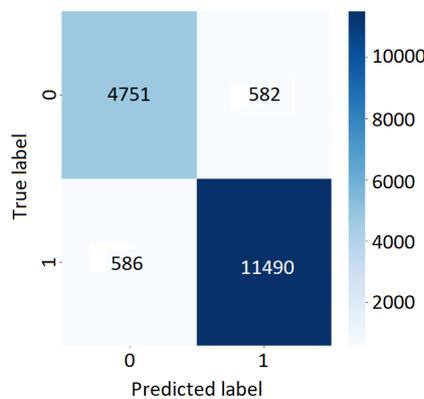
Fonte: Elaborada pelo autor.

Observa-se na tabela anterior que o modelo atingiu uma acurácia acima de 85% para os quatro conjuntos de dados testados.

#### 5.2.1.1 B2W

A seguir está listada a matriz de confusão e as métricas (precisão, sensibilidade e f1) referente a base de dados B2W.

Figura 13 – Matriz de confusão da base de dados B2W.



Fonte: Elaborada pelo autor.

Como pode ser observado na matriz de confusão acima, referente a base de dados B2W, o número de acertos para a classe 0 é de 4751 contra 582 predições erradas pelo modelo. Já na classe 1, obtivemos 11490 predições acertadas pelo modelo e 586 predições erradas.

Tabela 3 – Métricas B2W

	<b>precisão</b>	<b>revocação</b>	<b>f1</b>
<b>classe 1</b>	95,17%	95,14%	95,16%
<b>classe 0</b>	89,02%	89,08%	89,05%

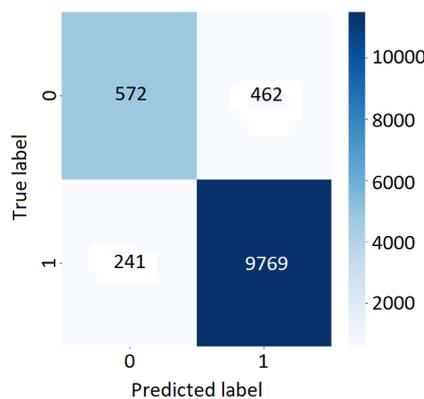
Fonte: Elaborada pelo autor.

Após a análise da base de dados B2W, obtivemos as métricas apresentadas na tabela acima. A partir dessa tabela é possível observar que a precisão do modelo referente a classe 1 apresenta um resultado maior que 95%, o que segundo a literatura pode ser considerado um ótimo resultado. Já a classe 0, apresentou uma precisão abaixo da classe 1, atingindo cerca de 89% de precisão. Isso pode ser atribuído ao desbalanceamento da base de dados que por sua vez possui menor frequência de ocorrências de classe 0 em comparação a classe 1.

### 5.2.1.2 Buscape

A seguir está listada a matriz de confusão e as métricas (precisão, sensibilidade e f1) referente a base de dados Buscape.

Figura 14 – Matriz de confusão da base de dados Buscape.



Fonte: Elaborada pelo autor.

Com base na matriz de confusão obtido após a análise da base de dados Buscape, o número de acertos para a classe 0 é de 572 contra 462 predições erradas pelo modelo. Já na classe 1, obtivemos 9769 predições acertadas pelo modelo e 241 predições erradas.

Tabela 4 – Métricas Buscape

	<b>precisão</b>	<b>revocação</b>	<b>f1</b>
<b>classe 1</b>	95,48%	97,59%	96,52%
<b>classe 0</b>	70,35%	55,31%	61,93%

Fonte: Elaborada pelo autor.

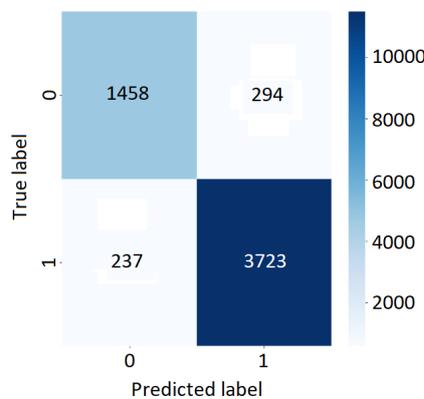
Na tabela acima estão apresentadas as métricas obtidas a partir da análise da base de dados Buscape. A partir dessa tabela é possível observar que a precisão do modelo referente a classe 1 apresenta um resultado maior que 95%, o que segundo a literatura pode ser considerado um ótimo resultado. Já a classe 0, apresentou uma precisão bem abaixo da classe 1, atingindo

uma precisão de pouco mais de 70%. Essa diferença pode ser atribuída ao desbalanceamento da base de dados que por sua vez possui menor frequência de ocorrências de classe 0 em comparação a classe 1.

### 5.2.1.3 Olist

A seguir está listado a matriz de confusão e as métricas (precisão, sensibilidade e f1) referente a base de dados Olist.

Figura 15 – Matriz de confusão da base de dados Olist.



Fonte: Elaborada pelo autor.

Como pode ser observado na matriz de confusão acima, referente a base de dados Olist, o número de acertos para a classe 0 é de 1458 contra 294 predições erradas pelo modelo. Já na classe 1, obtivemos 3723 predições acertadas pelo modelo e 237 predições erradas.

Tabela 5 – Métricas Olist

	<b>precisão</b>	<b>revocação</b>	<b>f1</b>
<b>classe 1</b>	92,68%	94,01%	93,34%
<b>classe 0</b>	86,01%	83,21%	84,59%

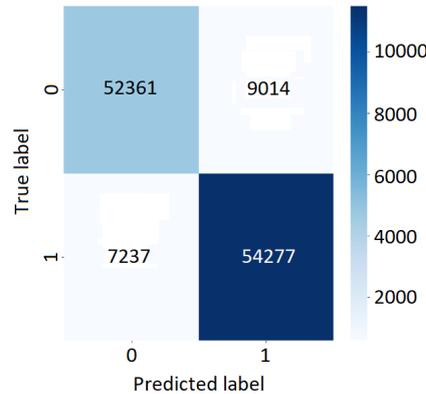
Fonte: Elaborada pelo autor.

As métricas obtidas após a análise da base de dados Olist pode ser observado logo acima. A partir dessa tabela é possível observar que a precisão do modelo referente a classe 1 apresenta um resultado maior que 92%, o que segundo a literatura pode ser considerado um bom resultado. Já a classe 0, apresentou uma precisão bem abaixo da classe 1, atingindo uma precisão de pouco mais de 86%. Essa diferença pode ser atribuída ao desbalanceamento da base de dados que por sua vez possui menor frequência de ocorrências de classe 0 em comparação a classe 1.

### 5.2.1.4 Concat

A seguir está listado a matriz de confusão e as métricas (precisão, sensibilidade e f1) referente a base de dados Concat.

Figura 16 – Matriz de confusão da base de dados Concat.



Fonte: Elaborada pelo autor.

Com base na matriz de confusão obtido após a análise da base de dados *Concat*, o número de acertos para a classe 0 é de 572 contra 462 predições erradas pelo modelo. Já na classe 1, obtivemos 9769 predições acertadas pelo modelo e 241 predições erradas.

Tabela 6 – Métricas *Concat*

	precisão	revocação	f1
<b>classe 1</b>	85,75%	88,23%	86,97%
<b>classe 0</b>	87,85%	85,31%	86,56%

Fonte: Elaborada pelo autor.

Após a análise da base de dados *Concat*, obtivemos as métricas apresentadas na tabela acima. A partir dessa tabela é possível observar que a precisão do modelo referente a classe 1 apresenta um resultado maior que 85%. Já a classe 0, apresentou uma precisão um pouco maior, chegando a mais de 87%. É possível observar que para essa base de dados os valores das precisões estão bem mais próximos quando comparados as bases de dados previamente analisadas. Isso pode ser atribuído ao fato desses dados estarem bem mais balanceada do que as anteriores.

## 6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho teve como motivação a popularização do *e-commerce* na última década, bem como o desejo dessas empresas em compreender melhor seus usuários, e por último a necessidade desses usuários de expressarem suas opiniões e sentimentos a respeito de experiências, produtos e serviços. Para isso, desenvolvemos um modelo para análise de sentimentos em comentários e *reviews* coletados de diferentes plataformas de *e-commerce* como B2W, Buscape, Olist e UTLCorpus. Devido ao fato das bases de dados escolhidas sofrerem de um problema de desbalanceamento, criamos um conjunto de dados com a combinação das bases de dados previamente citadas. Após a concatenação das bases de dados, utilizamos uma técnica de *undersample* para igualar a quantidade da classe de maior frequência com a de menor frequência. Após a coleta desses dados aplicamos várias técnicas de pré-processamento com o intuito de tratar os dados para a análise. Após o pré-processamento foi criado um modelo de rede neural recorrente utilizando o LSTM para analisar os comentários. Nossos resultados mostraram que o modelo proposto apresenta ótimos resultados para análise de sentimento positivo e negativo sobre dados de comentários de plataformas de *e-commerce*.

Como trabalhos futuros, pretende-se testar a utilização de outras redes neurais, por exemplo BiLSTM, com o intuito de fazer um comparativo entre os modelos propostos e identificar qual seria mais performático para a tarefa de análise de sentimentos. De forma geral, espera-se testar diferentes abordagens na construção do modelo, como a quantidade de células utilizadas na camada LSTM. Adicionalmente, pretende-se adicionar o sentimento neutro ao modelo. Por fim, planeja-se utilizar conjuntos de dados maiores e mais balanceados para melhorar o treinamento do modelo para todas as classes às quais se deseja prever.

## REFERÊNCIAS

- BRAHMBHATT, P.; RAJAN, S. N. Skin lesion segmentation using segnet with binary crossentropy. In: **Proceedings of the International Conference on Artificial Intelligence and Speech Technology (AIST2019), Delhi, India**. [S.l.: s.n.], 2019. p. 14–15.
- CAMACHO-COLLADOS, J.; PILEHVAR, M. T. On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. **arXiv preprint arXiv:1707.01780**, 2017.
- CHAVES, I. C.; MARTINS, A. D. F.; PRACIANO, F. D.; BRITO, F. T.; MONTEIRO, J. M.; MACHADO, J. C. Bpa: A multilingual sentiment analysis approach based on bilstm. In: **ICEIS (1)**. [S.l.: s.n.], 2022. p. 553–560.
- COSTA, E. D. **Global e-commerce strategies for small businesses**. [S.l.]: Mit Press, 2003.
- DIAS, F. **Brazilian Portuguese Sentiment Analysis Datasets**. 2022. <[https://www.kaggle.com/datasets/fredericods/ptbr-sentiment-analysis-datasets?select=utlc\\_movies.csv](https://www.kaggle.com/datasets/fredericods/ptbr-sentiment-analysis-datasets?select=utlc_movies.csv)>. Accessed: 2022-10-22.
- FARUQUI, M.; DODGE, J.; JAUHAR, S. K.; DYER, C.; HOVY, E.; SMITH, N. A. Retrofitting word vectors to semantic lexicons. **arXiv preprint arXiv:1411.4166**, 2014.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.; BENGIO, Y. **Deep learning, volume 1**. [S.l.]: MIT press Cambridge, 2016.
- HAGENAU, M.; LIEBMANN, M.; NEUMANN, D. Automated news reading: Stock price prediction based on financial news using context-capturing features. **Decision Support Systems**, Elsevier, v. 55, n. 3, p. 685–697, 2013.
- HASSAN, S.; MIHALCEA, R.; BANEAN, C. Random walk term weighting for improved text classification. **International Journal of Semantic Computing**, World Scientific, v. 1, n. 04, p. 421–439, 2007.
- ISOLA, P.; ZHU, J.-Y.; ZHOU, T.; EFROS, A. A. Image-to-image translation with conditional adversarial networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 1125–1134.
- JABBAR, J.; UROOJ, I.; JUNSHENG, W.; AZEEM, N. Real-time sentiment analysis on e-commerce application. In: IEEE. **2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)**. [S.l.], 2019. p. 391–396.
- JOHRI, P.; KHATRI, S. K.; AL-TAANI, A. T.; SABHARWAL, M.; SUVANOV, S.; KUMAR, A. Natural language processing: History, evolution, application, and future work. In: SPRINGER. **Proceedings of 3rd International Conference on Computing Informatics and Networks**. [S.l.], 2021. p. 365–375.
- KANNAN, S.; GURUSAMY, V.; VIJAYARANI, S.; ILAMATHI, J.; NITHYA, M.; KANNAN, S.; GURUSAMY, V. Preprocessing techniques for text mining. **International Journal of Computer Science & Communication Networks**, v. 5, n. 1, p. 7–16, 2014.
- KHAN, L.; AMJAD, A.; AFAQ, K. M.; CHANG, H.-T. Deep sentiment analysis using cnn-lstm architecture of english and roman urdu text shared in social media. **Applied Sciences**, MDPI, v. 12, n. 5, p. 2694, 2022.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

LANE, H.; HAPKE, H.; HOWARD, C. **Natural Language Processing in Action: Understanding, analyzing, and generating text with Python**. Manning Publications, 2019. ISBN 9781617294631. Disponível em: <<https://books.google.com.br/books?id=UyHgswEACAAJ>>.

LIU, B. **Sentiment Analysis and Opinion Mining**. Morgan & Claypool, 2012. (Synthesis digital library of engineering and computer science). ISBN 9781608458844. Disponível em: <<https://books.google.com.br/books?id=Gt8g72e6MuEC>>.

LUKOŠEVIČIUS, M.; JAEGER, H. Reservoir computing approaches to recurrent neural network training. **Computer Science Review**, Elsevier, v. 3, n. 3, p. 127–149, 2009.

MAKS, I.; VOSSEN, P. A lexicon model for deep sentiment analysis and opinion mining applications. **Decision Support Systems**, Elsevier, v. 53, n. 4, p. 680–688, 2012.

MEDHAT, W.; HASSAN, A.; KORASHY, H. Sentiment analysis algorithms and applications: A survey. **Ain Shams engineering journal**, Elsevier, v. 5, n. 4, p. 1093–1113, 2014.

MITCHELL, T. M.; MITCHELL, T. M. **Machine learning**. [S.l.]: McGraw-hill New York, 1997. v. 1.

MULLEN, T.; COLLIER, N. Sentiment analysis using support vector machines with diverse information sources. In: **Proceedings of the 2004 conference on empirical methods in natural language processing**. [S.l.: s.n.], 2004. p. 412–418.

OSISANWO, F.; AKINSOLA, J.; AWODELE, O.; HINMIKAIYE, J.; OLAKANMI, O.; AKINJOBI, J. Supervised machine learning algorithms: classification and comparison. **International Journal of Computer Trends and Technology (IJCTT)**, v. 48, n. 3, p. 128–138, 2017.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: **Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)**. [S.l.: s.n.], 2014. p. 1532–1543.

RUSSELL, S.; NORVIG, P.; CANNY, J. **Artificial Intelligence: A Modern Approach**. Prentice Hall/Pearson Education, 2003. (Prentice Hall series in artificial intelligence). ISBN 9780137903955. Disponível em: <<https://books.google.com.br/books?id=KI2WQgAACAAJ>>.

SHARMA, S.; SHARMA, S.; ATHAIYA, A. Activation functions in neural networks. **towards data science**, v. 6, n. 12, p. 310–316, 2017.

SISTA, R.; SINGH, R.; KUMAWAT, S. K.; DHANARE, R. Techniques used by e-commerce industries for customer analysis. In: IEEE. **2021 International Conference on Computer Communication and Informatics (ICCCI)**. [S.l.], 2021. p. 1–6.

TATO, A.; NKAMBOU, R. Improving adam optimizer. 2018.

TIAN, Y.; STEWART, C. History of e-commerce. In: **Encyclopedia of e-commerce, e-government, and mobile commerce**. [S.l.]: IGI Global, 2006. p. 559–564.

- TOMAN, M.; TESAR, R.; JEZEK, K. Influence of word normalization on text classification. **Proceedings of InSciT**, v. 4, p. 354–358, 2006.
- ULLAL, M. S.; SPULBAR, C.; HAWALDAR, I. T.; POPESCU, V.; BIRAU, R. The impact of online reviews on e-commerce sales in india: A case study. **Economic Research-Ekonomska Istraživanja**, Taylor & Francis, v. 34, n. 1, p. 2408–2422, 2021.
- XING, F. Z.; PALLUCCHINI, F.; CAMBRIA, E. Cognitive-inspired domain adaptation of sentiment lexicons. **Information Processing & Management**, Elsevier, v. 56, n. 3, p. 554–564, 2019.
- XU, K.; BA, J.; KIROS, R.; CHO, K.; COURVILLE, A.; SALAKHUDINOV, R.; ZEMEL, R.; BENGIO, Y. Show, attend and tell: Neural image caption generation with visual attention. In: PMLR. **International conference on machine learning**. [S.l.], 2015. p. 2048–2057.
- XU, T.; PENG, Q.; CHENG, Y. Identifying the semantic orientation of terms using s-hal for sentiment analysis. **Knowledge-Based Systems**, Elsevier, v. 35, p. 279–289, 2012.
- YANG, L.; LI, Y.; WANG, J.; SHERRATT, R. S. Sentiment analysis for e-commerce product reviews in chinese based on sentiment lexicon and deep learning. **IEEE access**, IEEE, v. 8, p. 23522–23530, 2020.
- YU, L.-C.; WU, J.-L.; CHANG, P.-C.; CHU, H.-S. Using a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news. **Knowledge-Based Systems**, Elsevier, v. 41, p. 89–97, 2013.
- ZHANG, B.; XU, X.; LI, X.; CHEN, X.; YE, Y.; WANG, Z. Sentiment analysis through critic learning for optimizing convolutional neural networks with rules. **Neurocomputing**, Elsevier, v. 356, p. 21–30, 2019.