

Instant News Check: Uma extensão de consulta de *Fake News*

Leonardo Lima Nascimento, Edgar Noda, André Constantino da Silva

¹Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas –
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) –
13.183-250 – Hortolândia – SP – Brasil

leonardo.lima.nas49@gmail.com, edgar.noda@ifsp.edu.br

andre.constantino@ifsp.edu.br

Abstract. Nowadays, the search for information using the internet has increased exponentially, with that the detection of Fake News has become a great challenge, so the objective of this article is the development of an application that can be updated daily and is be easy to use for the end user to get a Fake News checkpoint. On this article was created a browser extension and a database of Fake News. The database is updated with news that were released and analysed by 5 fact agencies, those agencies perform a analyse of the Fake News. Was also created a browser extention were the end user when notice that what he is reading may be a Fake News, he can go to the extension and it will collect the news details and it perform a seach on the database that if find a relative data will return to the user the source and link of the original fact check website and will return nothing if it doesn't find data on the database.

Resumo. Nos dias de hoje a busca por informação utilizando a internet aumentou exponencialmente, com isso a detecção de Fake News se tornou um grande desafio, assim o objetivo desse artigo é o desenvolvimento de uma aplicação que possa ser atualizada diariamente e seja de fácil utilização para o usuário final conseguir ter um ponto de verificação de Fake News. No trabalho foi realizado a criação de uma extensão de navegador e a criação de uma base de dados de Fake News. A base de dados é populada por notícias divulgadas em 5 agências de checagem que fazem a análise de notícias falsas e assim foi criado também uma extensão de navegador onde quando o usuário estiver acessando uma notícia que entender que pode ser falsa o mesmo pode verificar se o texto ou notícia consta no banco de dados de Fake News e se sim retorna para o usuário o site e o endereço para verificar a informação.

1. Introdução

A disseminação das notícias em mídias digitais possibilita que a informação seja propagado de uma maneira muito mais fácil, segundo um estudo realizado por [Luminate Group 2020] foi constatado que 92% dos entrevistados na pesquisa leem notícias em mídias digitais pelo menos 2 vezes por semana. Isso mostra que um número grande de pessoas procuram se informar de notícias por meio digital, o que facilita o acesso à informação pois, com um computador é possível acessar diversas notícias. Porém também traz uma grande possibilidade de disseminação de notícias falsas ou *Fake News*. De acordo com [Allcott and Gentzkow 2017] *Fake News* é definido como uma notícia falsa que é criada de forma intencional com o intuito de enganar o leitor. Este tipo

de ação não é uma invenção do mundo atual, existem relatos desse tipo de ação nos mais diversos períodos da história. Entretanto, a grande diferença nos dias de hoje é a maior facilidade e alcance que as mídias digitais proporcionam na disseminação dessas *Fake News*. segundo o estudo realizado por [IPSOS 2018], 62% dos entrevistados no Brasil responderam que já acreditaram em notícias falsas, uma porcentagem acima da média mundial que segundo o estudo é de 48%.

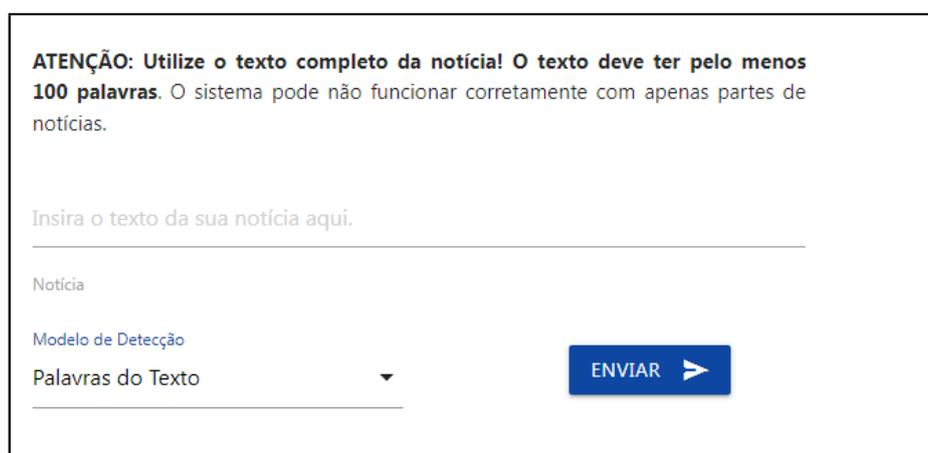
Sendo assim, este trabalho propõe a criação de uma plataforma que tem como objetivo ajudar no combate de *Fake News*, sendo essa aplicação composta de uma base de dados que será criada utilizando o projeto de *dataset*, que visa buscar em *websites* de agências verificadoras de *Fake News* e com isso criar uma base de dados com as informações, e então é proposto a criação de uma extensão que realize a busca nesta base de dados para assim auxiliar as pessoas na análise se uma notícia é de fato verdadeira ou se pode ser uma *Fake News*. Através da coleta das informações da notícia que o usuário está acessando e compare com os dados existentes no banco de dados.

2. Trabalhos Correlatos

Nesta seção foi identificado um trabalho que realiza função similar ao proposto neste trabalho, em busca de obter requisitos, êxitos e principais preocupações com objetivo de agregar a aplicação proposta neste trabalho.

2.1. Nilc-Fakenews

Nilc-*Fake News* é o resultado do trabalho realizado por [Monteiro et al. 2018], que resultou em um ambiente de consulta de *Fake News*, onde o usuário o informa ao menos 100 palavras à aplicação e assim é feito uma análise utilizando Processamento de Linguagem Natural (PLN) e Aprendizado de Máquina (AM) em busca de conteúdo enganoso e assim retornando se o que foi encontrado pode ser uma notícia falsa. Nesta aplicação o usuário precisa informar o texto e então é retornado a análise, conforme a Figura 1 e 2.



The image shows a web form for Nilc-Fakenews. At the top, there is a warning message: "ATENÇÃO: Utilize o texto completo da notícia! O texto deve ter pelo menos 100 palavras. O sistema pode não funcionar corretamente com apenas partes de notícias." Below this is a text input field with the placeholder "Insira o texto da sua notícia aqui." and a label "Notícia". Underneath the input field is a dropdown menu labeled "Modelo de Detecção" and another input field labeled "Palavras do Texto" with a small downward arrow. To the right of these fields is a blue button with the text "ENVIAR" and a right-pointing arrow.

Figura 1. Site Nilc-Fakenews

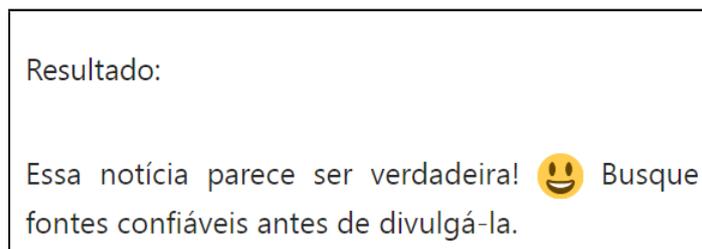


Figura 2. Resultado Site Nilc-Fakenews

3. Fundamentação Teórica

Segundo uma pesquisa realizada por [Poynter Institute 2022], no Brasil 4 de 10 pessoas afirmaram receber notícias falsas todos os dias, isso mostra o quanto a disseminação desse tipo de notícia tem se tornado algo preocupante, de acordo com a mesma pesquisa 43% das pessoas que receberam notícias falsas também já compartilharam esse tipo de notícia.

O combate a *Fake News* vem sendo um dos grandes desafios da atualidade para identificar e combater esse tipo de notícia, no âmbito da tecnologia da informação de acordo com [Pablo de Andrades Lima 2018] existe uma dificuldade no tratamento de notícias falsas em meio digital, devido as diferentes classificações de notícias e interesses mercadológicos também é citado que existem técnicas de inteligência artificial que visam combater as *Fake News* porém ainda não são funcionais.

Uma outra forma de combate as notícias falsas é através das agências de verificação, que utilizam uma técnica chamada checagem de fatos ou *fact-checking*, de acordo com [Porfírio and Marques 2022], *fact-checking* é a validação *ex post* ou seja fora depois do fato/notícia ter acontecido, através desta técnica as agências de verificação, fazem a análise das notícias e comparam com diversas fontes para identificar se é algo verdadeiro ou não.

4. Materiais e Métodos

Nesta seção abordaremos os Materiais e Métodos estudados e utilizados na elaboração deste trabalho.

4.1. Dataset FACTCK.BR

Criado por [Moreno and Bressan 2019] é um projeto para estudar *Fake News* em português, o mesmo extrai de agência verificadoras notícias falsas publicadas e armazena em um *Dataset*, o projeto também inclui um *script* para atualização do *dataset*. Esse *script* foi utilizado para criação da base de dados, esse *dataset* foi escolhido por ser dados de agências de checagem brasileiras e em português.

4.2. Processamento de Linguagem Natural (PLN)

Processamento de Linguagem Natural ou PLN faz parte do estudo da área de inteligência artificial onde tem por objetivo fazer o computador entender e interpretar a linguagem humana. Neste trabalho PLN foi utilizado na busca no banco de dados, devido a capacidade de trazer informações do mundo real para a linguagem computacional.

Em seu artigo sobre PLN [Liddy 2001] define Processamento de Linguagem natural como uma gama teoricamente motivada de técnicas computacionais para analisar e

representar textos que ocorrem naturalmente em um ou mais níveis de análise linguística com o propósito de alcançar processamento de linguagem para uma variedade de tarefas ou aplicativos.

4.3. Python

Para o desenvolvimento deste trabalho foi utilizado a linguagem Python, segundo [Python.Org 2022] Python é uma linguagem de programação interpretada, orientada a objetos e de alto nível com semântica dinâmica, é construída em alto nível de estrutura de dados combinada com tipagem dinâmica e ligação dinâmica.

Para este trabalho foi utilizado a versão 3.1 e também foram utilizados as seguintes bibliotecas: Pandas para criação e organização dos dados foi utilizada pela sua facilidade de exportação de dados e tratamento de dados, Newspaper3k utilizada para obter os dados dos sites de agências verificadoras e Flask para criação do *Web Service e Rest-API*.

4.4. Representational State Transfer (REST)

De acordo com [de Souza 2020] REST é um estilo de arquitetura web muito importante no desenvolvimento e gerenciamento de sistemas na internet. Com esse recurso, tanto o servidor quanto os clientes têm acesso a um procedimento mais rápido e fácil de conduzir. De acordo com [Fielding 2000] em sua tese de PhD, REST é um estilo híbrido derivado de vários estilos de arquitetura baseados em redes combinadas com restrições adicionais que definem uma interface de conector uniforme. Com isso REST pode ser utilizado como em integrações de aplicações por meio de interface Cliente-Servidor podendo ser assim utilizado em diferentes plataformas.

4.5. MySQL

De acordo com [MYSQL.com 2022] MySQL é o mais popular gerenciador de banco de dados relacional, sendo o mesmo de código aberto, sendo considerado como muito rápido, confiável, escalável e fácil de utilizar. Com o MySQL é possível criar e gerenciar tabelas e dados, assim utilizando algumas funções nativas como procura em linguagem natural por uma *string* em uma coleção de texto.

Foi utilizado a versão 8.0.21 do MySQL nesse trabalho e foi escolhido por conta da capacidade de execução de busca por PNL e o limite de recurso disponível na versão gratuita.

4.6. Javascript

Javascript é uma linguagem de programação desenvolvida com o objetivo de ser utilizada para a internet, onde o código criado é executado nos navegadores mais modernos. Também com javascript é possível utilizar diversas bibliotecas para o desenvolvimento.

Neste trabalho foi utilizado o objeto do Javascript chamado *XMLHttpRequest* para realizar a chamada REST-API para a base de dados.

JavaScript é a linguagem de programação da Web. A ampla maioria dos sites modernos usa JavaScript e todos os navegadores modernos – em computadores de mesa, consoles de jogos, tablets e smartphones –

incluem interpretadores JavaScript, tornando-a a linguagem de programação mais onipresente da história. JavaScript faz parte da tríade de tecnologias que todos os desenvolvedores Web devem conhecer: HTML, para especificar o conteúdo de páginas Web; CSS, para especificar a apresentação dessas páginas; e JavaScript, para especificar o comportamento delas. [Flanagan 2011]

4.7. HTML

HTML é linguagem de marcação utilizada no desenvolvimento de páginas da web, a marcação é interpretada pelos navegadores mais modernos e transformado conforme o código foi criado.

"Páginas da Web são construídas, antes de tudo, escrevendo instruções em HyperText Markup Language — Linguagem de Marcação Hipertexto (HTML). HTML é uma linguagem simples que diz ao computador como criar um documento; que vai dizer ao navegador, como o Firefox ou o Microsoft Internet Explorer, como a página da Web deve aparecer na tela. (...) [Crowder 2012]"

5. Metodologia

Nesta seção iremos abordar a metodologia utilizada para a execução deste trabalho, sendo:

- Estudo e Modificação do *script* atualizador do *dataset* FACTCK.BR.
- Criação da extensão para navegador Google Chrome e criação do *Web Service* para interação REST API.
- Integração da extensão com o *Web Service* para a realização das consultas.

6. Desenvolvimento

Nessa seção é abordado o desenvolvimento da aplicação proposta, a mesma foi criada para utilizar a arquitetura descrita na Figura 3.

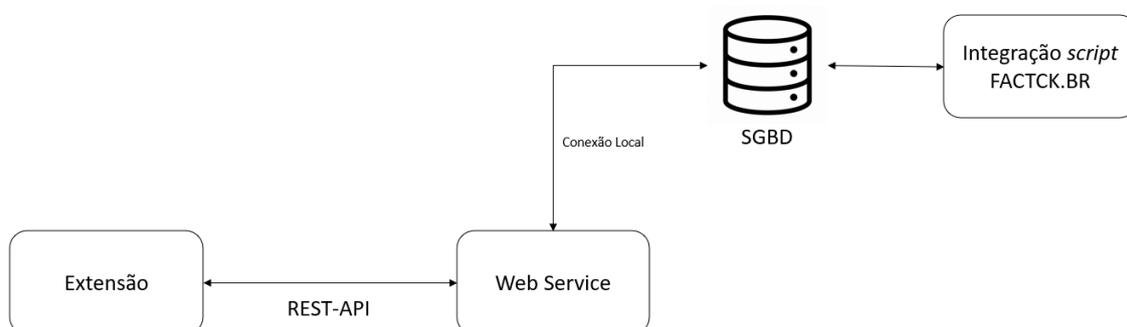


Figura 3. Arquitetura da aplicação

Essa seção foi dividida conforme os itens descritos na Figura 3:

- Integração Script FACKT.BR.

- Base de dados de *Fake News* (SGBD).
- *Web Service*.
- Extensão de navegador.

6.1. Integração Script FACTCK.BR

Para o desenvolvimento deste trabalho, foi utilizado como base o projeto FACTCK.BR criado por [Moreno and Bressan 2019] nele é proposto um *dataset* em português com dados analisados por 3 agências verificadoras, sendo elas:

- Aos Fatos (<https://aosfatos.org/>)
- Lupa (<https://piaui.folha.uol.com.br/lupa/>)
- Truco (<https://apublica.org/tag/truco/>)

Tabela 1. Informações coletadas pelo *dataset* FACTCK.BR

Coluna	Descrição
URL	Endereço web da artigo
Author	agência verificadora
datePublished	data de publicação
claimReviewed	Data de verificação
reviewBody	Corpo do artigo
Title	Título do artigo
ratingValue	Classificação da verificação
bestRating	Melhor Classificação
alternativeName	Nome Alternativo

Para obter as notícias para o *dataset*, também é integrado um código utilizando linguagem Python, o mesmo faz o acesso ao *feed* dos sites das agências de checagem e assim coletando as informações da página da notícia e guardando as informações, o *dataset* as informações são organizadas no *dataset* conforme a Tabela 1.

Durante a execução desse trabalho, houve a necessidade de fazer algumas adaptações no código do projeto FACTCK.BR. Foi necessário realizar atualização do código para fazer a comunicação do Python como *browser* e conseguir acessar as notícias e também foi alterado o código para reorganizar as informações para ser adicionado ao banco de dados.

Devido a atualização do código do site da agência verificadora Lupa (<https://piaui.folha.uol.com.br/lupa/>), infelizmente não permite a extração das notícias da forma que é feita no código original e a agência Truco (<https://apublica.org/tag/truco/>) não está sendo atualizado desde 2018, com isso ambas foram removidas deste projeto.

De acordo com [Moreno and Bressan 2019] O código original do projeto FACTCK.BR extrai os dados utilizando um esquema de dados estruturado chamado *ClaimReview* que era usado por agências de checagem de fatos para compartilhar seus resultados em mecanismos de busca, permitindo a coleta de dados em tempo real, porem

após análise realizada nos portais das agências de checagem esse esquema de dados não está sendo utilizado, então para continuar capturando as notícias foi utilizado a busca por uma estrutura chamada *Article*.

Então no projeto foi criado uma nova forma de coletar e buscar esses dados utilizando a estrutura *Article*, utilizando a biblioteca do *python Newspaper3k*, mantendo a mesma estrutura do projeto FACTCK.BR.

Podemos ver na Figura 4 o código original que coleta utilizando *ClaimReview*, onde podemos ver entre as linhas 83 e 108 onde é realizado o tratamento, localizando as *tag* do modelo *ClaimReview* e obtendo as informações.

```
83     for claimR in content.findAll('script', attrs={"type": "application/ld+json"}):
84         linha = []
85         try:
86             my_dict = text_pre_proc(claimR.get_text(strip=True))
87             if search(my_dict, 'ClaimReview'):
88
89                 linha.append(url)
90                 linha.append(my_dict['url'])
91                 linha.append(my_dict['datePublished'])
92                 linha.append(my_dict['claimReviewed'])
93                 try: linha.append(my_dict['reviewBody'])
94             except:
95                 try: linha.append(my_dict['description'])
96                 except: linha.append('Empty')
97             for title in re_char(content.title.get_text().replace('<title>', '').replace('</title>', '')):
98                 linha.append(title)
99                 break
100             try: linha.append(my_dict['reviewRating']['ratingValue'])
101             except: linha.append('Empty')
102             try: linha.append(my_dict['reviewRating']['bestRating'])
103             except: linha.append('Empty')
104             try: linha.append(my_dict['reviewRating']['alternateName'])
105             except: linha.append('Empty')
106             try: linha.append(my_dict['itemReviewed']['@type'])
107             except: linha.append('Empty')
108             break
```

Figura 4. Código Original

Na Figura 5 que coleta utilizando *Article* podemos ver entre as linhas 126 e 140 o cabeçalho da biblioteca *Newspaper3k* realizando o *download* da notícia acessada e nas linhas seguintes obtendo as informações.

```
126     toi_article = Article(url, language="pt")
127     toi_article.download()
128     toi_article.parse()
129     toi_article.nlp()
130
131     linha.append(url)
132     linha.append(toi_article.source_url)
133     linha.append(toi_article.publish_date.strftime("%Y-%m-%d"))
134     linha.append(re_char(toi_article.summary))
135     linha.append(re_char(toi_article.text))
136     linha.append(re_char(toi_article.title))
137     linha.append('Empty')
138     linha.append('Empty')
139     linha.append('Empty')
140     linha.append('Empty')
```

Figura 5. Código Novo

O novo código permitiu a integração de outras agências verificadoras, como dito anteriormente foi removido do projeto as agências Lupa (<https://piaui.folha.uol.com.br/lupa/>) e Truco (<https://apublica.org/tag/truco/>), porém como mantivemos o código original conseguimos continuar obtendo os dados da agência Aos Fatos (<https://aosfatos.org/>) e adicionar mais três agências como podemos ver abaixo:

- Aos Fatos (<https://aosfatos.org/>)
- G1 - Fato ou Fake (<https://g1.globo.com/fato-ou-fake/>)
- Projeto Comprova (<https://projetocomprova.com.br/publicações/>)
- e-farsas (<https://www.e-farsas.com>)
- Boatos (<https://www.boatos.org>)

Um desafio também foi ajustar o código que realiza a atualização do *dataset* FACTCK.BR pois o mesmo é gerado o *dataset* e salvo em formato TSV, que é uma forma de armazenamento em texto, dificultando a leitura para uma integração com uma aplicação web.

Assim para criação da base de dados proposta neste trabalho o código que realiza a atualização do *dataset* foi alterado para ao invés de salvar o *dataset* em formato TSV, o mesmo seja carregado em um banco de dados relacional.

Houve então a modificação e adição ao código em Python para que seja realizada a conexão do banco de dados e seja feita a atualização da base. Neste trabalho foi utilizado o MySQL para criação do SGBD, assim no MySQL foi criada uma tabela que foi chamada de *Fake News* mantendo o mesmo formato conforme a Tabela 1, na próxima seção irá ser descrito como o banco será utilizado.

6.2. Criação da Base de dados de *Fake News*

O próximo passo para o desenvolvimento da aplicação proposta é a criação da base de dados de *Fake News* utilizando as alterações propostas realizadas na seção anterior, foi criado um código que faz o controle do banco de dados e realiza a atualização e procura na base.

Como foi mantido as informações originais do código do projeto FACTCK.BR, ele cria um *dataset* utilizando a biblioteca pandas do *Python*, para organização dos dados, essa biblioteca também permite realizar conexão com o banco de dados e realizar a adição de dados para o banco de forma simples.

```

206     new_entries = pd.DataFrame(claimList, columns=toprow)
207     new_entries = new_entries.set_index('URL')
208
209
210     print("Atualizando Banco de dados")
211     result=Database.updatedatabase(new_entries)

```

Figura 6. Final do Código de Atualização

Como podemos ver na Figura 6, que é atualização do código do projeto FACTCK.BR que realiza o *download* das notícias novas das agências, podemos ver na linha 211 em que é chamado a classe *database* e o seu método para fazer a atualização.

Agora na Figura 7 mostra o método de atualização dos dados, onde é recebido por parâmetro as notícias novas, nas linhas 38 até 45 é realizado a conexão com o banco de dados, nas linha 47 e 48 através da biblioteca pandas é realizado a conexão e *upload* em uma tabela temporária no banco de dados; e então entre as linhas 50 e 55 é inserido na tabela principal as notícias que já não existem nele.

Para solução que realiza a busca no banco de dados, foi utilizada a função nativa do MySQL chamada *FULLTEXT* com pesquisa em linguagem natural, que realiza a busca

```

37     def updatedatabase(data):
38         mydb = mysql.connector.connect(
39             host="localhost",
40             user="root",
41             password="",
42             database="fakenews"
43         )
44
45         cursor = mydb.cursor()
46
47         engine = create_engine("mysql+pymysql://root:@localhost/fakenews?charset=utf8mb4")
48         data.to_sql('temporary_table', engine, if_exists='replace')
49
50         sql = "INSERT INTO fakenews2 SELECT * FROM temporary_table A WHERE NOT EXISTS (SELECT 1 FROM fakenews2 X
51             WHERE A.URL = X.URL)"
52         cursor.execute(sql)
53         affected_rows = "Foram Adicionados {} novas noticias ao Banco de dados".format(cursor.rowcount)
54         affected_rows += " em " + datetime.now().strftime("%d/%m/%Y as %H:%M:%S")
55         mydb.commit()
56         return affected_rows

```

Figura 7. Código de atualização do banco de dados

```

class Database:
    def searchDatabase(keyword):
        mydb = mysql.connector.connect(
            host="localhost",
            user="root",
            password="",
            database="fakenews"
        )

        mycursor = mydb.cursor()

        mycursor.execute("SELECT URL, title, author, (MATCH (title,claimReviewed,reviewBody,url)
            AGAINST ('+' + keyword + "' IN NATURAL LANGUAGE MODE)) as relevance FROM fakenews1 where ((MATCH
            (title,claimReviewed,reviewBody,url) AGAINST ('+' + keyword + "' IN NATURAL LANGUAGE MODE))>8)
            order by relevance desc limit 5;")

        myresult = mycursor.fetchall()

        return myresult

```

Figura 8. Método de consulta no banco de dados

no banco de dados e retorna as 5 melhores pesquisas similares ao que foi pesquisado, utilizando o critério de relevância do MySQL e validando se a relevância é maior que 0 isso significa que existe alguma semelhança entre os dados procurados e os registros do banco de dados. Conforme o código na Figura 8, recebendo como parâmetro as palavras chaves que é obtido pelo *web service* que será abordado na próxima seção. O método irá retornar o que foi encontrado no banco de dados ou vazio se não for encontrado resultado.

6.3. Web Service

Nessa parte do desenvolvimento do trabalho foi realizada a criação de um *Web Service* para fazer a integração entre a extensão e banco de dados utilizando REST-API.

Para isso foi utilizado a linguagem Python e a biblioteca Flask, criando um servidor que a extensão encaminhará o *URL* da notícia e então como podemos ver na linha 18 da Figura 9 o *web service* receberá a informação.

Podemos ver na linha 19 da Figura 9, primeiramente é executado o método *get-claimreview* passando como parâmetro a URL recebida, e então esse método conforme a Figura 10 mostra a utilização da biblioteca *Newspaper3k*, é então realizado o *download* da pagina informada e utilizando a biblioteca *spacy* que é própria para processamento de linguagem natural e seu modelo treinado padrão em português para obter as palavras chaves da notícia que foi recebida da extensão e então retorna essa informação ao *web*

service. O *Web Service* então faz executar o método *seachdatabase* que foi explicado na seção anterior e é realizada a busca na base de dados e então é retornado o que foi ou não encontrado. Se foi encontrado algo é retornado os dados em formato *json* para a extensão junto com o código 200 para assim ela fazer o tratamento e exibir ao usuário. Se não é encontrado nenhuma informação é retornado o código 201 sem nenhum dado.

```
9 app = Flask(__name__)
10 CORS(app)
11 api = Api(app)
12
13 @app.route("/")
14
15 class Search(Resource):
16     def get(self):
17         args = request.args
18         url = args.get('url')
19         data = Database.seachDatabase(Claim.get_claimReview(url))
20         if not data:
21             return "", 201 #retrun data was not found.
22         else:
23             df = pd.DataFrame(data,columns=['URL', 'title', 'author', 'relevance'])
24             data1 = df.to_json(orient='index')
25             parsed = json.loads(data1)
26             return parsed, 200 # return data and 200 OK code
27
28 api.add_resource(Search, '/search')
29
30
31 if __name__ == '__main__':
32     app.run() # run our Flask app
```

Figura 9. Código da Rest-API / *Web Service*

```
10 # Get ClaimReview
11 def get_claimReview(url):
12
13     toi_article = Article(url, language="pt")
14     toi_article.download()
15     toi_article.parse()
16     toi_article.nlp()
17     nlp = spacy.load("pt_core_news_lg")
18
19     keyword=" ".join(map(str, nlp(toi_article.text).ents))
20
21     return keyword
```

Figura 10. Método *get-claimreview*

6.4. Extensão de navegador

O próximo passo para o desenvolvimento da aplicação proposta foi a criação da extensão para o navegador Google Chrome. A ideia principal para a aplicação é quando o usuário estiver em uma notícia que ele desconfie ser uma *Fake News*, o mesmo utilizar a extensão para consultar se existe algo sobre aquela notícia no banco de dados criado.

Como o primeiro passo para criação da extensão foi realizado a criação do arquivo manifesto, que seria um arquivo com as informações iniciais para o desenvolvimento da extensão.

Conforme podemos ver na Figura 11, entre as linhas 2 e 9 é colocado algumas informações importantes do arquivo como Nome, versão e a permissão necessária para a extensão no navegador.

```

1 {
2   "manifest_version": 3,
3   "name": "instant-newsck",
4   "version": "0.1.0",
5   "permissions": [
6     "tabs"
7   ],
8   "browser_action":
9     { "default_popup": "index.html" }
10  }

```

Figura 11. Imagem do arquivo manifesto

Em seguida foi realizado a criação da tela da extensão, ou seja aquilo que o usuário final irá utilizar como podemos ver na Figura 12. Nessa etapa foi utilizando HTML e CSS para criação.



Figura 12. Tela Inicial da extensão

Após o usuário selecionar a opção para verificação a notícia pode ser encontrada no banco de dados ou não, como podemos ver na Figura 13 quando é encontrado alguma notícia no banco de dados que pode estar relacionado a noticia que está sendo pesquisada.

Essa Noticia pode ser FALSA!
Encontramos os registros abaixo em agencias verificadoras:

Agencia	Titulo	Rel	Link da Materia
 aosfatos.org	Governo Bolsonaro não suspendeu distribuição de insulina no SUS Aos Fatos	103	Visitar Site

Figura 13. Tela quando encontrado

Também podemos conferir na Figura 14 como é o retorno quando não é encontrado no banco de dados.

Não foi encontrado em
nosso banco de dados
nenhuma informação
sobre essa notícia

Figura 14. Tela quando não encontrado

```
14 changeColor.addEventListener("click", () => {
15     chrome.tabs.query({active: true, lastFocusedWindow: true}, function(tabs) {
16
17         document.getElementById("loader").hidden=false;
18         document.getElementById("changeColor").hidden=true;
19
20         try{
21
22             var tab = tabs[0];
23             var theUrl = "http://127.0.0.1:5000/search?url="+tab.url
24
25             var xmlhttp = new XMLHttpRequest();
26             xmlhttp.open( "GET", theUrl, false ); // false for synchronous request
27             xmlhttp.send( null );
28
29             var result=xmlhttp.responseText
30             var code=xmlhttp.status
31
32             document.getElementById("changeColor").hidden=true;
33             document.getElementById("loader").hidden=true;
34
```

Figura 15. Método Código javascript para solicitar dados

Além da parte visual a extensão internamente utiliza a linguagem Javascript para obter os dados do *Web Service*, podemos ver na Figura 15 onde é realizado a chamada de REST-API para o *Web Service* para obter os dados encontrados no banco de dados.

Nas as Figuras 16 e 17 é mostrado como é realizado o tratamento das informações e a exibição para o usuário.

```
if(code == 200){
    document.getElementById("text3").innerHTML="Essa Notícia pode ser FALSA! <br> Encontramos os
    registros abaixo em agencias verificadoras:";
    const jason = JSON.parse(result)

    fetch = document.getElementById('fetch');
    for(var i=1; i <=Object.keys(jason).length; i++){

        var newRow = fetch.insertRow(fetch.length);
        var cell1 = newRow.insertCell(0);
        var cell2 =newRow.insertCell(1);
        var cell3 =newRow.insertCell(2);
        var cell4 =newRow.insertCell(3);
        j=i-1
    }
}
```

Figura 16. Método em javascript para tratar dados quando encontrado na base de dados

```
document.getElementById("text3").innerText="Não foi encontrado em nosso banco de dados nenhuma
informação sobre essa notícia";
```

Figura 17. Método Código javascript para tratar dados quando não encontrado na base de dados

7. Testes de pesquisa na base de dados

Como foi explicado na sessão de desenvolvimento para a criação da base de dados, também foi executado uma série de testes para identificar a qualidade e assertividade da ferramenta de pesquisa desenvolvida, para isso foram selecionadas 10 registros de agências verificadoras que estão no banco de dados e 10 registros similares encontrados em pesquisa realizada na *internet*. Para classificar o teste iremos utilizar a relevância que o MySQL irá retornar para a pesquisa. De acordo com [Dev.mysql.com 2022], a relevância no MySQL é um número positivo de ponto flutuante que determina a similaridade das palavras chaves pesquisados com o texto comparado no banco de dados. Quando a relevância é zero, não há semelhança. A relevância é calculada pelo MySQL com base no número de palavras no documento, o número de palavras únicas no documento, o número total de palavras na coleção e o número de linhas que contêm uma palavra em particular.

Como podemos ver na Tabela 2 o resultado da relevância, sendo o menor resultado recebido sendo 60, mostrando que a pesquisa está sendo realizada e retornando resultados com relevância alta.

Tabela 2. Comparação de páginas e relevância recebida.

Link da notícia na base de dados	Link da notícia similar	Relevância Calculada
https://projetoaprova.com.br/publica%C3%A7%C3%B5es/e-falso-que-lula-nao-possa-ser-diplomado-petista-teve-condenacoes-anuladas-em-2021/	https://www.cnnbrasil.com.br/nacional/e-falso-que-lula-nao-possa-ser-diplomado-petista-teve-condenacoes-anuladas/	284
https://projetoaprova.com.br/publica%C3%A7%C3%B5es/video-foi-manipulado-para-mostrar-bolsonaro-a-frente-em-pesquisa-do-ipeec/	https://www.cnnbrasil.com.br/nacional/video-foi-editado-para-mostrar-bolsonaro-a-frente-na-pesquisa-ipeec-de-2o-turno/	256
https://apublica.org/2022/09/regioes-com-maior-concentracao-de-populacao-negra-tem-menos-acesso-a-energia-eletrica/	https://observatoriodaenergia.wordpress.com/2022/09/21/regioes-com-maior-concentracao-de-populacao-negra-tem-menos-acesso-a-energia-eletrica/	226
https://www.boatos.org/politica/mourao-escreveu-texto-no-estadao-que-comeca-com-nao-vi-nosso-presidente-proibir-nem-obrigar-ninguem-se-vacinar.html	https://politica.estadao.com.br/blogs/estadao-verifica/texto-que-defende-postura-de-bolsonaro-na-pandemia-nao-e-de-mourao-nem-foi-publicado-no-estadao/	149
https://g1.globo.com/fato-ou-fake/eleicoes/noticia/2022/09/20/e-fake-audio-atribuido-a-bolsonaro-em-que-ele-xinga-michelle-antes-do-desfile-de-7-de-setembro.ghtml	https://www.correiobraziliense.com.br/holofote/2022/09/5036730-e-falso-audio-em-que-presidente-xinga-michelle-bolsonaro-antes-do-desfile-de-7-de-setembro.html	184
http://www.e-farsas.com/homem-morre-ao-jogar-pino-ao-inves-da-granada-durante-protesto-sera-verdade.html	https://monitor7.r7.com/e-farsas/homem-morre-ao-jogar-pino-ao-inves-da-granada-durante-protesto-sera-verdade-23082022	116
https://g1.globo.com/fato-ou-fake/eleicoes/noticia/2022/09/19/e-fake-que-freixo-disse-no-rj1-que-policiais-nao-poderao-atirar-durante-operacao-em-favelas.ghtml	https://rdmonline.com.br/ao-rj1-freixo-diz-que-fara-operacao-policia-em-favelas-mas-que-helicoptero-nao-pode-atirar-trecho/	92
https://apublica.org/checagem/2017/08/truco-globo-nao-repassa-so-10-do-crianca-esperanca-para-a-unesco/	https://g1.globo.com/fato-ou-fake/noticia/2021/08/23/e-fake-que-globo-repassa-10percent-do-arrecadado-no-crianca-esperanca-a-unesco.ghtml	82
https://www.aosfatos.org/noticias/telegram-nao-disse-que-hackers-alteraram-mensagens-vazadas-da-lava-jato/	https://noticias.uol.com.br/confere/ultimas-noticias/2019/06/22/telegram-nao-admitiu-que-mensagens-divulgadas-da-lava-jato-foram-alteradas.htm	65
https://www.aosfatos.org/noticias/patricia-lelis-nao-denunciou-glauber-braga-por-agressao-e-ameaca/	https://esportes.yahoo.com/noticias/lupa-glauber-lelis-agressao-falso-125701082.html	60

Contudo os dados também retornaram 5 resultados para todas as buscas assim existe possibilidade de melhoria na busca de melhores resultados.

8. Conclusão

Com o grande aumento do uso das mídias digitais das pessoas para obter informação e consumir notícias, existe a necessidade da criação de ferramentas que possam ajudar as pessoas a obter um olhar crítico sobre o que está lendo, e com esse trabalho foi possível desenvolver uma ferramenta que expande as possibilidades e ajuda a combater as notícias falsas.

Assim os objetivos desse trabalho foram concluídos. Através da pesquisa e código do projeto de [Moreno and Bressan 2019], foi possível aumentar o escopo do mesmo aumentando o numero de agências verificadoras no projeto e assim possível criar uma aplicação completa. Com um *Back-end* (*Web Service*, Banco de dados e comunicação em REST-API) e um *Front-End* (Extensão de navegador), possibilitando também de uma forma relativamente simples adicionar novas agências de verificação, onde sites que utilizam a mesma arquitetura para organizar as noticias podem ser adicionados, o que pode ajudar a construir uma base de dados imensa sobre *Fake News* ajudando no combate desse tipo de notícia.

Com o desenvolvimento desse trabalho foi possível colocar em prática diversos conhecimentos adquiridos no curso, como desenvolvimento web e criação de banco de dados e assim evoluí-los, aplicando-os ao criar uma aplicação que pode ajudar na diminuição da propagação de *Fake News*.

9. Trabalhos Futuros

Com a execução desse trabalho foi identificado algumas oportunidades futuras para integração ao projeto, podemos verificar abaixo:

- Incluir outras agências verificadas ao projeto;
- Expandir a extensão para outros navegadores e plataformas, como *mobile* e verificação em *posts* de redes sociais;
- Integração para além das agências verificadoras com inteligência artificial e outras plataforma como o Nilc-Fakenews;
- Realizar análise de quais recursos computacionais seriam necessários para o projeto em grande escala, como capacidade de processamento e armazenamento;
- Obter melhorias na velocidade e qualidade das pesquisas no banco de dados.

Referências

- Allcott, H. and Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–36.
- Crowder, A. (2012). *Construindo Web Sites Para Leigos - Tradução da 4ª Edição*, page 17. Para Leigos. Alta Books Editora.
- de Souza, I. (2020). Saiba o que é rest (representational state transfer) e como usá-lo neste tutorial. <https://rockcontent.com/br/blog/rest/>. [Online; Acessado em 09 de junho de 2022].

- Dev.mysql.com (2022). Natural language full-text searches. <https://dev.mysql.com/doc/refman/8.0/en/fulltext-natural-language.html>. [Online; Acessado em 30 de novembro de 2022].
- Fielding, R. T. (2000). Chapter 5: Representational state transfer (rest). architectural styles and the design of network-based software architectures (ph.d. *University of California, Irvine*).
- Flanagan, D. (2011). *JavaScript: O Guia Definitivo*, page 19. Bookman Editora.
- IPSOS (2018). Fake news, filter bubbles, post-truth and trust. <https://www.ipsos.com/sites/default/files/ct/news/documents/2018-09/fake-news-filter-bubbles-post-truth-and-trust.pdf>. [Online; Acessado em 09 de junho de 2022].
- Liddy, E. D. (2001). Natural language processing.
- Monteiro, R. A., Santos, R. L. S., Pardo, T. A. S., de Almeida, T. A., Ruiz, E. E. S., and Vale, O. A. (2018). Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In Villavicencio, A., Moreira, V., Abad, A., Caseli, H., Gamallo, P., Ramisch, C., Gonçalo Oliveira, H., and Paetzold, G. H., editors, *Computational Processing of the Portuguese Language*, pages 324–334, Cham. Springer International Publishing.
- Moreno, J. and Bressan, G. (2019). Factck.br: A new dataset to study fake news. In *Anais do XXV Simpósio Brasileiro de Sistemas Multimídia e Web*, pages 525–527, Porto Alegre, RS, Brasil. SBC.
- MYSQL.com (2022). What is mysql? <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>. [Online; Acessado em 09 de junho de 2022].
- Pablo de Andrades Lima, A. (2018). Existem ferramentas digitais capazes de reduzir a disseminação das fake news? https://guri.unipampa.edu.br/uploads/evt/arq_trabalhos/17804/seer_17804.pdf. [Online; Acessado em 26 de outubro de 2022].
- Porfírio, R. A. d. C. and Marques, A. (2022). Fake news e fact-checking: Mapeamento de práticas de verificação utilizadas por agências de checagem sul-americanas. <https://bdtd.ucb.br:8443/jspui/bitstream/tede/2961/2/RachelAdieneDissertacao2022.pdf>. [Online; Acessado em 06 de Novembro de 2022].
- Python.Org (2022). What is python? executive summary. <https://www.python.org/doc/essays/blurbs/>. [Online; Acessado em 09 de junho de 2022].
- Luminate Group (2020). Consumo e pagamento por notícias digitais: Oportunidades e desafios do modelo de assinaturas na américa latina. <https://luminategroup.com/storage/981/Reporte-Consumo-e-Pago-de-Noti%CC%81cias-Digitais-Brasil-%28PT%29---Luminate-2020.pdf>. [Online; Acessado em 09 de junho de 2022].
- Poynter Institute (2022). A global study on information literacy understanding generational behaviors and concerns around false and misleading information online. <https://www.poynter.org/wp-content/uploads/2022/08/>

A-Global-Study-on-Information-Literacy-1.pdf. [Online; Acessado em 26 de outubro de 2022].

Documento Digitalizado Público

Artigo TCC - Leonardo Lima Nascimento

Assunto: Artigo TCC - Leonardo Lima Nascimento
Assinado por: Edgar Noda
Tipo do Documento: Anexo
Situação: Finalizado
Nível de Acesso: Público
Tipo do Conferência: Documento Digital

Documento assinado eletronicamente por:

- **Edgar Noda, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 17/12/2022 09:36:00.

Este documento foi armazenado no SUAP em 17/12/2022. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1189352

Código de Autenticação: 538010160f

