FÁBIO TAMARU NAKAMURA
HUGO MARTINS DA CRUZ
JEAN LEE BERNARDES

LIBRA: SISTEMA INTERATIVO DE COLETA, ANOTAÇÃO E DETECÇÃO AUTOMÁTICA DE FAKE NEWS

São Paulo
2022

FÁBIO TAMARU NAKAMURA
HUGO MARTINS DA CRUZ
JEAN LEE BERNARDES

LIBRA: SISTEMA INTERATIVO DE COLETA, ANOTAÇÃO E DETECÇÃO AUTOMÁTICA DE FAKE NEWS

Trabalho apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro de Computação.

Orientador:
Prof. Dr. Fábio Levy Siqueira

Coorientadora:
Profa. Dra. Roseli de Deus Lopes

São Paulo
2022

RESUMO

Atualmente, as *fake news*, difundidas em redes sociais, são utilizadas como parte de estratégias de campanhas políticas, interferindo negativamente em processos democráticos. No meio acadêmico, a detecção automática de desinformações utiliza técnicas de NLP e *deep learning* e se consolida como uma área emergente. Contudo, não é comum ainda encontrar projetos que aproximem os avanços científicos ao público de forma interativa. Além disso, há pouca diversidade de bases públicas e anotadas, principalmente em domínios mais restritos, como línguas diferentes do inglês. Este trabalho tem o objetivo de mitigar essas deficiências com um sistema *open source* que auxilie no combate à disseminação de *fake news*, acessível por meio de um *bot* no Twitter e capaz de gerar um *dataset* anotado com seu uso. Para isso, foram utilizados serviços de computação em nuvem e um modelo de aprendizado de máquina baseado em características de propagação. Os resultados do projeto se aproximam da especificação de requisitos e dos objetivos propostos, porém não os cumprem em sua completude devido a limitações impostas pela plataforma do Twitter ao final do ciclo de desenvolvimento. Assim, conclui-se que há necessidade de adaptações e melhorias para reduzir a dependência da solução de sistemas de terceiros.

Palavras-Chave – *Fake news*, redes sociais, NLP, *deep learning*, *open source*, *bot*, Twitter, *dataset*, aprendizado de máquina, propagação.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A internet trouxe mudanças na forma de se comunicar, de se informar e de interagir com o mundo. No Brasil, de acordo com o *Brazil Digital Report* da *We Are Social* (KEMP, 2020), somos mais de 150 milhões de usuários da internet, sendo que, destes, 140 milhões fazem uso de redes sociais. As grandes redes, como Twitter, Facebook, WhatsApp e Instagram, se tornaram partes de nossa rotina, acelerando a velocidade em que as informações são propagadas, o que traz facilidades, mas também diversas questões e problemas.

Nesse sentido, várias instâncias do contexto político, como debates, campanhas, comunicações de órgãos oficiais e organização política no geral, também se transferiram para o digital. As redes sociais se tornaram importantes ferramentas de formação de opinião pública e política. Um estudo publicado em 2019 pelo Data Senado (BAPTISTA, 2019), órgão de pesquisas e análises do Senado Federal, revelou que 83% dos entrevistados compartilham da opinião de que redes sociais têm muita influência sobre a opinião das pessoas e 45% afirmaram ter decidido o voto levando em consideração informações vistas em alguma rede social. Dos entrevistados, 79%, por exemplo, dizem sempre utilizar o WhatsApp como fonte de informação e 83% dizem já ter identificado uma notícia falsa nas redes sociais. Essa mudança no paradigma da opinião pública e política trouxe consigo e potencializou diversos problemas presentes na vida real, como polarização e discurso de ódio. Dentre esses problemas, observou-se um aumento na propagação de desinformações, rumores e notícias falsas, que já se configuram como agentes do rumo político dos países e da democracia em si (DOURADO, 2020).

No Brasil, a eleição presidencial de 2018 foi um grande exemplo de como as *fake news* são usadas como estratégia de campanhas políticas e deturpam significativamente processos democráticos (PASQUINI, 2018; TRE-MT, 2018; CPOP, 2018; VALENTE, 2018). Esse uso se reflete no aumento da atuação de agências de checagem de fatos, que têm como objetivo desmentir essas notícias e divulgar esse trabalho em larga escala. Hoje, essas organizações contam com centenas de colaboradores e possuem relevância na mídia

em geral (PALACIOS, 2019). Dois exemplos são a Agência Lupa[1] e a Agência Aos Fatos[2].

Além disso, a comunidade científica observou um aumento no número de publicações científicas envolvendo ferramentas que utilizam aprendizado de máquina para combater o problema, fazendo com que a detecção automática de *fake news* se configure como uma área emergente de interesse no meio acadêmico (ISLAM et al., 2020). Porém, ainda não é comum, principalmente no Brasil, encontrar projetos que buscam implementar esse conhecimento, isto é, que propõem soluções para trazer benefícios mais próximos do grande público.

Abordagens recentes para o problema de detecção de *fake news* em redes sociais são usualmente baseadas em arquiteturas de redes neurais profundas, utilizando processamento de linguagem natural (OSHIKAWA et al., 2018) e processamento de grafos (HAN et al., 2020). Essas técnicas se propõem a analisar o conteúdo das informações disseminadas e a forma como se propagam para classificar e gerar insumos explicáveis na identificação de *fake news*. Para isso, elas processam grandes volumes de dados textuais e dados não estruturados para treino, validação e implementação dos modelos (ISLAM et al., 2020).

As técnicas modernas de processamento de linguagem natural ainda enfrentam um grande desafio: na mesma medida em que necessitam de muitos dados para treinamento, há pouca diversidade de *datasets* anotados para o seu desenvolvimento, principalmente em domínios mais restritos, como o tratado neste trabalho, e em idiomas diferentes do inglês (HEDDERICH et al., 2020). No geral, essas bases necessitam de uma etapa de anotação manual humana, que torna o processo pouco escalável e atrasa o desenvolvimento de algoritmos mais robustos em nível nacional.

1.1 Objetivo

O objetivo deste trabalho é gerar um sistema que auxilie no combate à disseminação de *fake news* em redes sociais, desenvolvido em conjunto com o grupo interdisciplinar de pesquisa IBRA USP, descrito com mais detalhes na Seção 4.3. Esse objetivo se divide em duas frentes complementares:

1. Frente primária: tem como alvo o grande público das redes sociais exposto à disseminação de *fake news*, na forma de um *bot* interativo no Twitter que alerte e auxilie

[1] https://lupa.uol.com.br/
[2] https://www.aosfatos.org/

os usuários na identificação de *fake news*.

2. Frente secundária: tem como objetivo agregar valor à comunidade científica que desenvolve modelos de aprendizado de máquina no contexto de *fake news*, na forma de bases de dados anotadas, *insights* e do código fonte do sistema capaz de gerar essas bases.

Em sua frente primária, o projeto propõe uma solução que atenderá os usuários que consomem notícias em redes sociais, na forma de um sistema que analisa as chances de uma notícia ser falsa e disponibiliza informações que auxiliam na tomada de decisão dos usuários. Para isso, a partir de uma marcação do perfil do *bot* na postagem de Twitter, fatores relevantes relacionados ao seu conteúdo, forma de propagação nas redes e informações sobre os perfis que estão interagindo (curtindo, compartilhando ou comentando) são indicados. Alguns exemplos dessas informações são:

- Quantidade de perfis suspeitos interagindo com a notícia: perfis com características robóticas compartilhando ou comentando a postagem, quantidade de postagens idênticas em um curto período de tempo.
- Aspectos sobre os padrões linguísticos e de difusão: detectados pela execução de um modelo de aprendizado de máquina, podendo indicar que a notícia tem alta probabilidade de ser falsa ou enganosa.

Considerando essa capacidade de coleta de dados e classificação de publicações de forma automatizada, o projeto visa ainda, em sua frente secundária, à disponibilização de um *dataset* e do código fonte (*open source*) de uma ferramenta capaz de gerar bases de dados anotadas automaticamente, que poderão ser utilizadas pela comunidade científica que estuda o tema. Esse *dataset* inclui, por exemplo, dados textuais e metadados (como data, hora, geolocalização, autor, número de curtidas, número de respostas, número de *retweets* etc) das publicações, além dos *outputs* dos modelos do sistema. O código fonte do sistema, por sua vez, é disponibilizado em conjunto com detalhes sobre a arquitetura implementada e pode ser utilizado para auxiliar a construção de ferramentas de anotação automática de *tweets* em outros contextos.

Tendo esses dois objetivos em mente, foram utilizadas técnicas de desenvolvimento e implementação (*deploy*) de algoritmos de aprendizado de máquina (mais especificamente processamento de linguagem natural e redes neurais profundas) e tecnologias ligadas a coleta, processamento e armazenamento de dados. Ademais, houve emprego de técnicas

e conceitos de engenharia de software para estruturar a arquitetura necessária para o funcionamento dos diversos módulos do sistema.

1.2 Justificativa

No Brasil, as soluções de combate a *fake news* e desinformação no ambiente digital envolvem principalmente agências de checagem e sites interativos de projetos.

As agências de checagem, como a Agência Lupa e a Aos Fatos, verificam em tempo real cada notícia falsa que ganha notoriedade ao se espalhar. Esse trabalho conta com a mão de obra qualificada de jornalistas e outros profissionais, porém não é muito escalável, justamente porque a verificação é realizada de forma manual (FILHO et al., 2021).

Além das agências, existem projetos digitais que atuam diretamente no nicho das *fake news*. Um exemplo relevante é o detector Fake Check (MONTEIRO et al., 2018), financiado pelo CNPq e CAPES e mantido pelo Instituto De Ciências Matemáticas e de Computação de São Carlos (ICMC). O sistema Fake Check identifica se uma notícia pode ou não ser falsa, utilizando modelos de aprendizado de máquina que se baseiam em características de escrita, como classes gramaticais mais frequentes, para identificar padrões. Para realizar uma verificação, o usuário deve colar o texto de uma notícia em uma caixa de texto do site. Feito isso, o sistema traz um veredito sobre a confiabilidade da notícia (se há alta probabilidade de ser falsa ou verdadeira), mas sem explicar como chegou no resultado. Esse projeto não tem uma interface tão próxima com o grande público, já que o usuário deve abrir o site ou o WhatsApp para acessá-lo, e não é tão prático e direto, pois é necessário copiar manualmente toda a notícia, evitando as imagens e anexos do site, por exemplo, para inseri-la no sistema.

Dessa forma, o LIBRA propõe uma ferramenta automatizada, de fácil utilização e mais próxima da realidade de utilização dos usuários de redes sociais. A ideia é que, dentro da própria interface do Twitter e a partir de uma simples marcação do perfil do projeto na postagem, o sistema ative o processamento de dados e a predição de modelos de aprendizado de máquina, fornecendo insumos explicáveis para o usuário.

Além disso, atualmente existem poucas bases de dados organizadas e anotadas ligadas ao problema das *fake news*. Podemos citar como exemplo o *dataset* brasileiro FACTCK.BR (MORENO et al., 2019). Publicado em 2019, a base contém notícias checadas manualmente por agências de checagem utilizando o ClaimReview[3]. Acredita-se na

[3]https://www.claimreviewproject.com/

possibilidade de adição de valor ao se criar mais *datasets* parecidos com este, incluindo ainda informações sobre a propagação dessas notícias em rede, e na disponibilização do código fonte de um sistema que auxilia na coleta e anotação automática dessas bases. O *bot* desenvolvido neste projeto armazena e cataloga automaticamente os *tweets* e notícias nos quais for marcado, assim como metadados referentes a sua estrutura (geolocalização, data, horário, usuários) e informações sobre as interações feitas na rede social (curtidas, comentários, *retweets*, etc).

É importante ressaltar que julga-se necessária a criação de diversas abordagens para mitigar um problema tão complexo como o das *fake news*. Acredita-se que a solução aqui proposta e as existentes são complementares, não concorrentes.

1.3 Organização do Trabalho

Este trabalho está organizado da forma descrita a seguir.

- No capítulo 2, são apresentados os principais conceitos utilizados no desenvolvimento do projeto.
- No capítulo 3, são apresentadas as histórias de usuário e os requisitos do sistema desenvolvido.
- No capítulo 4, são apresentadas as diferentes fases de desenvolvimento do projeto e a metodologia de trabalho utilizada.
- No capítulo 5, é apresentado o funcionamento, o treinamento e o teste do modelo de aprendizado de máquina utilizado, o dEFEND.
- No capítulo 6, são apresentados os aspectos da implementação e da arquitetura do sistema como um todo.
- No capítulo 7, são apresentados os testes utilizados para medir o sucesso da aplicação implementada.
- Por fim, no capítulo 8, são apresentadas as considerações finais, com possíveis trabalhos futuros.

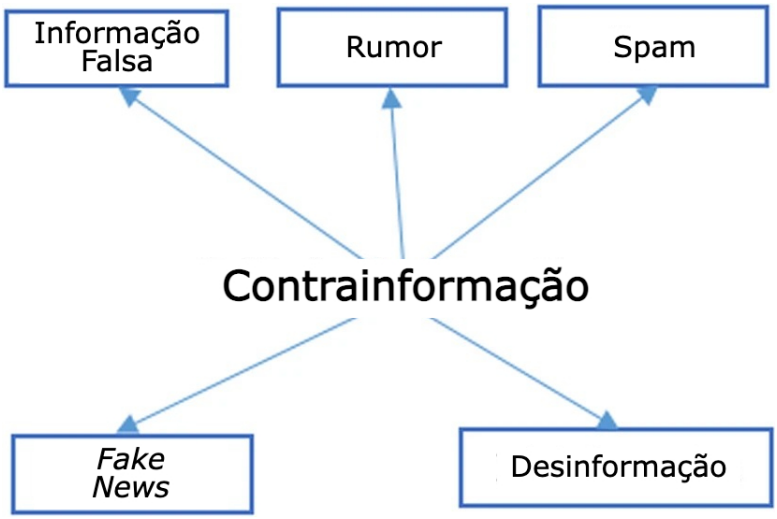
2 ASPECTOS CONCEITUAIS

Neste capítulo serão apresentados os principais conceitos utilizados neste trabalho. Por se tratar de um projeto multidisciplinar, serão abordados aspectos conceituais de diferentes áreas do conhecimento, como sociologia e jornalismo, na definição de *fake news*, e ciência de dados e engenharia de software, no âmbito do sistema.

2.1 *Fake news*

A tradução literal de *fake news* seria notícia falsa. Porém, o seu termo é mais específico. Adotando a classificação usualmente utilizada na literatura (ISLAM et al., 2020; HAMID et al., 2020) e ilustrada na figura 1, *fake news* é um tipo de contrainformação (*misinformation*), ou seja, uma subcategoria de informações incorretas e/ou enganosas. As *fake news* se diferenciam das outras classificações por distorcer ou modificar notícias reais, imitando a forma de uma noticia tradicional. A sua disseminação é, por definição, intencional e feita com um propósito específico (WU et al., 2019).

Figura 1 – Tipos de Contrainformação



Fonte: Baseada em Islam et al. (2020)

A diferença para as informações falsas (*false info*) e a desinformação (*disinformation*), também indicadas na figura 1, é sutil, pautada principalmente no formato e na intencionalidade de sua criação (WU et al., 2019). Estas não seguem necessariamente o formato de uma notícia tradicional, podendo ser veiculadas, por exemplo, como vídeos com conteúdo duvidoso. No caso das informações falsas, não há a intenção de enganar em sua criação, ao contrário do que ocorre com as *fake news* e a desinformação. Os rumores correspondem a informações não verificadas, que podem ser verdadeiras ou falsas, enquanto o *spam* é apenas uma enxurrada de informação sem relevância (WU et al., 2019). É importante ressaltar que este projeto trata especificamente do problema das *fake news*, isto é, do tipo de contrainformação que é feita de forma proposital.

2.2 Processamento de Linguagem Natural

Processamento de linguagem natural, normalmente referido como NLP (sigla em inglês para *Natural Language Processing*), é um campo da inteligência artificial que trata da interação de computadores com a linguagem humana. Um sistema de NLP tem como objetivo ler, decifrar, entender e analisar a comunicação humana para cumprir tarefas que geram valor (EISENSTEIN, 2019). Esse campo possui aplicações importantes em diversas áreas do conhecimento e possui diversos exemplos de utilização no dia-a-dia, como:

- Assistentes pessoais de voz como a Siri, a Alexa e a Cortana, que processam a voz humana para interagir com os usuários (BELLEGARDA, 2014).
- Algoritmos de busca e ranqueamento na web: retorna conteúdos baseado na relevância e similaridade com o buscado, como o algoritmo de busca do Google (NAYAK, 2019; DEVLIN et al., 2018).
- Tradutores de textos: traduzem textos, frases e palavras entre diversas línguas (WU et al., 2016).
- Sistemas de correção gramatical: revisam textos para adequá-los a normas gramaticais (HLÁDEK; STAŠ; PLEVA, 2020), como o algoritmo de revisão do Microsoft Word.

No campo científico, suas técnicas e conceitos são também estudados de forma a combater problemas complexos da digitalização, como fraudes bancárias (RODRIGUEZ,

2020; CHANG et al., 2022) e identificação de discurso de ódio para moderação de conteúdo (SILVA, 2018; POLETTO et al., 2021).

O conceito de NLP começa a ser desenhado no começo do século XX, quando Ferdinand de Saussure, um professor da University of Geneva, traz ao mundo a percepção de "*Language as a Science*" e introduz a abordagem estruturalista da língua, expandida para outras áreas, como a computação, nos próximos anos (ENGLER, 2004). Em 1950, Alan Turing descreve os testes para identificar uma máquina inteligente, e já introduz a ideia de um computador processar a linguagem humana para gerar respostas ao usuário. A partir disso, a relação entre inteligência artificial, NLP e computação começa a ser desenvolvida.

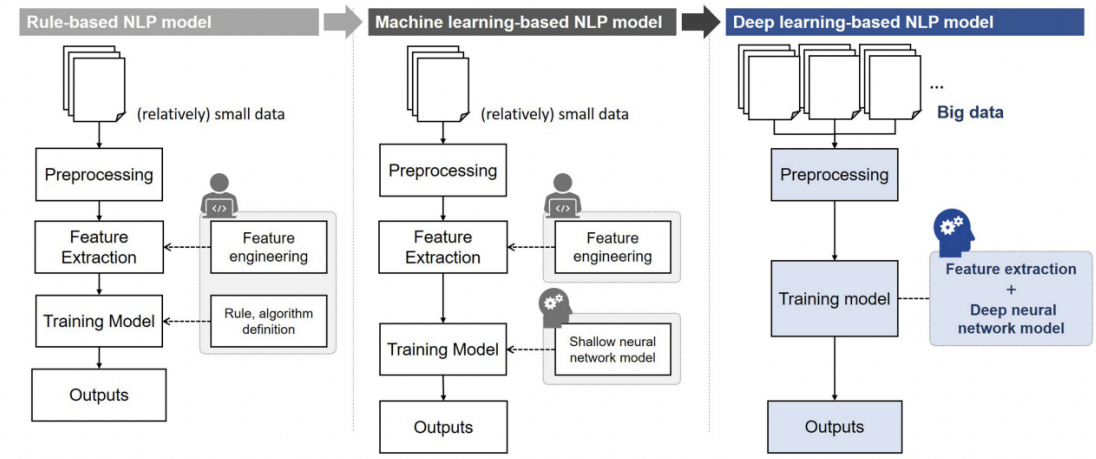
A partir dos trabalhos iniciais citados acima, os problemas de processamento de linguagem natural começaram a ser tratados com algoritmos baseados em regras (*Rule-based NLP*) (SANTAHOLMA, 2007), que se utilizam das estruturas gramaticais das línguas para extrair informações. Essas regras requerem um grande conhecimento linguístico e devem ser implementadas individualmente e de forma manual nos códigos. Nesse sentido, o sistema pode se tornar altamente complexo, porém não requer grandes quantidades de dados para funcionar (DORASH, 2017).

As abordagens seguintes são baseadas numa concepção estatística dos contextos de aplicação (*Statistical NLP*) e utilizam técnicas de aprendizado de máquina como ferramentas. Essa troca de paradigma gerou uma revolução no campo de estudo linguístico, que foi possibilitada pela evolução do poder computacional e revolução digital que o mundo observou nas últimas décadas (MAJID et al., 2021). Um exemplo de aplicação que foi possibilitada pela mudança de paradigma foi a criação dos primeiros algoritmos de tradução de texto, feitos pela empresa IBM e que utilizam técnicas estatísticas para a estimação de parâmetros do modelo (BROWN et al., 1993).

As abordagens mais atuais utilizam-se de técnicas de *deep learning*, ou aprendizado estruturado profundo, em português, para modelar abstrações de alto nível e de grandes volumes de dados (OTTER, 2020; TORFI et al., 2020).

Essas abordagens são apresentadas na figura 2, que esquematiza a estrutura de cada uma das diferentes abordagens do NLP, explicitando a evolução e as características de cada uma. A abordagem baseada em regras (*Rule-based NLP*), por exemplo, é implementada em soluções com menos dados, possui uma etapa análoga de pré-processamento de dados em relação às outras abordagens e, assim como a abordagem baseada em aprendizado de máquina (*Machine learning based NLP*), possui uma etapa manual de seleção

Figura 2 – Fluxograma da Evolução do NLP



Fonte: (SONG et al., 2020)

de parâmetros. Já a abordagem baseada em aprendizado estruturado profundo (*Deep learning-based NLP*) inclui a etapa de seleção de parâmetros na própria etapa de aprendizado. As duas últimas abordagens serão descritas com mais detalhes nas próximas seções.

2.2.1 Abordagem de Aprendizado de Máquina para Processamento de Texto

A abordagem clássica do processamento de texto baseado em aprendizado de máquina possui duas etapas iniciais que preparam os dados e embasam a criação/desenvolvimento dos modelos (SONG et al., 2020):

1. Pré-processamento: começa na etapa de aquisição dos dados e envolve diversas tarefas de codificação e simplificação do texto. A primeira etapa da simplificação é a "tokenização", onde basicamente as sentenças e palavras são separadas e transformadas em *tokens*. Em seguida alguns tratamentos são realizados, como remoção de *stopwords* (palavras com pouco significado, como artigos, pronomes e preposições) e normalização de texto (deixar todas as palavras em minúsculo).

2. Análise exploratória: busca dar uma compreensão melhor sobre o objeto de estudo. Nessa etapa, é comum observar análises de frequência de frases, palavras, letras ou números e análises de dispersão. O objetivo aqui é entender quais as características que melhor descrevem o conjunto de dados.

Essas duas etapas ajudam os desenvolvedores a entender quais serão as *features*, isto é, as propriedades mensuráveis escolhidas para serem parâmetros de entrada do modelo preditivo e outros pontos importantes na sua construção. Esse modelo utiliza-se de técnicas estatísticas ou de aprendizado de máquina como regressões, redes neurais, algoritmos bayesianos, árvores de decisão, entre outras, para realizar uma predição a respeito dos dados. Se a acurácia do modelo não for satisfatória, o desenvolvedor irá otimizar manualmente os parâmetros de entrada para chegar num resultado melhor (FERRAZ et al., 2021).

Na abordagem clássica do processamento de texto baseado em aprendizado de máquina, a etapa de identificação de *features* é feita ainda de forma manual. Além disso, algumas limitações aparecem, pois este método não se baseia na concepção de entendimento da língua pelo modelo, e sim na identificação de alguns padrões de estrutura linguística que podem se repetir (SONG et al., 2020).

2.2.2 Abordagem de Aprendizado Estruturado Profundo para Processamento de Texto

As abordagens mais atuais utilizam-se de técnicas para modelar abstrações de alto nível usando redes com várias camadas de processamento (redes profundas), compostas de diversas transformações lineares e não lineares. O aumento crescente no uso desse método se dá principalmente porque a estrutura das redes neurais profundas permite treinar modelos com um conjunto maior de dados em um menor tempo, ao mesmo tempo que adiciona várias camadas de abstração (DONG, 2021). Isso se torna possível com a evolução das unidades de processamento gráfico (GPUs), pois muitas das tarefas dessas redes podem ser paralelizadas. A grande diferença para o método explicado anteriormente é o fato de que o processo de extração de *features* e de construção do modelo são realizados em conjunto, como parte do aprendizado. Este método ainda pode ser realizado por etapas, onde o modelo aprende a extrair as *features* e depois pode ser adaptado para uma tarefa específica, processo chamado de Transferência de Aprendizado (*Transfer Learning*) (ALCOFORADO et al., 2022). Nesse sentido, a abordagem de aprendizado estruturado profundo para processamento de texto carrega consigo algumas das técnicas descritas anteriormente, principalmente no pré-processamento dos dados.

2.3 *Deep Learning* para detecção de *fake news*

Atualmente, *deep learning* se configura como uma técnica efetiva e escalável do estado da arte na detecção de *fake news* (WU et al., 2019). Além de características linguísticas, existem modelos que utilizam características de propagação de uma notícia nas redes para encontrar padrões. O trabalho em Liu (2018), por exemplo, utilizou uma rede neural profunda para classificar o caminho de propagação de uma notícia, construído a partir de *tweets* e *retweets* para detectar *fake news*. Nas seções a seguir são especificados alguns aspectos conceituais dentro do contexto de *Deep Learning* para detecção de *fake news*, que embasaram a construção do projeto.

2.3.1 *Hierarchical Propagation Networks Representation*

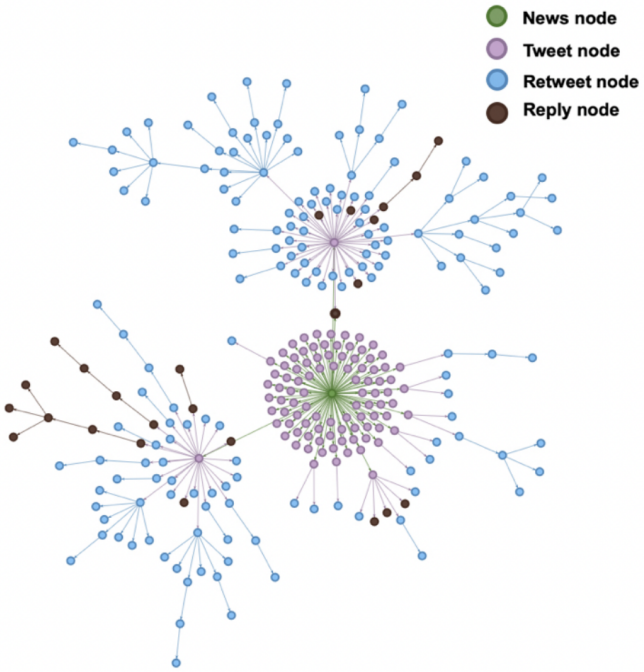
A representação de redes de propagação hierárquicas, ou *Hierarchical Propagation Networks Representation*, em inglês, baseia-se no fato de que a rede de propagação de informações em redes sociais possui uma estrutura hierárquica (SHU et al., 2020). A figura 3 esquematiza um exemplo de rede hierárquica a partir de uma cascata de nós que se inicia em uma notícia (nós verdes), e se propaga a partir de *tweets* (nós roxos), *retweets* (nós azuis) e *replies* (nós pretos). A partir dessa representação, é possível classificar a propagação em dois níveis distintos:

- Redes de nível Macro: consistem na propagação global da notícia e incluem seus nós (*news node*), dos *tweets* (*tweet node*) e dos *retweets* (*retweet node*). Aqui os nós representam os *tweets* e as bordas representam a relação de *retweets* entre eles.
- Redes de nível Micro: consistem no compartilhamento local e representam a árvore de conversação (*conversation tree*) a partir das respostas aos *tweets/retweets* (*reply nodes*) e das relações entre essas respostas (bordas). No Twitter, um usuário pode responder tanto o *tweet* quanto uma resposta desse *tweet*. No segundo caso, forma-se a chamada *thread*, que é representada pela cadeia de *reply nodes*.

Com essa separação feita e a estrutura hierárquica construída, os modelos conseguem extrair padrões tanto da da rede de nível macro, quanto da rede de nível micro de propagação das *fake news* na rede. As principais *features* podem ser divididas em três grupos de análises (SHU et al., 2020):

1. Análise estrutural: busca incluir nos modelos informações sobre a estrutura hierárquica proposta anteriormente em ambos os níveis. Algumas *features* são profundidade da

Figura 3 – Representação da propagação de informações em redes sociais



Fonte: (SHU et al., 2020)

árvore, número de nós, número de cascatas, número de robôs compartilhando, *tweet* com o maior engajamento.

2. Análise temporal: busca revelar a frequência e intensidade em que a informação é disseminada e que gera engajamento. Inclui *features* como a diferença média de tempo entre *retweets* adjacentes, diferença de tempo entre o primeiro *tweet* e os últimos *retweets*, diferença de tempo entre a postagem do Twitter e o primeiro *retweet*, diferença de tempo média entre respostas adjacentes.

3. Análise linguística: presente em sua maior parte na rede de nível micro, busca entender como os usuários estão expressando suas emoções, reações e opiniões sobre a notícia. Nesse sentido, os modelos utilizam-se de técnicas de NLP para gerar *scores* de sentimento, possuindo *features* linguísticas como média de *score* de sentimento de respostas, sentimento médio do primeiro nível da cascata mais profunda.

2.3.2 Detecção Explicável de *Fake News*

Um desafio no desenvolvimento de modelos de *deep learning* é a criação de soluções explicáveis/interpretáveis, isto é, que tragam o porquê um determinado *output* foi dado em uma predição (SHU et al., 2019). Isso ocorre pela complexidade inerente das várias

camadas e milhares de transformações lineares e não lineares que a arquitetura de uma rede profunda possui. As aplicações existentes são muitas vezes citadas como modelos de natureza caixa-preta (*black-box*), justamente pela dificuldade em entender quais aspectos dos dados de entrada nortearam a decisão da rede (XIE et al., 2020). No problema atacado por este projeto, essa preocupação é ainda mais relevante, por alguns motivos:

1. O projeto possui caráter educativo e visa dar insumos aos usuários para tomar uma decisão sobre uma notícia. É essencial que o modelo traga quais são os aspectos relevantes para a conclusão tomada.

2. No contexto de grande polarização em que o debate político em redes sociais se encontra, é muito importante que o projeto preserve sua credibilidade buscando as soluções mais transparentes possíveis.

3. As explicações derivadas podem trazer novos *insights* e conhecimento, antes oculto, tanto para usuários quanto para outros pesquisadores.

Nesse sentido, define-se aqui o critério de inerência, parte da taxonomia de interpretabilidade de modelos de aprendizado de máquina (MOLNAR, 2020). A taxonomia proposta nasce do fato de que a interpretabilidade dos métodos pode ser categorizada de acordo com critérios.

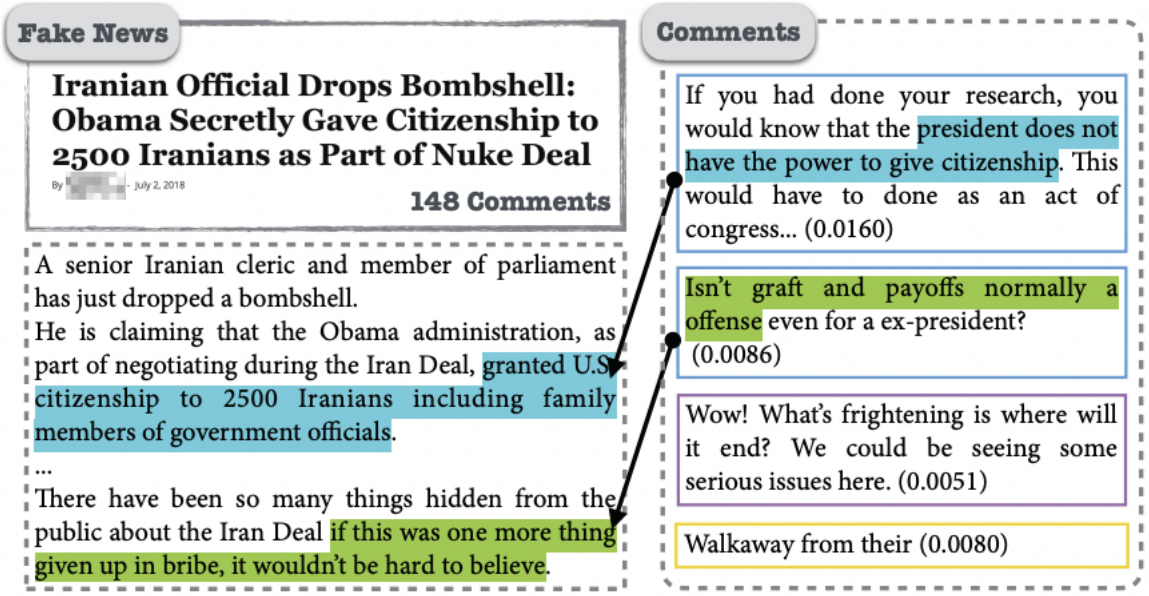
O critério de inerência, por exemplo, consiste em determinar se a interpretabilidade será alcançada por meio de restrições impostas à complexidade do modelo (intrínseco) ou aplicando métodos que analisam o modelo após o treino (*post-hoc*). Define-se assim:

- Explicabilidade intrínseca: modelos que incorporam a explicação nas suas estruturas, através de *features* que têm um grande papel em interpretar as predições. Pode ser designada por transparência, ou seja, "como o modelo funciona?".

- Explicabilidade *post-hoc*: requer um segundo modelo para prover explicações de um modelo existente. Procura responder à questão: "o que mais o modelo nos pode dizer?"

Alguns trabalhos dentro do contexto de *Deep Learning* para identificação de *fake news* incluem em sua arquitetura módulos de explicabilidade intrínseca. O *framework dEFEND* (*Explainable Fake News Detection*) (SHU et al., 2019), por exemplo, propõe utilizar um mecanismo de atenção cooperativa para capturar a explicabilidade de frases das notícias relacionando-as a comentários de usuários.

Figura 4 – Comentários explicáveis capturados pelo *dEFEND*



Fonte: (SHU et al., 2019)

A figura 4 esquematiza como o *dEFEND* relacionou alguns comentários da postagem feita no Twitter com trechos chaves da notícia, de forma a tornar a saída do modelo explicável.

2.4 Coleta de Dados

O termo coleta de dados refere-se ao processo de absorver dados de uma determinada fonte, a fim de armazená-los em uma outra infraestrutura (TIKITO et al., 2017). A motivação para realizar esse fluxo é centralizar dados de uma ou mais fontes distintas em um mesmo local, sob o pretexto de facilitar e uniformizar o acesso a eles.

A coleta pode ser dividida em 3 etapas: extração (*Extract*), transformação (*Transform*) e armazenamento (*Load*) (VASSILIADIS, 2009). A etapa de extração consiste em adquirir os dados crus da fonte de dados. Sobre os dados adquiridos podem ser efetuadas transformações, removendo informações desnecessárias e alterando o formato dos arquivos gerados. A etapa de armazenamento, por sua vez, consiste em carregar os arquivos gerados pela extração ou pela transformação em um sistema de dados.

Classifica-se a coleta por sua frequência de execução, em *streaming* ou em lotes (BEN-

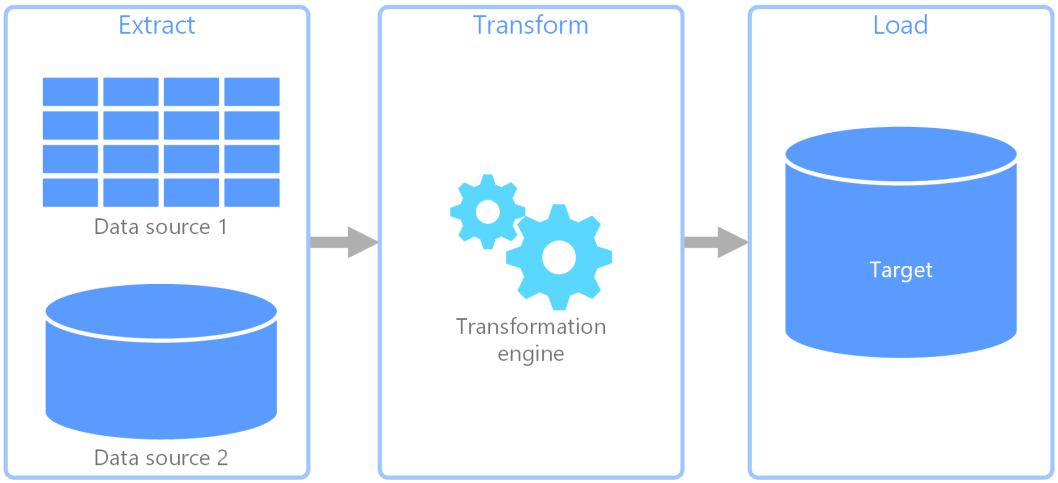
JELLOUN et al., 2020):

- Ingestão em *streaming*: os dados são adquiridos em tempo real, ou seja, a coleta é acionada por um evento indicando a existência de novos dados.
- Ingestão em lotes (*batch*): os dados são adquiridos em uma frequência predeterminada, ocorrendo a ingestão referente a todo o período entre duas execuções de uma vez.

Outra forma de classificação da ingestão é de acordo com a ordem de execução das etapas de extração, transformação e armazenamento. Dois tipos comuns são (MICROSOFT, 2022):

- *Extract Transform Load* (ETL): as coletas classificadas como ETL executam a transformação sobre os dados crus antes de armazená-los, como ilustrado na figura 5.

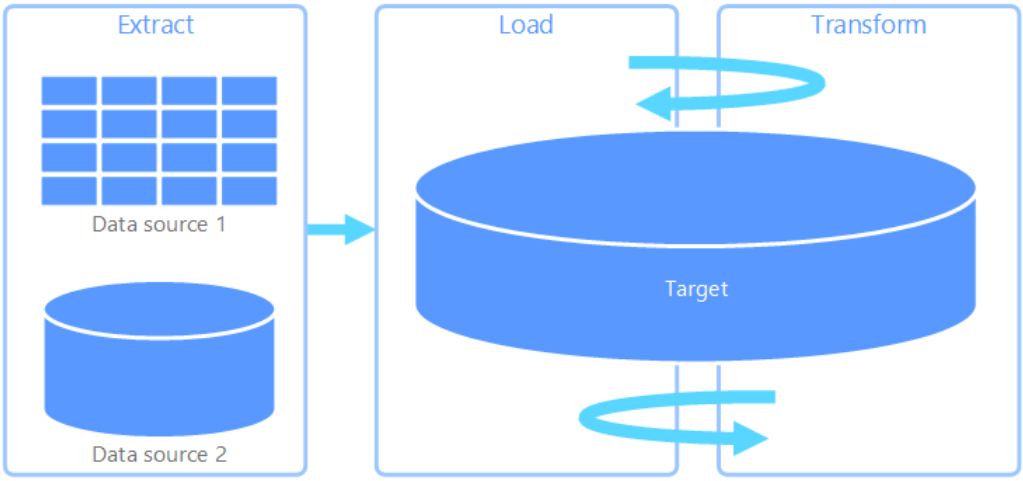
Figura 5 – Diagrama do fluxo de dados em uma ingestão ETL



Fonte: (MICROSOFT, 2022)

- *Extract Load Transform* (ELT): as coletas classificadas como ELT armazenam os dados crus para então efetuar transformações sobre eles. Assim, tanto dados crus quanto processados são armazenados e podem ser reutilizados, como ilustrado na figura 6.

Figura 6 – Diagrama do fluxo de dados em uma ingestão ELT



Fonte: (MICROSOFT, 2022)

3 ESPECIFICAÇÃO DE REQUISITOS

Para atingir os objetivos estabelecidos, o projeto cria um *bot* interativo para os usuários do Twitter. Nesse sentido, são apresentados aqui as histórias de usuário, os requisitos não-funcionais e os requisitos do modelo de aprendizado de máquina, utilizados para embasar decisões do projeto.

O sistema possui quatro histórias de usuário:

1 Como usuário do Twitter, quero marcar o *bot* em uma publicação de Twitter, contendo possivelmente uma *fake news*.

2 Como usuário do Twitter, quero ser avisado quando a resposta do *bot* for feita e ler as saídas do sistema.

3 Como pesquisador, quero ter acesso ao *dataset* das postagens, contendo os textos e metadados das postagens utilizadas no sistema do *bot*.

4 Como pesquisador, quero ter acesso aos *outputs* do modelo de aprendizado de máquina e ao código fonte do sistema para tirar *insights*.

A primeira e a segunda história do usuário têm início com um usuário qualquer do Twitter marcando a conta do projeto em um *tweet*; após isso, o *bot* é acionado, coleta os dados do *tweet* e os armazena. Em seguida, esses dados são analisados pelo modelo de aprendizado de máquina, que gera métricas para indicar possíveis indícios de que a notícia não é confiável. Essas métricas são transformadas em um texto informativo e esse texto é apresentado para o usuário que chamou o *bot*. Com esse resultado em mãos, o usuário terá mais insumos para decidir se confia ou não naquela postagem. Um exemplo fictício do funcionamento do sistema é mostrado na figura 7. Seguindo o padrão descrito, o *bot* (*@ProjetoLibra*) é marcado em uma postagem suspeita na rede social, respondendo automaticamente com os resultados da análise.

Figura 7 – Exemplo de funcionamento da primeira história de usuário

Já a terceira e quarta história do usuário funcionam da seguinte maneira: os dados armazenados (*dataset* anotado e *outputs* explicáveis dos modelos) pelo sistema, assim como o código fonte da ferramenta que possibilitou a sua geração, são disponibilizados em um repositório público e qualquer pesquisador ou usuário da internet que tiver interesse pode acessá-los.

3.1 Requisitos não funcionais

Tendo em vista as histórias de usuário, são definidos os requisitos não funcionais para se atingir os objetivos.

- Alta velocidade de atendimento: para que o *bot* seja utilizado de forma natural pelos usuários do Twitter e sem perda de interesse e contexto, é essencial que ele forneça

uma resposta rápida, com um tempo menor que 10 minutos após ser marcado em uma postagem.

- Alta taxa de resposta: para que o sistema atenda aos usuários é necessário que ele responda o maior número de marcações possíveis. Prevendo que o projeto pode enfrentar eventuais erros, estabelece-se como requisito uma taxa de resposta de 80%.

- Projeto sem custos: é necessário construir um sistema com custo zero para os participantes do projeto em todos os seus âmbitos (desenvolvimento, arquitetura, etc).

- Banco de dados escalável: a quantidade de dados armazenados no banco de dados tende a crescer bastante com o passar do tempo, por esse motivo ele deve ser escalável, isto é, se o projeto necessitar de mais espaço, deve ser possível alocar mais memória automaticamente.

3.2 Requisitos do modelo de aprendizado de máquina

Tendo em vista que o sistema é focado na interação com os usuários, três requisitos foram elencados durante o processo de escolha do modelo:

- Interpretabilidade: é necessário que o modelo gere *outputs* explicáveis do porquê chegou em uma determinada conclusão.

- Transparência: é necessário fornecer de forma transparente e clara como o modelo funciona, quais são seus *inputs* e *outputs* e quais ferramentas são utilizadas.

- Performance: é necessário que o modelo de classificação escolhido atinja uma boa performance. A métrica principal escolhida para medir esse requisito foi a *F1 Score*, média harmônica entre a precisão e a sensibilidade medida na fase de testes do modelo. O *F1 Score* mínimo considerado foi de 70%, valor mínimo reportado como aceitável pela literatura (SHU et al., 2020, 2019; LU et al., 2020).

É importante ressaltar que a complexidade do projeto se encontra principalmente nos requisitos não funcionais do sistema e no desenvolvimento e implementação dos algoritmos de aprendizado de máquina.

4 METODOLOGIA DO TRABALHO

Neste capítulo serão relatadas as diferentes fases do projeto e as metodologias de trabalho utilizadas em cada uma. Para isso, além dos aspectos metodológicos, são apresentados os diferentes módulos que compõem o sistema, o *roadmap* seguido e, por fim, a dinâmica e divisão de trabalho empregada na relação com o grupo de pesquisadores do projeto interdisciplinar IBRA USP.

4.1 Fase inicial de pesquisa

A primeira fase do projeto foi consultar especialistas e membros de ONGs para conhecer e discutir sobre problemas que poderiam ser atacados, a fim também de se obter uma maior imersão em diferentes temas. Em um mês foram prospectadas vinte ONGs por meio de *emails* e redes sociais institucionais de membros das organizações. Desses contatos, houve doze respostas, e nove reuniões foram marcadas. Esse processo, resumido na Tabela 1, foi essencial para se chegar a um alinhamento de expectativas interno e se obter um direcional para o projeto.

A discussão e a imersão proporcionadas por essa prospecção levou à decisão de focar no problema das *fake news* em ambientes digitais, levando em conta a preocupação unânime por parte das ONGs e a possibilidade de se utilizar tecnologias de aprendizado de máquina, arquitetura *serverless* e desenvolvimento de software. Além disso, atacando este problema, haveria um alinhamento de objetivos com os membros do projeto do IBRA USP, possibilitando troca de conhecimento e proximidade durante o desenvolvimento, como é descrito na Seção 4.3.

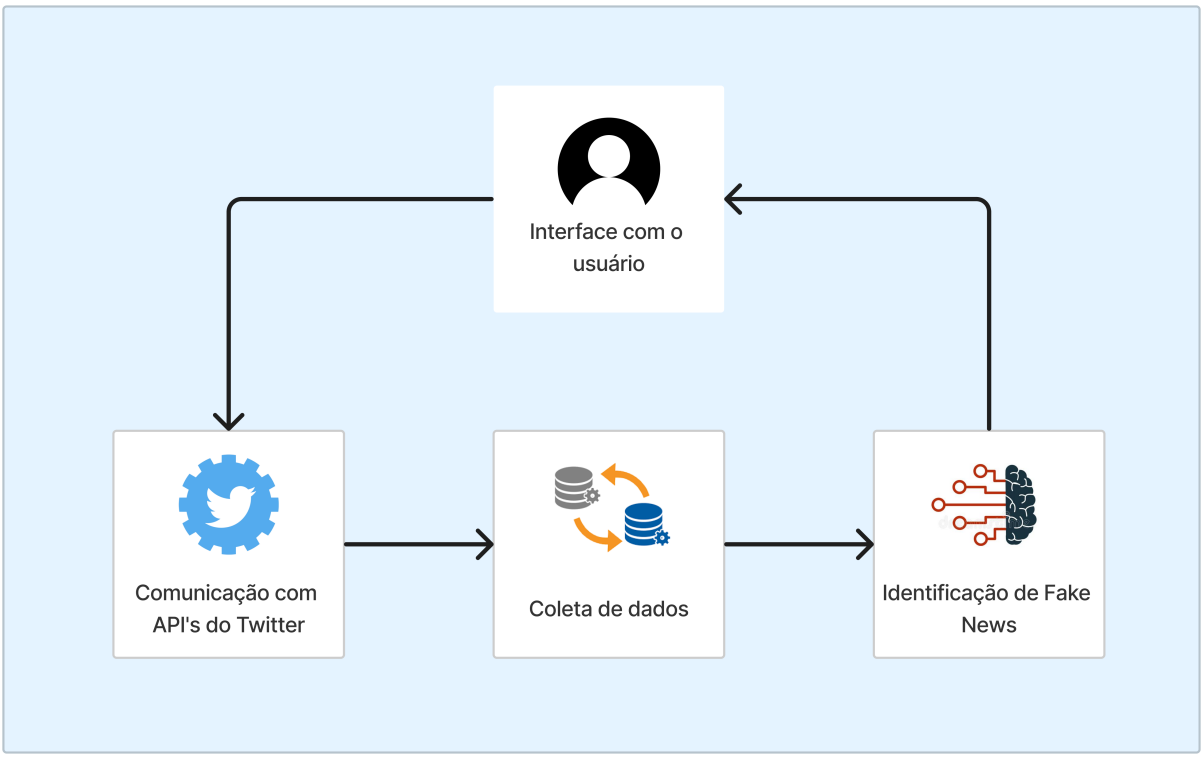
Tabela 1 – ONGs e projetos prospectados

ONG	Contatado	Resposta	*Call* feita
App Meu Deputado	✓	✓	✓
Eleito Por Mim	✓	✓	✓
Open Knowledge Brasil	✓	✓	✓
Instituto Liberdade Digital	✓	✓	✓
Monitor Nuclear	✓	✓	✓
Politize!	✓	✓	✓
Pacto pela democracia	✓	✓	✓
Projeto 7c0	✓	✓	✓
Transparência Eleitoral Brasil	✓	✓	✓
Aos Fatos	✓	✓	×
Voto Consciente	✓	✓	×
Revista AzMina	✓	✓	×
Política Por Inteiro	✓	×	×
Monitor do debate político Meio digital da USP	✓	×	×
Eleições Sem Fake	✓	×	×
Rede internacional de checagem de fatos	✓	×	×
Agência Lupa	✓	×	×
Professor Nazareno Andrade	✓	×	×
Transparência Brasil	✓	×	×
M0naBot	✓	×	×
Total	20	12	9

4.2 Fase de Desenvolvimento

Após a definição do escopo e da parceria com o IBRA USP, foi elaborada uma metodologia de trabalho baseada em entregas que incrementaram progressivamente a adição de valor e a complexidade do sistema. Para isso, foram estabelecidos diferentes módulos de desenvolvimento e uma arquitetura básica para o sistema.

Figura 8 – Esquematização dos Módulos do Sistema



Conforme a Figura 8, foram definidos 4 módulos principais, de acordo com a finalidade de seus componentes. A interação entre eles é representada pelas setas, que indicam sua sequência de acionamento.

- Módulo de interface com o usuário: representa o início e o fim do fluxo do sistema. Trata-se da entidade no Twitter responsável pela comunicação com o público, tanto ao receber a marcação do usuário quanto ao responde-lá. Aciona o módulo de comunicação com APIs do Twitter.

- Módulo comunicação com APIs do Twitter: responsável pela comunicação do sistema com o Twitter, com o uso dos diferentes *endpoints* disponibilizados na API para detecção de marcação do perfil do *bot* e coleta dos dados necessários. Aciona o módulo de coleta de dados.

- Módulo de coleta de dados: corresponde aos componentes responsáveis pela coleta, processamento e armazenamento dos dados não-estruturados provenientes da co-

municação do Twitter, de forma a alimentar o modelo de aprendizado de máquina. Aciona o módulo de identificação de *fake news*.

- Módulo de identificação de *fake news*: diz respeito ao desenvolvimento e *deploy* do modelo de aprendizado de máquina em nuvem e à obtenção de *insights* dos dados coletados. Aciona o módulo de interface com o usuário, enviando a resposta com informações relevantes sobre a postagem de *input*.

O desenvolvimento foi realizado com uma progressão vertical, isto é, desenvolvendo e implementando paralelamente as partes, ao invés de sequencialmente. Dessa forma, foi possível investir esforços de trabalho na conexão dos diferentes módulos e na orquestração do sistema em um estágio razoavelmente inicial do projeto. Essa estratégia se mostrou importante, pois desenvolver a comunicação entre os módulos e implementar o sistema de ponta a ponta seguindo os requisitos foram alguns dos maiores desafios do desenvolvimento.

Além disso, essa metodologia evitou uma discrepância na robustez dos módulos ao final do projeto, reduzindo o risco de, por exemplo, o módulo de aprendizado de máquina estar muito desenvolvido, porém sem utilidade para o usuário pois o módulo comunicação com o Twitter não está funcional.

Foram elencadas as diferentes fases e o *roadmap* de entregas:

- **Fase 1 (maio - abril)**
  - Pesquisa de tema e entrevistas com ONGs
  - Definição do problema a ser atacado
- **Fase 2 (junho - agosto)**
  - Definição do modelo de *machine learning* e início do desenvolvimento
  - Definição da arquitetura e das ferramentas a serem utilizadas
  - Início do desenvolvimento da comunicação com o Twitter
- **Fase 3 (agosto - outubro)**
  - Comunicação com API do Twitter desenvolvida localmente
  - Modelo treinado e testado localmente
  - Coleta de dados desenvolvida localmente

- **Fase 4 (outubro - novembro)**
  - *Deploy* dos diferentes módulos na arquitetura
  - Desenvolvimento da orquestração dos fluxos e conexão dos módulos
- **Fase 5 (novembro - dezembro)**
  - Divulgação e realização de testes e avaliações
  - Ajustes e melhorias de acordo com problemas elencados durante as etapas anteriores

O cronograma da tabela 2 representa o planejamento das entregas descritas acima. O MVP (*Minimum Viable Product*), previsto para novembro, corresponde ao sistema com início de execução a partir da marcação do perfil no Twitter, coleta e armazenamento de dados a respeito do *tweet* de *input*, execução do modelo de aprendizado de máquina e resposta com *insights* gerados ao fim funcionais. A ideia é construir uma versão funcional ponta a ponta do sistema para iniciar os testes.

Tabela 2 – Cronograma

Fase	Maio	Junho	Julho	Agosto	Setembro	Outubro	Novembro	Dezembro
Fase 1								
Fase 2								
Fase 3								
Fase 4							MVP	
Fase 5								

4.3 Projeto IBRA USP

O projeto foi desenvolvido com apoio do projeto IBRA USP, de forma a integrar os conhecimentos interdisciplinares dos integrantes dos dois grupos. Nesta seção é feita uma breve apresentação do grupo de pesquisa e qual foi o seu escopo de atuação dentro deste trabalho.

O IBRA (*Internet Balancing Reasoned Algorithm*) é um projeto multidisciplinar de pesquisa da USP sob coordenação da Professora Doutora Roseli de Deus Lopes, professora

do Departamento de Engenharia de Sistemas Eletrônicos da Escola Politécnica da Universidade de São Paulo, que tem como objetivo gerar conhecimento e contribuições científicas no campo de moderação de conteúdo em redes sociais. O grupo do projeto é composto por 15 integrantes entre professores, doutorandos, mestrandos e estudantes bolsistas das áreas de direito, matemática e engenharia e foca no estudo e desenvolvimento de modelos de aprendizado de máquina que atuem no combate à desinformação e ao discurso de ódio em seus diversos espectros. O projeto é dividido em 5 frentes:

- Frente de Recursos: responsável por desenvolver ferramentas de coleta e anotação de dados automática e implementar conceitos de *adversarial robustness* em NLP (MORRIS et al., 2020).
- Frente de Imagens: responsável por desenvolver modelos de detecção de manipulação de imagens dentro do contexto de propagação de *fake news*.
- Frente de Áudio: responsável por desenvolver modelos de detecção de manipulação de áudio dentro do contexto de propagação de *fake news*.
- Frente de Propagação: responsável por desenvolver modelos que estudem a propagação de *fake news* no contexto de redes sociais, utilizando também *features* linguísticas.
- Frente modelo brasileiro para *tweets*: responsável por desenvolver um modelo de linguagem de propósito geral para *tweets* no contexto do Brasil.

Os integrantes do IBRA, mais especificamente da Frente de Propagação, auxiliaram no desenvolvimento, treinamento e teste dos modelos de aprendizado de máquina localmente, dentro do módulo de identificação de *fake news*, descrito na Seção 4.2. Essa parte do projeto foi realizada a partir da colaboração entre ambas as partes. Além disso, foram realizadas reuniões semanais com o grupo para mentoria e discussão das priorizações e avanços. O grupo de TCC foi responsável por realizar o *deploy* do modelo no sistema e pelos demais módulos.

5 MODELO DE APRENDIZADO DE MÁQUINA

Esse capítulo visa explicar a escolha do *framework dEFEND* como modelo de aprendizado de máquina e o seu desenvolvimento e implementação no projeto em conjunto com o Projeto IBRA USP. Nesse sentido, os aspectos levantados se encontram dentro do contexto de desenvolvimento de modelo do módulo de Identificação de *Fake News*, apresentado na Seção 4.2.

5.1 *dEFEND: Explainable Fake News Detection*

A escolha do modelo se deu dentro do contexto de pesquisa do Projeto IBRA USP. Os integrantes da Frente de Propagação já haviam iniciado o processo de desenvolvimento de modelos supervisionados de classificação binária utilizando dois *frameworks*: o *dEFEND* (SHU et al., 2019), baseado em *features* linguísticas do texto da notícia e dos comentários feitos na rede social do Twitter, e o *HPNF* (SHU et al., 2020), sigla para *Hierarchical Propagation Networks Representation*, com foco em analisar os grafos de propagação das postagens nas redes sociais.

A ideia inicial era de implementar os dois modelos no sistema de forma complementar, visto que são baseados em propostas diferentes, mas com resultados de performance condizentes com os requisitos do projeto, como os apresentado pelos autores. Além disso, inicialmente planejava-se fazer o treinamento dos modelos utilizando *datasets* brasileiros, colhidos pelos integrantes da Frente de Propagação do IBRA.

Porém, foram considerados os riscos de projeto de não haver tempo hábil de colher os *datasets* brasileiros e utilizá-los no desenvolvimento de ambos os modelos, tendo em vista a necessidade de desenvolvimento dos outros módulos do projeto. Nesse sentido, a decisão tomada foi de desenvolver inicialmente um modelo baseado no *framework dEFEND* e treinado com o *dataset* proposto pelos autores do artigo, em inglês, de forma a obter mais rapidamente um modelo pronto e não atrasar o desenvolvimento da conexão entre os módulos e o *deploy* do sistema de ponta a ponta. Como este modelo é parci-

almente independente da língua (*language-agnostic*), ele poderia ser adaptado do inglês para português brasileiro.

Além disso, o *dEFEND* se propõe a não só prever se uma notícia é falsa ou não, mas também a explicar seu veredito, de forma a atingir o requisito de interpretabilidade do projeto. Assim, planejava-se primeiro chegar em um sistema funcional, para depois incrementá-lo com a análise e saída do *HPNF* e com o *dEFEND* treinado em português.

Para explicar o melhor o funcionamento do *dEFEND*, são resumidos os quatro componentes principais do modelo, retirados de seu artigo original:

- Codificador do conteúdo da notícia: inclui um codificador de palavras e um de frases e aprende a representação das palavras utilizando mecanismos de RNNs e redes neurais hierárquicas.
- Codificador de comentários do usuário: codifica os comentários feitos utilizando mecanismos de redes neurais recorrentes.
- Atenção cooperativa: cria uma rede de atenção cooperativa de sentença/comentário, que tem como objetivo capturar a explicabilidade de frases das notícias e comentários de redes sociais para a explicar o resultado do *output* de forma intrínseca.
- Detector de *fake news*: faz a predição sobre a notícia utilizando uma rede neural treinada com o algoritmo RMSprop (RUDER, 2016).

O código de funcionamento do *framework* foi retirado do *GitHub* original dos autores do artigo e os códigos de treinamento e teste do modelo, descritos na Seção 5.2, foram iniciados pelos integrantes do IBRA e terminados pelos integrantes deste projeto.

5.2 Desenvolvimento do Modelo

O treinamento e teste do modelo foi realizado utilizando-se uma base robusta e já anotada, criada pelos pesquisadores do *FakeNewsNet* (SHU et al., 2018), um sistema utilizado em estudos relevantes de detecção de *fake news* (SINGHAL et al., 2020; SHU et al., 2020). A coleta, descrita no artigo, é realizada a partir de notícias de duas plataformas com checagem de fatos: GossipCop[1] e PolitiFact[2]. O *dataset* não é disponibilizado pelos

[1]https://www.suggest.com/
[2]https://www.politifact.com/

autores e a coleta deve ser realizada a partir do código[3] público utilizando APIs do Twitter e *crawlers*. Porém, foram necessárias alterações razoáveis, visto que sua última versão foi disponibilizada em setembro de 2021 e alguns problemas de dependências de bibliotecas foram encontrados.

Os dados incluem informações das notícias (por exemplo, corpo do texto), links na internet e os respectivos rótulos de verdadeira ou falsa. Já os dados das postagens em rede social (*tweets*, *retweets* e comentários) possuem, além dos textos, alguns metadados relacionados ao engajamentos com as postagens, como informações dos usuários e hora da interação.

Foram coletadas cerca de 21 mil notícias falsas e suas respectivas postagens e comentários. A proporção de notícias falsas na base é de aproximadamente 25%:

Tabela 3 – Amostra coletada para treino por fonte

Fonte	Notícias Falsas	Notícias Verdadeiras	Total
PolitiFact	404	558	962
GossipCop	4,876	15,212	20,088
Total	5,280	15,770	21,050

A partir da base completa e seguindo suas proporção de verdadeiro/falso, separou-se um conjunto de treino e um conjunto de teste com uma proporção de 75% e 25% da amostra, respectivamente. Dessa forma, o modelo foi treinado apenas utilizando os dados do primeiro conjunto e foi testado fazendo predições nos dados do segundo conjunto. As métricas de performance (acurácia, sensibilidade e F1 Score) foram calculadas e são mostrados na Tabela 4.

Tabela 4 – Resultados do Treino dEFEND

Método	Acurácia	Sensibilidade	F1 Score
dEFEND	0.85	0.75	0.81

Considerou-se que os resultados obtidos foram condizentes com os requisitos definidos e que o modelo poderia ser utilizado dentro da arquitetura do projeto. Vale ressaltar que o treinamento foi realizado localmente e um dos maiores desafios do projeto foi produtizar o modelo dentro do sistema. O *dataset* brasileiro não foi coletado a tempo, fazendo com

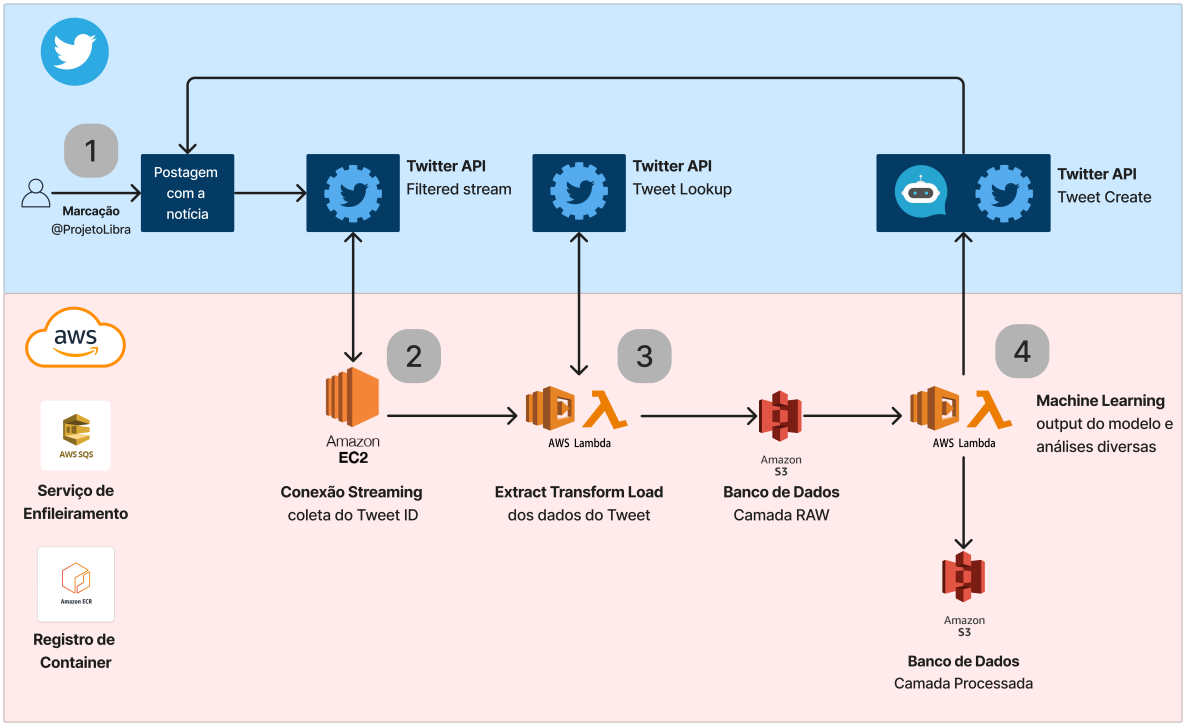
[3] https://github.com/KaiDMML/FakeNewsNet

que o treinamento e implementação do *HPNF* e do *dEFEND* brasileiro fossem planejados em trabalhos futuros.

6 IMPLEMENTAÇÃO

Neste capítulo são descritos os aspectos da arquitetura do projeto, com uma visão geral das etapas de funcionamento, uma explicação sobre a as tecnologias utilizadas e as justificativa de escolha e, por fim, um detalhamento mais aprofundado do funcionamento do sistema. Foi feito um fluxograma indicando os principais agentes do sistema e o relacionamento entre eles (figura 9).

Figura 9 – Fluxograma de Funcionamento de Projeto



Na esquematização há dois grandes blocos de contexto. O bloco superior, em azul-claro, representa a parte do processo que é feita via interface e funcionalidades que o próprio Twitter fornece, como as APIs, o perfil do *bot*, as postagens e os comentários. O segundo bloco, em laranja, engloba as diferentes instâncias do sistema, que foram desenvolvidas e implementadas utilizando ferramentas de computação, armazenamento e

redes da *Amazon Web Services* (AWS). A maioria das etapas e conexões descritas a seguir foram implementadas na linguagem *Python*, com a utilização de bibliotecas específicas. A comunicação entre as ferramentas é representada pelas setas, que indicam fluxos de dados e de mensagens de acionamento.

O processo é composto pelas seguintes etapas, enumeradas na figura 9, e descritas de forma mais profunda na Seção 6.2:

1. Marcação do perfil do *bot* em uma postagem no Twitter, realizada pelo usuário.
2. Detecção em tempo real da marcação do perfil do *bot*, etapa que engatilha uma nova instância de execução do fluxo.
3. ETL dos dados do *tweet*, que coleta, transforma e armazena os dados utilizados pelo modelo de aprendizado de máquina.
4. Execução do modelo de *machine learning*, processamento dos dados para criação da resposta à marcação e armazenamento dos *outputs* do sistema.

6.1 Tecnologias Utilizadas

Tendo em vista as várias funções, instâncias e relacionamentos do sistema, a escolha das ferramentas e a sua implementação representam desafios centrais do projeto. Esta seção, além de detalhar as tecnologias utilizadas, visa justificar a estruturação da arquitetura da forma que foi apresentada e também explicitar alguns desafios enfrentados.

6.1.1 *Amazon Web Services* (AWS)

AWS é uma plataforma de computação em nuvem oferecida pela Amazon, que dispõe serviços com facilidade de configuração e alta disponibilidade, como computação *serverless*, máquinas virtuais escaláveis, além de opções de hospedagem para as demais tecnologias que são utilizadas no projeto. Além de possuir ferramentas que atendem aos requisitos, principalmente não funcionais do sistema (Seção 3.1), a plataforma foi escolhida pela facilidade em se encontrar documentações, materiais de apoio e fóruns onde a implementação de ferramentas são discutidas.

Vale ressaltar que, sendo um dos requisitos o de construir um sistema sem custo, foram escolhidas somente ferramentas que possuem um *free tier* estabelecido, isto é, um

nível gratuito mensal[1].

6.1.1.1 Amazon EC2

A *Amazon Elastic Compute Cloud* (EC2) é um serviço que permite o aluguel de computadores virtuais. A EC2 é ideal para aplicações que necessitam de alta disponibilidade, por ser possível a execução do código de maneira ininterrupta, salvo possíveis erros e falhas do sistema. Além disso, a ferramenta possui um nível gratuito que permite 750 horas por mês de execução da instância mais básica, *t2.micro*, suficiente para códigos simples[2].

Nesse sentido, a EC2 foi escolhida para hospedagem do ponto inicial do sistema - etapa 2 de Detecção em tempo real da marcação - por atender aos requisitos de alta velocidade de atendimento e alta taxa de resposta do sistema, na medida em que permite que códigos mais simples, como é o caso do *script* dessa etapa, sejam executados de forma contínua durante todo o mês.

6.1.1.2 Amazon SQS

O *Amazon Simple Queue Service* (SQS) é um serviço de mensageria que permite enviar, armazenar e receber mensagens em filas, facilitando a comunicação entre diferentes componentes de um sistema de computação. É possível enviar e receber mensagens programaticamente com facilidade, o que essencialmente permite a comunicação entre quaisquer sistemas ou componentes de computação de alto nível.

O *Amazon SQS* foi utilizado nas comunicações entre as ferramentas da *AWS*, nas etapas 2 (Detecção da marcação), 3 (ETL) e 4 (execução do modelo de aprendizado de máquina) representadas na figura 9.

6.1.1.3 Amazon Lambda

A *Amazon Lambda* é um serviço de computação *serverless* e orientado a eventos que permite executar códigos para variados tipos de aplicações sem a necessidade, por parte dos desenvolvedores, de provisionar ou gerenciar servidores. As máquinas utilizadas são alocadas e iniciadas sob demanda, apenas para o tempo de execução necessário. O serviço possui compatibilidade com programas escritos diversas linguagens, incluindo *Python*, com suporte nativo a gatilhos com componentes em diferentes serviços, como através de

[1] https://aws.amazon.com/pt/free/?all-free-tier
[2] https://aws.amazon.com/pt/ec2/instance-types/

mensagens do Amazon SQS. A ferramenta ainda possui um nível gratuito mensal que possibilita 1 milhão de solicitações (chamadas de instância) gratuitas por mês e até 3,2 milhões de segundos de tempo de computação.

A escolha da *Amazon Lambda* atende ao requisito de alta velocidade de atendimento, na medida em que permite que várias instâncias diferentes da mesma função sejam acionadas de forma paralela, sem que um fluxo interfira na performance do outro. Nesse sentido, é adequada para as etapas 3 (ETL) e 4 (execução do modelo de aprendizado de máquina) e mais apropriada que a EC2 por não haver a necessidade de execução contínua das máquinas, o que pode gerar custo adicional sem necessidade.

6.1.1.4 Amazon ECR

O *Amazon Elastic Container Registry* (ECR) é um serviço de registro de imagens de contêineres, de forma que seja possível implantá-las de forma confiável e fácil em qualquer lugar. Uma contêiner é um pacote de software executável que contém todo o necessário para executar uma aplicação, como configurações, bibliotecas e ferramentas, de forma que o código seja executado de forma isolada e padronizada em qualquer ambiente.

Foi escolhido para suporte das funções Lambda, que são criadas a partir de imagens *Docker*, um software de hospedagem de contêineres.

6.1.1.5 Amazon S3

O *Amazon Simple Storage Service* (Amazon S3) é um serviço de armazenamento de alta escalabilidade e disponibilidade, com facilidade e velocidade de acesso de escrita e leitura programaticamente. Os objeto são armazenados em *buckets*, que possuem um sistema de arquivos próprio e podem ser acessados até 20000 vezes por mês sem custo.

O Amazon S3 foi escolhido como solução de armazenamento, presente nas etapas 3 (ETL) e 4 (execução do modelo de aprendizado de máquina), pois além de atender ao requisito de escalabilidade do banco de dados do sistema, possui flexibilidade. Pode ser utilizado como banco de dados não relacional e, dessa forma, otimizado de acordo com as necessidades especificas da aplicação e dos dados, em especial no formato de armazenamento dos dados. No caso deste trabalho, os arquivos são organizados de acordo com seus identificadores únicos e em formato JSON, de forma que sejam facilmente acessados pelo modelo de aprendizado de máquina e disponibilizáveis como *dataset*.

6.1.2 API do Twitter

A aplicação faz uso de diversos recursos disponibilizados pelo Twitter para interagir com a plataforma, permitindo de forma automática ações como publicação e leitura de *tweets* e busca por dados e metadados de publicações. A interação com esses recursos é feita por meio de uma API REST, cujo acesso é feito utilizando com chaves de acesso concedidas a projetos acadêmicos pela empresa. As chaves são solicitadas através de um formulário descrevendo os objetivos e o planejamento do projeto[3]. A comunicação com a API é feita a partir de códigos desenvolvidos em *Python* e implementadas nas ferramentas descritas na Seção 6.1.1.

Os *endpoints* utilizados estão elencados a seguir.

- *Filtered Stream*: os *endpoints Filtered Stream* permitem a recepção em tempo real de eventos no Twitter, através de uma conexão de *streaming*. É utilizado como ponto de entrada do sistema, criando uma notificação quando ocorre uma marcação do perfil do robô na rede social.
- *Tweets Lookup*: os *endpoints Tweets Lookup* permitem a busca por informações de qualquer postagem no Twitter, os chamados *tweets*, utilizando o método GET. São utilizados na coleta de dados da postagem original em que o *bot* é marcado.
- *Retweets Lookup*: permite a coleta de dados de *retweets* de uma postagem, ou seja, compartilhamentos simples, utilizando o método GET.
- *Quote Tweets Lookup*: permite a coleta de dados de *quote tweets* de uma postagem, ou seja, compartilhamentos com comentários, utilizando o método GET.
- *Create Tweet*: permite a criação de postagens programaticamente, utilizando o método POST. É o ponto final de um fluxo do sistema, enviando a resposta gerada à rede social.

6.2 Arquitetura

Nesta seção descreve-se o fluxo detalhado de execução das etapas do sistema, representado na figura 9, utilizando-se das especificações técnicas e justificativas apresentadas na Seção 6.1.

[3]https://developer.twitter.com/

1. O fluxo começa quando o o usuário comenta o *tweet* com a notícia, marcando a conta Twitter do projeto. Essa etapa é o gatilho para uma nova instância de fluxo do sistema ser iniciada.

2. O computador virtual (*Amazon EC2*) executando continuamente o *script* que implementa o *endpoint Filtered Stream* da API do Twitter recebe em tempo real o objeto que representa o evento de marcação. Esse objeto possui as chaves únicas identificadoras - denominadas *TweetIDs* - da postagem original e da marcação. Após recebê-las, a *EC2* envia uma mensagem para uma fila do *Amazon SQS*, com os parâmetros do objeto.

3. Uma instância da *Amazon Lambda*, unidade de serviço de computação *serverless* que contém o programa responsável pela coleta de dados, é acionada no envio da mensagem anterior para a fila. Optou-se por uma arquitetura ETL, uma vez que, analisados os *endpoints* da API do Twitter e seus respectivos objetos retornados, concluiu-se que os dados não teriam grande utilidade em seu estado cru, ou seja, sem nenhum processamento ou cruzamento com dados de outros *endpoints*. Os passos do processo de extração, processamento e armazenamento dos dados do *tweet* são detalhados a seguir:

    - Extração dos dados: utilizando o *TweetID* recebido do passo anterior, a *Lambda* realiza requisições aos *endpoints Tweets Lookup*, *Retweets Lookup* e *Quote Tweets Lookup*, de forma a coletar os textos e relacionamentos de todos os tipos de interações com a postagem e também seus metadados, como data, hora, geolocalização, informações dos usuários, número de respostas/curtidas entre outros. Além disso, é feita uma requisição para o *endpoint Search Tweet* para receber informações de *tweets* com exatamente o mesmo texto que a publicação/notícia original, mas que não estão relacionadas a ela diretamente.
    - Processamento: os dados crus, recebidos nas requisições, são transformados em JSON, seguindo uma estrutura que permite uma fácil leitura em etapas futuras.
    - Armazenamento: após o processamento inicial, esses dados são armazenados em um *bucket* de dados disponibilizado pelo serviço *Amazon Simple Storage Service* (S3). Esse primeiro *bucket* representa a camada *raw* do armazenamento de dados do sistema, isto é, os dados sofrem apenas pequenos processamentos antes de serem armazenados e possuem muitas outras informações que não são utilizados no sistema descrito neste trabalho. Essa estrutura permite que fu-

turamente diferentes projetos possam ser desenvolvidos de forma incremental utilizando-se desse mesmo fluxo, como a adição de outros modelos de aprendizado de máquina com diferentes *inputs*. A comunicação entre a *Lambda* e o S3 é feita pelo próprio código, com a biblioteca de *Python boto3* da AWS.

4. Após o armazenamento ser finalizado, a função Lambda da coleta de dados envia uma mensagem contendo as mesmas duas chaves identificadoras (*TweetIDs*) que iniciaram o fluxo para uma segunda fila do Amazon SQS. Essa mensagem aciona uma outra instância de Lambda, responsável por toda a parte final do fluxo:

   - O primeiro passo é buscar os dados do *tweet* no *bucket raw* do S3, utilizando também a biblioteca *boto3* e as chaves da mensagem recebida.
   - Tendo todos os dados, o *script* faz uma série de processamentos de forma a transformá-los em *inputs* válidos para o modelo *dEFEND* (descrito no Capítulo 5) e também calcular os *insights* que irão para a resposta.
   - A própria função Lambda executa o *dEFEND* a partir dos dados, faz a predição e compila todos os *outputs* do processamento e as conclusões. A partir disso, a resposta à marcação é gerada e postada por meio do *endpoint Create Tweet*.
   - Por fim, todos os *inputs* e *outputs* processo completo são armazenados em outro *bucket*, que representa a camada processada do banco de dados do projeto. Essas informações são disponibilizadas em um repositório público, correspondentes ao *dataset* anotado.

6.3 Resultados alcançados

A implementação da arquitetura descrita neste capítulo se consolidou em um sistema acessível pelo Twitter, através de uma marcação do perfil do projeto na rede social, conforme exemplificado na figura 10. Após alguns minutos, é feita uma resposta automática com informações relevantes encontradas. Além do código fonte de todas as partes do sistema, os dados resultantes de sua execução são disponibilizados em um repositório público no GitHub[4]. Os arquivos do *dataset* se encontram em formato JSON e contém os identificadores únicos dos *tweets* analisados, suas respostas, *retweets*, *quote tweets*, os *tweets* com o mesmo texto encontrados e o resultado da execução do modelo *dEFEND*.

[4]https://github.com/Projeto-LIBRA

Figura 10 – Exemplo de marcação e resposta do perfil do projeto

7 TESTES E AVALIAÇÃO

O projeto, já em sua etapa de validação com o público, sofreu um contratempo ao ter suas contas do Twitter bloqueadas e as chaves de *academic research*, que concediam acesso às APIs, revogadas permanentemente. Tendo esse imprevisto como contexto, são apresentados neste capítulo os planos de teste que foram estruturados para avaliar o sucesso do sistema como um todo e seus respectivos resultados. Nota-se que os testes para medir a performance do modelo implementado *dEFEND* já foram descritos na Seção 5.2.

No dia 30 de novembro de 2022, a conta no Twitter do projeto, à qual as chaves de acesso às APIs estavam associadas, foi permanentemente suspensa, impossibilitando que os *endpoints* utilizados no sistema fossem acessados.

Figura 11 – *Email* recebido do Twitter

A equipe de moderação do Twitter não enviou maiores detalhes sobre as razões do bloqueio, mesmo após envio de recurso e várias tentativas de contato. Apesar do infortúnio, foi possível realizar testes do sistema de ponta a ponta e até implementar melhorias e correções de *bugs*. O projeto entrou em produção no dia 26 de novembro e foi acionado 36 vezes antes de ser suspenso.

É interessante destacar que o Twitter passou por diversas mudanças devido a uma mudança de liderança conturbada no mesmo período de conclusão deste trabalho, o que trouxe incertezas e pode ter contribuído para a revogação de chaves de acesso à API, de-

vido a eventuais alterações nas políticas da empresa e nas equipes responsáveis (DUFFY, 2022).

O contratempo também causou modificações nos planejamentos de trabalhos futuros deste grupo e do projeto IBRA.

7.1 Questionário

A primeira pesquisa de usuários definida foi um questionário de *feedback*, que pode ser visualizado nas figuras 15 e 16 do Apêndice A, enviado junto da resposta dada pelo *bot*. A pesquisa tinha 5 objetivos principais, relacionados a entender a percepção dos usuários ao utilizar o sistema:

- Entender se o usuário concordou ou não com o veredito do sistema.
- Entender se o usuário achou o sistema útil para tomar uma decisão sobre a confiabilidade do *tweet*/notícia, de forma a validar o objetivo principal do sistema.
- Entender se o tempo de resposta estava adequada na percepção do público, de forma a validar o requisito não-funcional de velocidade de resposta.
- Entender se o usuário considera que sistemas interativos de identificação de notícias falsas que podem ser úteis para o combate a esse problema em redes sociais.
- Compilar sugestões e demais *insights* que poderiam surgir.

O propósito de utilizar um questionário está relacionado com o fato de que esse tipo de pesquisa possibilita a coleta de dados quantitativos e também qualitativos. O formulário foi estruturado com perguntas predominantemente fechadas e de rápido preenchimento, mas também abria a possibilidade, não obrigatória, de escrever livremente qualquer *feedback* que o usuário desejasse.

Infelizmente, com o contratempo da perda de acesso à API, o questionário obteve apenas 3 respostas, que não é um número razoável de para inferir qualquer conclusão. Nesse sentido, a avaliação do projeto se concentrou na mensuração de métricas sistêmicas.

7.2 Avaliação sistêmica

Os testes explicados nesta seção dizem respeito aos requisitos não-funcionais do sistema, descritos na Seção 3.1. As análises apresentadas a seguir foram feitas a partir de

dados de *inputs* e *outputs* do sistema e *logs* de execução dos módulos de computação da AWS. Vale ressaltar que o processamento dessas informações também foi feito a partir de um fluxo automático construído pelo projeto, utilizando as ferramentas AWS CloudWatchLogs e AWS Lambda para compilar os resultados armazenados na camada processada do banco de dados e disponibiliza-los para análise mais profunda.

A primeira análise realizada diz respeito ao número de vezes que a marcação da conta do *bot* foi realizada e o percentual de respostas dadas. Essas métricas tem como objetivo medir a taxa de resposta do sistema, definida como requisito nao funcional, de forma a entender se havia usuários que não estavam sendo atendidos.

Figura 12 – Gráfico: Evolução do número de chamadas por classificação

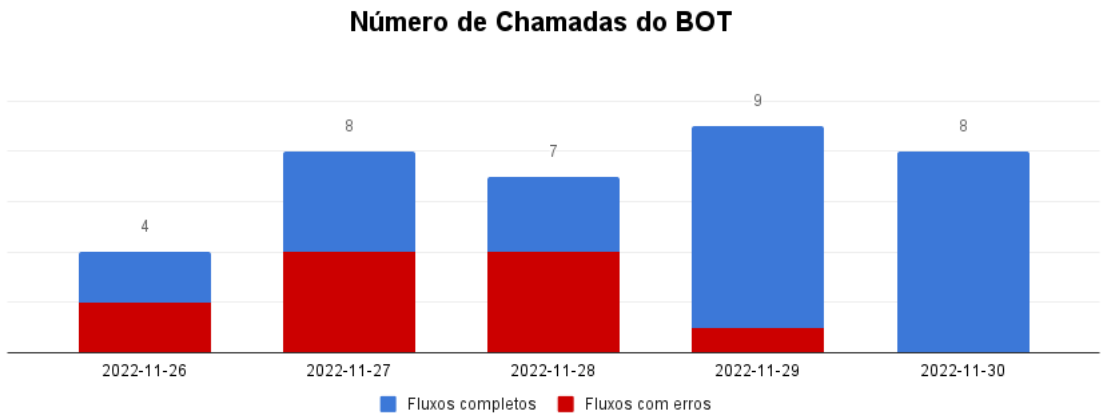


Figura 13 – Gráfico: Taxa de resposta ao usuário

A partir dos resultados gráfico da figura 12 e 13 é possível observar que o volume de marcações que não obtiveram respostas se concentra no começo do período e diminui até o dia 30 de novembro, em que não houveram fluxos com erros. Esses erros foram colhidos e analisados nos *logs* das máquinas e se deram por três motivos principais:

- Alguns *tweets* possuem milhares de comentários e *Retweets* associados. Para alguns casos o tempo limite de execução das funções Lambda não eram suficientes. Foi necessário aumentar o limite padrão para contornar este tipo de falha.
- A API do Twitter bloqueia a coleta de dados de usuários que não possuem perfis públicos e isso gerava um erro no código. Para este erro, foi necessário incluir um fluxo de exceção no código da ETL.
- O *endpoint* de busca de Twitter, utilizado para encontrar outras postagens com o mesmo texto do *tweet* analisado, retornava erro caso determinados caracteres estivessem presentes. Foi necessário incluir outro fluxo de exceção no código da ETL para contornar este erro.

No último dia de coleta de resultados, os *bugs* descritos haviam sido corrigidos e todos os fluxos foram executados sem apresentar erros, atingindo uma taxa de resposta de 100%.

A segunda análise diz respeito ao requisito de alta velocidade de atendimento e a métrica escolhida foi o SLA médio das resposta, isto é, a média de tempo que o sistema demora para dar uma resposta. É relevante destacar que apenas fluxos completos de ponta a ponto foram considerados na conta.

Figura 14 – Gráfico: Evolução do tempo médio de resposta

A partir do gráfico da figura 14, observa-se que tempo de resposta médio foi consideravelmente maior nos dois primeiros dias, cerca de duas vezes o valor dos dias seguintes. O motivo principal dessa diferença é que as filas do Amazon SQS estavam configuradas para fazer uma passagem de mensagens em *batch*, isto é, várias mensagens por vez e não na medida em que eram recebidas. O fluxo foi corrigido a partir do dia 28 de novembro e não foram observadas variações grandes no tempo de resposta, que permaneceu bem abaixo dos 10 minutos estabelecidos no planejamento de requisitos.

Tabela 5 – Resultados de Performance do Sistema

Chamadas	Respostas	Taxa de Resposta	SLA médio (minutos)
36	25	69,4%	05:32

O resumo dos resultados é mostrado na tabela 5. Conclui-se que o requisito de tempo de resposta foi atingido, porém a taxa de resposta foi inferior à especificada de 80%. Apesar disso, é importante ressaltar que a tendência de crescimento observada nos últimos dias de avaliação foi interrompida pela pausa no funcionamento do sistema.

8 CONSIDERAÇÕES FINAIS

O principal objetivo deste trabalho era a geração um sistema interativo que auxiliasse no combate à disseminação de desinformações em redes sociais. Com a criação e implementação de uma infraestrutura funcional de coleta e processamento de dados, acoplada ao Twitter através de um perfil automatizado, que analisa as postagens a partir de um modelo de aprendizado de máquina e permite aos usuários obter de forma rápida e prática informações que podem indicar que uma publicação é suspeita, esse objetivo foi atingido parcialmente. As complicações com o acesso à API da rede social impedem o uso do sistema pelo público geral neste momento, o que era parte do objetivo primário.

Apesar do imprevisto, os testes e avaliações do sistema mostraram eficiência e adequação da arquitetura desenvolvida, levando em conta o *deploy* e funcionamento de um sistema sem custos, com um banco de dados escalável e com um tempo de atendimento abaixo do colocado como ideal e uma taxa de resposta razoável do *bot* às marcações, atendendo aos requisitos definidos para o projeto, ainda que no curto espaço de tempo em que foi possível utilizá-lo em sua última versão. É importante ressaltar que a taxa de resposta foi incrementada conforme correções eram aplicadas para erros e falhas detectadas em produção, e que a amostragem relativamente baixa de execuções, causada pela revogação do acesso à rede social, limitou a qualidade dessa validação do sistema.

O objetivo secundário do projeto era agregar valor à comunidade científica que desenvolve modelos de aprendizado de máquina no contexto de *fake news*, na forma de bases de dados anotadas, *insights* e o código fonte do sistema capaz de gerar essas bases, de acordo com a especificação no Capítulo 3. Considera-se que este objetivo foi atingido parcialmente: apesar de o *dataset* não ser grande o suficiente para ser utilizado, devido as limitações de coleta impostas pelo Twitter, os dados coletados e gerados pelo projeto e o código fonte responsável por sua criação estão disponíveis em um repositório público no GitHub[1] e podem ser utilizados em eventuais pesquisas e na construção de ferramentas de anotação automática de *tweets* em outros contextos, conforme detalhado na Seção 6.3.

[1] https://github.com/Projeto-LIBRA/libra-dataset

Além disso, a proximidade com o IBRA USP e suas diversas frentes, descrita na Seção 4.3, dá bons indícios que o projeto terá sua contribuição continuada. Portanto, considera-se que existe a necessidade de uma reformulação do sistema e uma coleta mais extensiva de dados para que o trabalho atinja o impacto planejado em sua completude.

8.1 Trabalhos futuros

Apesar da baixa adesão aos resultados frente às ambições inciais e das limitações impostas ao trabalho em sua fase final, foram gerados artefatos com potencial de contribuição para os públicos-alvo, na forma de um modelo de aprendizado de máquina acessível para o usuário comum de rede social, da base de dados anotada e o código fonte do sistema que gerou as bases, que podem ser mais amplamente divulgados após ajustes no projeto. Assim, acredita-se que existem trabalhos futuros relevantes levando em conta o escopo planejado.

Para contornar a limitação elucidada pela revogação da chave de acesso à API do Twitter, o trabalho mais crítico é desacoplar o sistema de uma rede social específica, criando alternativas que permitam sua execução de forma mais independente de sistemas de terceiros. Essa melhoria pode ser implementada, por exemplo, através de uma página *web* onde o usuário insere entradas manualmente. É importante ressaltar que esta seria apenas uma segunda opção no caso de alguma indisponibilidade do sistema, uma vez que um dos diferenciais do projeto é a facilidade do público geral acessá-lo. Assim, podem ser consideradas também integrações com outras redes sociais que disponibilizem seus dados publicamente, de forma a ampliar o alcance do trabalho e reduzir sua dependência do Twitter.

O sistema pode ainda ser incrementado com mais etapas de execução. Para aumentar a qualidade e variedade de suas respostas, podem ser adicionados, por exemplo, modelos baseados em *Hierarchical Propagation Networks* (Seção 2.3.1) e o modelo *dEFEND* treinado a partir de *datasets* mais específicos do contexto brasileiro. Para melhorias na consulta aos artefatos gerados no banco de dados, existe a possibilidade de criação de *scripts* de processamento dos dados em formatos mais adequados para grandes quantidades, com disponibilização em um serviço de banco de dados com consultas em linguagem. Isso poderia ser feito, por exemplo, adicionando um passo a mais de processamento de dados para converter os resultados do sistema ao formato *Parquet*, que é consideravelmente mais performático para leituras que o JSON, além de ocupar menos espaço em disco (KUKREJ, 2020), e uso de serviços que permitem consultas em SQL a arquivos

armazenados no *Amazon S3*.

REFERÊNCIAS

ALCOFORADO, A. et al. Zeroberto: Leveraging zero-shot text classification by topic modeling. In: SPRINGER. *International Conference on Computational Processing of the Portuguese Language*. [S.l.], 2022. p. 125–136.

BAPTISTA, R. Redes sociais influenciam voto de 45% da população, indica pesquisa do DataSenado. *Senado Notícias*, 2019. Disponível em: ⟨https://www12.senado.leg.br/noticias/materias/2019/12/12/redes-sociais-influenciam-voto-de-45-da-populacao-indica-pesquisa-do-datasenado⟩. Acesso em: 22 abril 2022.

BELLEGARDA, J. R. Spoken language understanding for natural interaction: The Siri experience. In: MARIANI, J. et al. (Ed.). *Natural Interaction with Robots, Knowbots and Smartphones*. New York, NY: Springer New York, 2014. p. 3–14. ISBN 978-1-4614-8280-2.

BENJELLOUN, S. et al. Big data processing: batch-based processing and stream-based processing. In: IEEE. *2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)*. [S.l.], 2020. p. 1–6.

BROWN, P. F. et al. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, v. 19, n. 2, p. 263–311, 1993.

CHANG et al. Design of a NLP-empowered finance fraud awareness model: the anti-fraud chatbot for fraud detection and fraud classification as an instance. *Journal of Ambient Intelligence and Humanized Computing*, Springer, p. 1–17, 2022.

CPOP. Eleições 2018: a relação entre fake news e os candidatos Jair Bolsonaro e Fernando Haddad. *CPOP-UFPR*, 2018. Disponível em: ⟨http://www.cpop.ufpr.br/portal/eleicoes-2018-a-relacao-entre-fake-news-e-os-candidatos-jair-bolsonaro-e-fernando-haddad/⟩. Acesso em: 20 abril 2022.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

DONG. A survey on deep learning and its applications. *Computer Science Review*, Elsevier, v. 40, p. 100379, 2021.

DORASH, M. Machine Learning vs. Rule Based Systems in NLP. *Medium*, 2017. Disponível em: ⟨https://medium.com/friendly-data/machine-learning-vs-rule-based-systems-in-nlp-5476de53c3b8⟩. Acesso em: 20 abril 2022.

DOURADO, T. M. S. G. Fake news na eleição presidencial de 2018 no Brasil. Instituto de Humanidades, Artes e Ciências Professor Milton Santos, 2020.

DUFFY. Caos do Twitter vira público enquanto Musk gera conflito e demite funcionários. *CNN*, 2022. Disponível em: ⟨https://www.cnnbrasil.com.br/business/caos-do-twitter-vira-publico-enquanto-musk-gera-conflito-e-demite-funcionarios/⟩. Acesso em: 11 dezembro 2022.

EISENSTEIN, J. *Introduction to natural language processing*. [S.l.]: MIT press, 2019.

ENGLER, R. The making of the cours de linguistique générale. *The Cambridge Companion to Saussure*, Cambridge: Cambridge University Press, p. 47–58, 2004.

FERRAZ, T. P. et al. Debacer: a method for slicing moderated debates. In: SBC. *Anais do XVIII Encontro Nacional de Inteligência Artificial e Computacional*. [S.l.], 2021. p. 667–678.

FILHO, M. et al. Checagem de fatos numa democracia em xeque: implementação da plataforma sem migué nas eleições municipais de São Luís. *Revista Observatório*, v. 7, n. 3, p. a5pt–a5pt, 2021.

HAMID, A. et al. Fake news detection in social media using graph neural networks and NLP techniques: A covid-19 use-case. *arXiv preprint arXiv:2012.07517*, 2020.

HAN et al. Graph neural networks with continual learning for fake news detection from social media. *arXiv preprint arXiv:2007.03316*, 2020.

HEDDERICH, M. A. et al. A survey on recent approaches for natural language processing in low-resource scenarios. *arXiv preprint arXiv:2010.12309*, 2020.

HLÁDEK, D.; STAŠ, J.; PLEVA, M. Survey of automatic spelling correction. *Electronics*, Multidisciplinary Digital Publishing Institute, v. 9, n. 10, p. 1670, 2020.

ISLAM, M. R. et al. Deep learning for misinformation detection on online social networks: a survey and new perspectives. *Social Network Analysis and Mining*, Springer, v. 10, n. 1, p. 1–20, 2020.

KEMP, S. Digital 2020: Brazil. *DataReportal - Global Digital Insights*, 2020. Disponível em: ⟨https://datareportal.com/reports/digital-2020-brazil⟩. Acesso em: 9 abril 2022.

KUKREJ, M. Data lake - comparing performance of known big data formats. *Towards Data Science*, 2020. Disponível em: ⟨https://towardsdatascience.com/data-lake-comparing-performance-of-known-big-data-formats-eace705b6fd8⟩. Acesso em: 29 novembro 2022.

LIU, Y. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In: *Proceedings of the AAAI conference on artificial intelligence*. [S.l.: s.n.], 2018. v. 32, n. 1.

LU et al. Gcan: Graph-aware co-attention networks for explainable fake news detection on social media. *arXiv preprint arXiv:2004.11648*, 2020.

MAJID et al. Ml revolution in NLP: A review of machine learning techniques in natural language processing. 2021.

MICROSOFT. *ETL (extrair, transformar e carregar) - Azure Architecture Center*. 2022. Disponível em: ⟨https://docs.microsoft.com/pt-br/azure/architecture/data-guide/relational-data/etl⟩. Acesso em: 20 abril 2022.

MOLNAR, C. *Interpretable machine learning*. [S.l.]: Lulu. com, 2020.

MONTEIRO, R. A. et al. Contributions to the study of fake news in portuguese: New corpus and automatic detection results. In: SPRINGER. *International Conference on Computational Processing of the Portuguese Language*. [S.l.], 2018. p. 324–334.

MORENO et al. FACTCK. BR: a new dataset to study fake news. In: *Proceedings of the 25th Brazillian Symposium on Multimedia and the Web*. [S.l.: s.n.], 2019. p. 525–527.

MORRIS, J. X. et al. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. *arXiv preprint arXiv:2005.05909*, 2020.

NAYAK, P. *Understanding searches better than ever before*. 2019. ⟨https://blog.google/products/search/search-language-understanding-bert/⟩. (Accessed on 04/23/2022).

OSHIKAWA et al. A survey on natural language processing for fake news detection. *arXiv preprint arXiv:1811.00770*, 2018.

OTTER. A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, IEEE, v. 32, n. 2, p. 604–624, 2020.

PALACIOS, M. Fake news e a emergência das agências de checagem: terceirização da credibilidade jornalística. *Políticas da língua, da comunicação e da cultura no espaço lusófono. Vila Nova de Famalicão: Edições Humus*, p. 77–92, 2019.

PASQUINI, P. 90% dos eleitores de Bolsonaro acreditaram em fake news, diz estudo. *Folha*, 2018. Acesso em: 20 abril 2022. Disponível em: ⟨https://www1.folha.uol.com.br/poder/2018/11/90-dos-eleitores-de-bolsonaro-acreditaram-em-fake-news-diz-estudo.shtml⟩.

POLETTO, F. et al. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, Springer, v. 55, n. 2, p. 477–523, 2021.

RODRIGUEZ, J. F. A natural language processing approach to fraud detection. Italy, 2020.

RUDER, S. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

SANTAHOLMA, M. E. Grammar sharing techniques for rule-based multilingual NLP systems. In: *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA)*. [S.l.: s.n.], 2007.

SHU, K. et al. defend: Explainable fake news detection. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. [S.l.: s.n.], 2019. p. 395–405.

SHU, K. et al. FakeNewsNet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media. *arXiv preprint arXiv:1809.01286*, 2018.

SHU, K. et al. Hierarchical propagation networks for fake news detection: Investigation and exploitation. In: *Proceedings of the International AAAI Conference on Web and Social Media*. [S.l.: s.n.], 2020. v. 14, p. 626–637.

SILVA, S. Detecção de discurso de ódio em português usando cnn combinada a vetores de palavras. In: *Proceedings of KDMILE 2018, Symposium on Knowledge Discovery, Mining and Learning, São Paulo, SP, Brazil*. [S.l.: s.n.], 2018.

SINGHAL, S. et al. Spotfake+: A multimodal framework for fake news detection via transfer learning (student abstract). In: *Proceedings of the AAAI conference on artificial intelligence*. [S.l.: s.n.], 2020. v. 34, n. 10, p. 13915–13916.

SONG, J. et al. Deep learning-based extraction of predicate-argument structure (pas) in building design rule sentences. *Journal of Computational Design and Engineering*, Oxford University Press, v. 7, n. 5, p. 563–576, 2020.

TIKITO et al. Data collect requirements model. In: *Proceedings of the 2nd international Conference on Big Data, Cloud and Applications*. [S.l.: s.n.], 2017. p. 1–7.

TORFI, A. et al. Natural language processing advancements by deep learning: A survey. *arXiv preprint arXiv:2003.01200*, 2020.

TRE-MT. Fake news - agências de checagem desmontam boatos sobre a urna eletrônica. *TRE-MT*, 2018. Disponível em: ⟨https://www.tre-mt.jus.br/eleicoes/eleicoes-plebiscitos-e-referendos/eleicos-anteriores/eleicoes-2018/fakenews⟩. Acesso em: 20 abril 2022.

VALENTE, J. Fake news sobre candidatos inundam redes sociais em período eleitoral. *Agência Brasil*, 2018. Disponível em: ⟨https://agenciabrasil.ebc.com.br/geral/noticia/2018-10/um-dia-da-eleicao-fake-news-sobre-candidatos-inundam-redes-sociais⟩. Acesso em: 20 abril 2022.

VASSILIADIS, P. A survey of extract–transform–load technology. *International Journal of Data Warehousing and Mining (IJDWM)*, IGI Global, v. 5, n. 3, p. 1–27, 2009.

WU, L. et al. Misinformation in social media: definition, manipulation, and detection. *ACM SIGKDD Explorations Newsletter*, ACM New York, NY, USA, v. 21, n. 2, p. 80–90, 2019.

WU, Y. et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

XIE, N. et al. Explainable deep learning: A field guide for the uninitiated. 2020.

APÊNDICE A – QUESTIONÁRIO DE AVALIAÇÃO

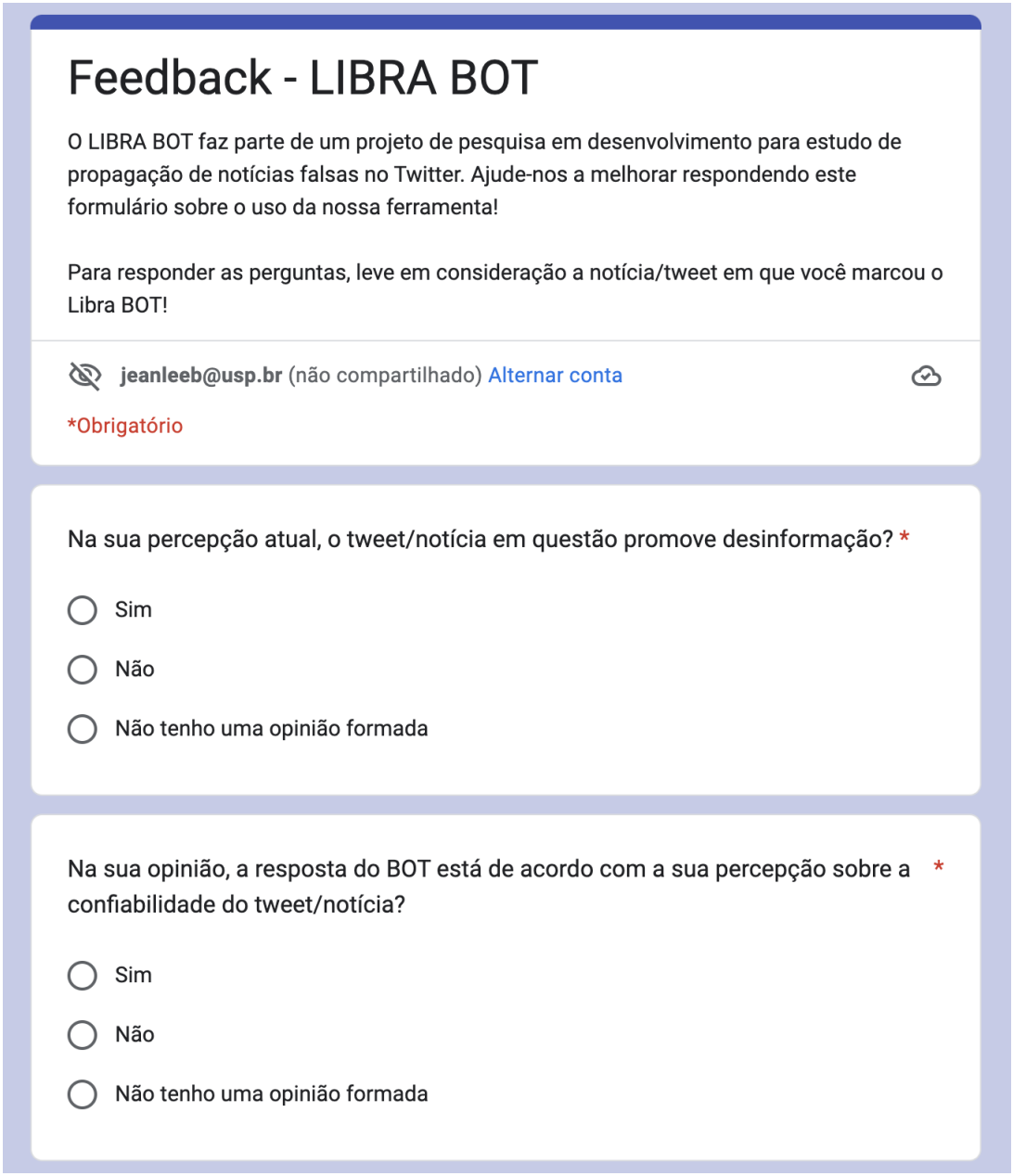


Figura 15 – Parte 1 do questionário de avaliação do projeto

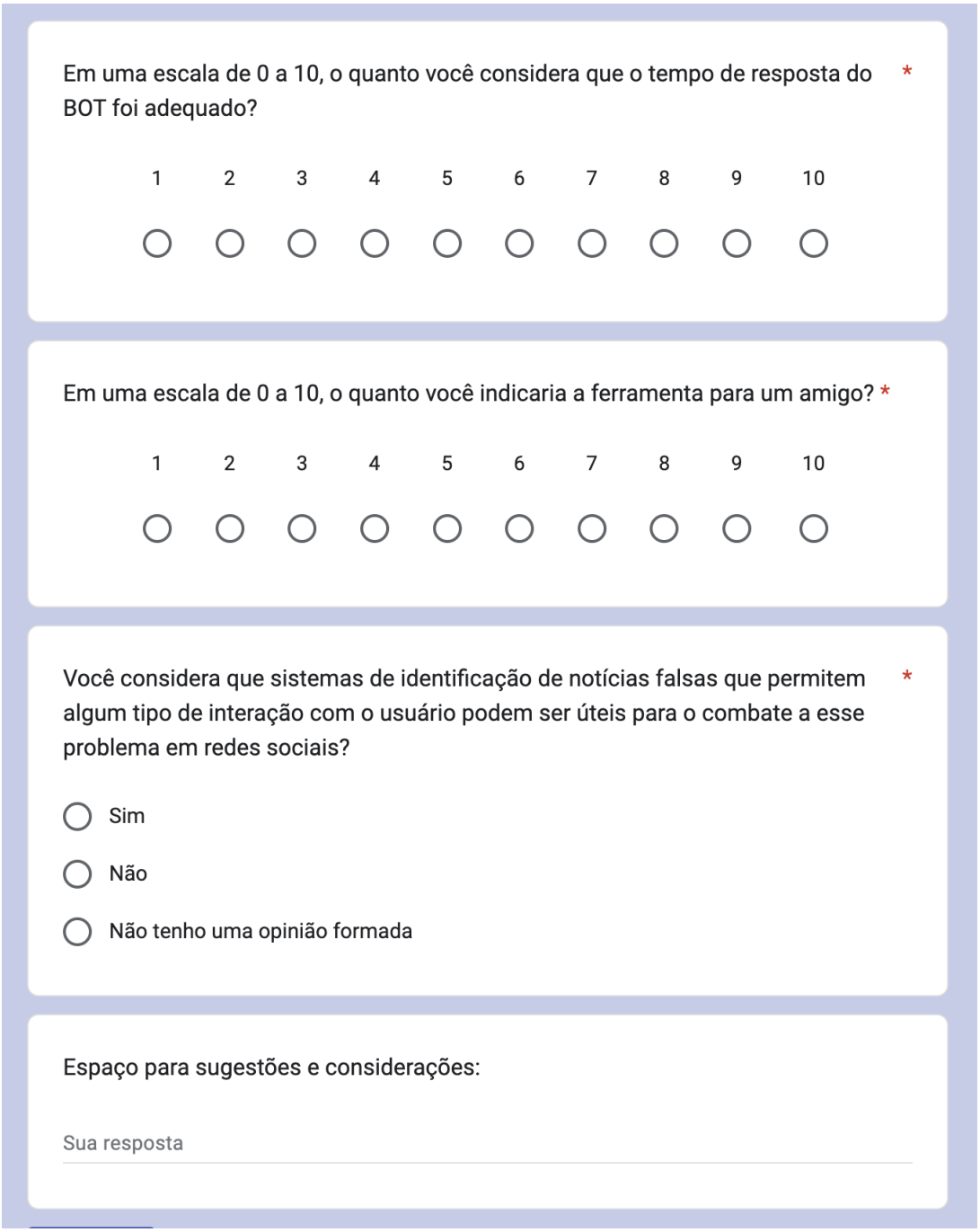


Figura 16 – Parte 2 do questionário de avaliação do projeto