Aplicação de Modelo LLAMA 2 para envio de lembretes através do Telegram

Rogério Hoepers Vitorassi
UDC
Santa Terezinha de Itaipu, Brasil
rogeriohv83@gmail.com

Miguel Diogenes Matrakas
UDC
Foz do Iguaçu, Brasil
mdmatraks@gmail.com

Alessandra Bussador
UDC
Foz do Iguaçu, Brasil
alebussador@gmail.com

***Abstract*—This project presents the development of a *chatbot* prototype aimed at improving communication between the Health Department of Foz do Iguaçu and users of the public health system. Utilizing the LLaMA-2-7b-chat model integrated into Telegram, the *chatbot* is currently capable of sending automatic appointment reminders, but it will eventually allow patients to confirm, reschedule, or cancel their appointments.**

***Keywords*—*chatbot*; Telegram; LLAMA 2.**

***Resumo*—Este projeto apresenta o desenvolvimento de um protótipo de *chatbot* voltado para melhorar a comunicação entre a Secretaria de Saúde de Foz do Iguaçu e os usuários do sistema público de saúde. Utilizando o modelo LLaMA-2-7b-chat integrado ao Telegram, o *chatbot* atualmente é capaz de enviar lembretes automáticos sobre consultas mas futuramente ele irá permitir que os pacientes confirmem, remarquem ou cancelem seus horários.**

***Palavras-chave*—*chatbot*; Telegram; LLAMA 2.**

I. Introdução

A evolução tecnológica está transformando o atendimento em instituições públicas e privadas, incluindo o setor de saúde, onde a eficiência e qualidade dos serviços são constantemente buscadas. Nesse cenário, os *chatbots*, sistemas baseados em Inteligência Artificial, se destacam por oferecer um atendimento rápido e personalizado. A Secretaria de Saúde de Foz do Iguaçu enfrenta desafios na comunicação com os pacientes, como a dificuldade em confirmar ou remarcar consultas por telefone, levando à ociosidade nos atendimentos.

Um estudo em 2020 [1] evidenciou que a adoção de *chatbots* pode reduzir a necessidade de atendimentos humanos, otimizando os processos. Aplicar essa tecnologia na Secretaria de Saúde pode ser uma solução para melhorar a comunicação com os cidadãos, permitindo o agendamento e remarcação de consultas de forma mais eficiente. Portanto a pergunta de pesquisa é: Como desenvolver e empregar um *chatbot* de atendimento para confirmar, remarcar ou cancelar horários se necessário?

A. Justificativa:

A necessidade da Secretaria de Saúde em melhorar a comunicação com os pacientes justifica a criação de um *chatbot*. Os métodos atuais, como chamadas telefônicas, são ineficazes, gerando esquecimentos e consultas não realizadas. Um *chatbot* pode minimizar esses problemas, melhorando o número de atendimentos realizados.

B. Objetivos:

O trabalho visa desenvolver um protótipo de *chatbot* para aprimorar a comunicação e enviar lembretes aos pacientes, além de oferecer suporte educativo. Isso beneficiará tanto os pacientes quanto a Secretaria, otimizando o agendamento. Objetivos Específicos:

- Ajustar o protótipo para interação natural no agendamento de consultas;
- Criar uma base de dados de chatlogs para futuros ajustes;
- Integrar o *chatbot* à API do Telegram.

II. Fundamentação Teórica

A. Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) tem evoluído significativamente nas últimas décadas, acompanhando os avanços na inteligência artificial. Inicialmente, os modelos de linguagem eram baseados em técnicas estatísticas, utilizadas em tarefas como reconhecimento de fala, tradução automática e recuperação de informação. Esses modelos, apesar de úteis, apresentavam limitações em termos de precisão e escalabilidade. Com a introdução do aprendizado profundo (*deep learning*), os modelos passaram a capturar melhor as relações sequenciais em dados linguísticos, gerando saídas mais coerentes e contextualmente adequadas.

Nos últimos anos, os modelos baseados em atenção, exemplificados pela arquitetura *Transformer*, revolucionaram o campo

do PLN. Essa abordagem, que utiliza técnicas de autoatenção para focar em segmentos específicos de sequências de entrada, mostrou-se eficaz em diversas tarefas, como modelagem de linguagem, tradução e geração de textos. O Google BERT, lançado em 2018, marcou um grande avanço ao introduzir uma compreensão bidirecional do contexto das palavras. Em seguida, os modelos GPT-2 e GPT-3 da OpenAI elevaram ainda mais a qualidade da geração de texto, produzindo saídas quase indistinguíveis das escritas por humanos. A partir de 2021, diversas colaborações e novos modelos, como o Claude da Anthropic, o GLAM do Google e o Gopher da *DeepMind*, continuaram a expandir os limites do PLN, abordando tanto a compreensão contextual quanto a geração de diálogos mais naturais [2].

O desenvolvimento desses modelos tem impacto direto na interação homem-máquina, permitindo comunicações mais humanizadas em diversas aplicações, desde assistentes virtuais até sistemas de atendimento ao cliente. No núcleo desses avanços estão técnicas como a tokenização, que envolve a segmentação do texto em unidades menores (*tokens*) [3].

Outro conceito essencial no PLN é a representação de palavras ou *tokens* por meio de *embeddings*, que transformam palavras em vetores numéricos, viabilizando o aprendizado de máquina. Técnicas como *Word2Vec* e *GloVe* tornaram-se populares ao representarem palavras em um espaço vetorial, permitindo que os modelos capturem a semântica e as relações contextuais entre as palavras. Esses modelos, baseados em redes neurais, são treinados para prever a próxima palavra em uma sentença com base nas anteriores, aprendendo padrões complexos a partir de vastos conjuntos de dados. Essa capacidade de capturar dependências contextuais tem levado a avanços em aplicações como tradução automática, análise de sentimentos e *chatbots* [2].

B. Grandes Modelos de Linguagem

Os grandes modelos de linguagem (LLM - do inglês: Large Language Models) são uma categoria avançada de inteligência artificial projetada para processar e entender a linguagem humana em grande escala. Baseados principalmente em arquiteturas *transformer*, esses modelos utilizam técnicas de aprendizado profundo e são treinados em vastos conjuntos de dados textuais. Sua principal característica é a capacidade de aprender padrões complexos, representar semântica e captar relações contextuais na linguagem natural, o que os torna capazes de gerar textos semelhantes aos escritos por humanos, responder perguntas, traduzir idiomas, e realizar outras tarefas de processamento de linguagem [2].

Modelos como o ChatGPT e LLaMA 2 são exemplos conhecidos de LLM que têm expandido as fronteiras do que é possível em termos de geração de linguagem, compreensão e tradução. Graças à sua versatilidade e alto desempenho, esses modelos têm sido amplamente utilizados em aplicações como *chatbots*, tradução automática, análise de sentimentos e criação de conteúdo [2].

C. chatbots

A história dos assistentes de conversação, ou *chatbots*, tem raízes na ficção científica, como nos livros de Isaac Asimov, que previam a presença de mecanismos inteligentes com características humanas. Em 1950, Alan Turing introduziu o ”Teste de Turing”, que buscava determinar se uma máquina poderia enganar um humano durante uma conversa, conceito que permanece relevante hoje com o avanço dos *chatbots*. Assistentes virtuais, como os utilizados em empresas para atendimento ao cliente, analisam questões e respondem de forma adequada, interagindo em linguagem natural por meio de e-mails, mensagens, redes sociais ou voz [4].

Existem diferentes tipos de *chatbots*, variando conforme o nível de controle da interação. Existem as conversas que são impulsionadas pelo *chatbot*, onde as interações seguem roteiros predefinidos e limitados, enquanto outros são mais flexíveis, permitindo que o usuário tenha maior controle sobre as respostas. Em interações impulsionadas pelo usuário, os *chatbots* precisam identificar a intenção do usuário e responder de maneira apropriada [5].

As relações entre humanos e *chatbots* podem ser de curto ou longo prazo. Em relações de curto prazo, o usuário interage com o *chatbot* uma única vez, sem que suas informações sejam armazenadas. Já em relações de longo prazo, o *chatbot* utiliza perfis personalizados, como em assistentes pessoais que acompanham o histórico de interações. *chatbots* de longo prazo são comuns em plataformas como Facebook Messenger e são usados para conteúdos complexos ou *coaching* [5].

Exemplos práticos incluem *chatbots* de suporte ao cliente e assistentes pessoais. Os primeiros são projetados para interações curtas e focadas em resolver problemas específicos dos usuários, sendo frequentemente integrados a sistemas de CRM para criar relacionamentos duradouros. Já os assistentes pessoais ajudam os usuários em tarefas diárias, como buscar informações ou controlar dispositivos, com o objetivo de facilitar a interação e alcançar o resultado desejado de forma eficiente [5].

D. Integração no Telegram

A integração com a API do Telegram oferece uma plataforma poderosa para o desenvolvimento de aplicações, especialmente *chatbots*. A API do Telegram permite a criação de *bots* que atuam como interfaces para programas executados em servidores, facilitando a comunicação entre usuários e sistemas de mensagens. Esses *bots* não exigem números de telefone adicionais e interagem com os servidores por meio de uma interface HTTPS, que gerencia a criptografia e a comunicação [6]. Outra ferramenta relevante é a *Telegram Database Library (TDLib)*, que simplifica a criação de aplicativos Telegram rápidos e seguros, cuidando de aspectos como rede, criptografia e armazenamento local, liberando os desenvolvedores para focarem em design e interface [7].

Além disso, a API aberta do Telegram permite a criação de clientes personalizados e o estudo de aplicativos já existentes para entender seu funcionamento [7]. Comparativamente, o WhatsApp também possui uma API que oferece recursos de processamento de linguagem natural, ideal para *bots* empresariais. No entanto, sua configuração é mais complexa e envolve custos, além de ser necessário ativar a API com um número de celular específico. Colaborações com fornecedores como Twilio podem facilitar o desenvolvimento de *bots* no WhatsApp, mas, em geral, as APIs do Telegram se destacam por serem mais acessíveis e abertas, oferecendo maior flexibilidade para desenvolvedores [8].

III. Desenvolvimento

Nesta secção será detalhado o processo de desenvolvimento do *chatbot* utilizando o modelo LLAMA2, integrado com um banco de dados de teste. O objetivo do sistema é enviar lembretes de consultas agendadas 24 horas antes do horário marcado, com base em um banco de dados contendo informações fictícias de pacientes.

A. Estrutura Geral do Sistema

Um protótipo de teste foi implementado utilizando a linguagem Python, integrando um modelo de linguagem natural baseado no LLaMA-2-7b-chat, um banco de dados MySQL e um *bot* no Telegram para comunicação com os usuários. O desenvolvimento foi realizado em três etapas principais:

- Integração do Modelo LLaMA-2-7b-chat: O primeiro passo foi integrar o modelo de linguagem LLaMA-2-7b-chat ao projeto. Para isso, foi utilizada a API da *hugging face*, por meio das bibliotecas transformers e *huggingface_hub*. Essas ferramentas permitiram o carregamento do modelo, responsável por interpretar as mensagens dos usuários e gerar respostas adequadas com base no contexto. A integração também incluiu a autenticação com o token da *hugging face* e a configuração da arquitetura do modelo, garantindo a correta utilização durante a execução do bot. O LLaMA 2 foi escolhido por seu ótimo desempenho e por permitir uso gratuito, sem a necessidade de acesso a APIs pagas.
- Conexão com o Telegram e Testes Iniciais: Após a integração do modelo, a comunicação com o Telegram foi configurada utilizando a biblioteca python-telegram-bot. Esse processo permitiu que o *bot* enviasse e recebesse mensagens, possibilitando a interação em tempo real com os usuários. Durante essa fase, foram realizados testes para garantir que todo o fluxo de comunicação, desde a recepção das mensagens até a resposta gerada pelo modelo, estivesse funcionando corretamente. O Telegram foi escolhido por oferecer uma API gratuita, o que facilita a integração sem restrições ou custos, sendo ideal para protótipos como o proposto neste trabalho. No entanto, caso necessário, a substituição do Telegram pelo WhatsApp seria possível com algumas alterações no código.
- Desenvolvimento do Banco de Dados e Testes: A próxima etapa consistiu na criação de um banco de dados MySQL simples, contendo informações fictícias de pacientes, incluindo nomes, IDs do Telegram, datas e horários de consultas. A integração com o banco de dados foi realizada utilizando a biblioteca mysql-connector-python, que permitiu a conexão, execução de consultas SQL, e manipulação dos dados armazenados. Essa estrutura de dados foi fundamental para testar o sistema, simulando o envio de lembretes automáticos 24 horas antes do horário agendado para cada consulta.

B. Configuração e Ambiente de Desenvolvimento

Inicialmente, foi necessário configurar o ambiente com o Conda no Visual Studio Code, utilizando as bibliotecas necessárias, incluindo *transformers*, *mysql.connector* e *python-telegram-bot*. Além disso, foi utilizado o modelo LLaMA-2-7b-chat, disponível na *Hugging Face*, para gerar respostas e interpretar mensagens dos usuários. O código foi desenvolvido e testado em um ambiente com a GPU NVIDIA RTX 2060.

C. Desafios Encontrados

Os principais desafios encontrado durante o desenvolvimento foi a implementação de um sistema que pudesse processar mensagens dos usuários simultaneamente à verificação contínua de lembretes de consultas. Na fase atual de desenvolvimento,

está previsto apenas o início da conversação, fazendo com que o *bot* inicie o diálogo com a mensagem presente na figura 1.

Após a correção desse erro, será implementado aprimoramentos para permitir que o *bot* responda em tempo real enquanto realiza consultas no banco de dados, com um *log* de conversas e atualização automática de horários conforme as respostas dos pacientes.

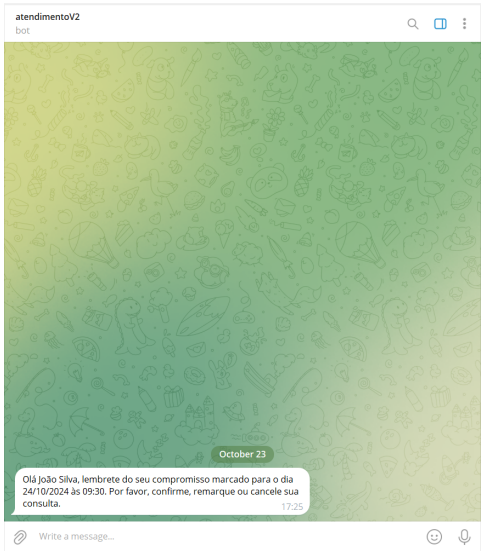

Fig. 1. Mensagem enviada pelo *chatbot*

IV. Materiais e Métodos

O estudo adotou uma pesquisa aplicada com uma abordagem mista (qualitativa e quantitativa), focada em desenvolver um protótipo de *chatbot* que facilite a comunicação entre a Secretaria de Saúde de Foz do Iguaçu e os usuários do sistema de saúde. O protótipo foi testado em um ambiente simulado no Telegram.

A Secretaria de Saúde foi escolhida como cenário de pesquisa devido à relevância do problema e à possibilidade de obter dados para o desenvolvimento do *chatbot*.

V. Conclusão

Em conclusão, o desenvolvimento deste protótipo de *chatbot* demonstra o potencial de soluções baseadas em inteligência artificial para aprimorar a comunicação entre instituições de saúde e seus usuários. A implementação do modelo Llama-2-7B-Chat-GGUF integrado ao Telegram irá permitir automatizar tarefas como o envio de lembretes de consultas, proporcionando uma experiência mais prática e acessível.

Embora a versão atual do *chatbot* se limite ao envio de lembretes, os próximos passos para este projeto incluem a ampliação das funcionalidades para permitir que os pacientes confirmem, remarquem ou cancelem seus horários, garantindo uma interação mais completa e eficiente. Para realizar isso, a ideia é utilizar o modelo LLaMA 2, para identificar a intenção do paciente ao responder ao lembrete e agir de forma automatizada. Além disso, como o *chatbot* está integrado com um banco de dados MySQL, as informações de agendamentos e interações serão atualizadas em tempo real, permitindo ao sistema registrar respostas e ajustar horários conforme necessário.

VI. Agradecimentos

Gostaria de expressar meus sinceros agradecimentos às equipes do SMTI e SMSP de Foz do Iguaçu, que se dispuseram a colaborar de forma para a realização deste trabalho. Agradeço pela disponibilidade, suporte e contribuição valiosa, que foram fundamentais para o desenvolvimento desta pesquisa.

Referências

[1] V. A. Lugli and J. de Lucca Filho, *O uso do chatbot para a excelência em atendimento*, Revista Interface Tecnológica, vol. 17, no. 1, pp. 205-218, Jul. 2020. DOI: 10.31510/infa.v17i1.840.

[2] A. Kulkarni, A. Shivananda, A. Kulkarni, and D. Gudivada, *Applied Generative AI for Beginners*. Apress, 2023. DOI: 10.1007/978-1-4842-9994-4.

[3] H. M. Caseli and M. G. V. Nunes, *Processamento de linguagem natural: conceitos, técnicas e aplicações em português*, BPLN, 2023.

[4] L. T. Cruz, A. J. Alencar, and E. A. Schmitz, *Assistentes Virtuais Inteligentes e Chatbots: um guia prático e teórico sobre como criar experiências e recordações encantadoras para os clientes da sua empresa*. Brasport. [Online]. Available: https://books.google.com.br/books?id=iSCADwAAQBAJ. ISBN: 9788574529097.

[5] A. Følstad, M. Skjuve, and P. B. Brandtzaeg, "Different *chatbot*s for different purposes: Towards a typology of *chatbot*s to understand interaction design," in *Lecture Notes in Computer Science*, vol. 11551, Springer, 2019, pp. 145–156. DOI: 10.1007/978-3-030-17705-8_13.

[6] K. A. Nugraha and D. Sebastian, "Designing consultation *chatbot* using telegram API and webhook-based nodejs applications," in *7th International Conference on Education and Technology (ICET 2021)*, Atlantis Press, 2021, pp. 119-122.

[7] Telegram FZ LLC and Telegram Messenger Inc., *Telegram Bot API*, 2024. [Online]. Available: https://core.telegram.org/bots/api. [Accessed: 23-May-2024].

[8] N. A. Khan and J. Albatein, "COVIBOT-An intelligent WhatsApp based advising bot for Covid-19," in *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, IEEE, 2021, pp. 418-422.