Reconhecimento de voz para a personalização de LLM's

1st Daniel Fazzioni
Bacharelado de Inteligência Artificial
Universidade Federal de Goiás (UFG)
Goiânia, Goiás
fazzioni@discente.ufg.br

2nd Gustavo Luiz Bueno Pereira
Bacharelado de Inteligência Artificial
Universidade Federal de Goiás (UFG)
Goiânia, Goiás
gustavobueno@discente.ufg.br

3rd Pedro Schindler Freire Brasil Ribeiro
Universidade Federal de Goiás (UFG)
Universidade Federal de Goiás (UFG)
Goiânia, Goiás
schindler@discente.ufg.br

4th Thiago Pedroso de Jesus
Bacharelado de Inteligência Artificial
Universidade Federal de Goiás (UFG)
Goiânia, Goiás
thiagopedroso@discente.ufg.br

***Abstract*—Este artigo explora o uso de tecnologias de reconhecimento de voz e personalização em assistentes virtuais, destacando a importância do processamento de linguagem natural e aprendizado de máquina nesse contexto. O foco principal é analisar como a evolução dessas tecnologias permite uma interação mais natural e intuitiva entre humanos e máquinas, e como o reconhecimento individual de usuários através de suas vozes pode levar a experiências personalizadas significativamente melhoradas. Abordamos os fundamentos técnicos, os desafios e as implicações éticas envolvidas, além de discutir as aplicações práticas e futuras possibilidades dessa tecnologia.**

***Index Terms*—speech recognition, large language model, virtual assistant, neural network**

I. Introdução e revisão bibliográfica

A trajetória da tecnologia de reconhecimento de voz, especialmente no contexto de assistentes virtuais, representa um campo fascinante de estudo. Historicamente, os avanços nessa área refletem uma busca incessante por interfaces mais naturais e intuitivas entre humanos e máquinas. Este trabalho investiga a evolução do reconhecimento de voz e sua aplicação na personalização de experiências com assistentes virtuais. A capacidade desses sistemas de entender comandos verbais e responder de maneira contextualizada abre caminhos para uma interação personalizada, moldando significativamente nosso relacionamento com a tecnologia. Vários estudos e desenvolvimentos recentes, incluindo avanços em processamento de linguagem natural (NLP) e aprendizado de máquina, fornecem um pano de fundo para nossa pesquisa. Este artigo propõe explorar métodos e técnicas inovadoras empregadas na identificação de usuários através de suas vozes, visando aprimorar a funcionalidade e a personalização dos assistentes virtuais. [1].

II. Fundamentos Teóricos

Neste segmento, detalhamos as técnicas fundamentais utilizadas no reconhecimento de voz e na personalização de assistentes virtuais. A abordagem central envolve a análise e interpretação de ondas sonoras, transformando-as em dados digitais que podem ser processados e compreendidos por algoritmos de inteligência artificial. Por meio do python, essa conversão é realizada através de técnicas avançadas que incluem o processamento de sinais acústicos e a modelagem linguística. Além disso, exploramos o uso de modelos de linguagem probabilísticos para prever e entender sequências de palavras em contextos específicos de fala. A identificação do usuário através da voz é alcançada por meio de algoritmos que reconhecem padrões únicos no timbre e na cadência da fala, permitindo que o assistente virtual adapte suas respostas de acordo com as preferências individuais do usuário. Examinamos também as implicações éticas e os desafios associados ao reconhecimento de voz, como a necessidade de considerar diferentes dialetos e minimizar preconceitos. Este estudo visa não apenas aprofundar a compreensão dos aspectos técnicos, mas também explorar as vastas possibilidades e aplicações futuras do reconhecimento de voz na personalização da experiência do usuário com assistentes virtuais.

A. Reconhecimento de Voz

O reconhecimento de voz no nosso projeto é implementado através do modelo "SpeakerVerification_en_titanet_large", instalado a partir da biblioteca "NeMo", da NVIDIA. O modelo é capaz de extrair características vocais únicas do locutar, comportando-se como uma ferramenta de IA conversacional que abrange o reconhecimento automático de fala (ASR), processamento de linguagem natural (PLN) e síntese de texto para fala (TTS). Nele, utilizamos a técnica pré-implementada pela biblioteca de verificação de falantes, onde o método *verify_speakers* é responsável por comparar dois áudios de fala e retornar *True* caso ambos tenham em seu conteúdo a voz de uma mesma pessoa, e *False* para caso contrário. Também vale citar que, por mais que o modelo tenha sido treinado somente com falas em inglês, utilizando datasets como *Voxceleb* e *Fisher*, apresentou uma performance confortável para esse caso de uso no idioma Português-BR.

B. Transcrição do aúdio

Também faz-se necessário a obtenção do conteúdo do áudio para ser utilizado como input do LLM. Com esse propósito, utilizamos o modelo *Faster Whisper*, uma reimplementação do Whisper [2] através da aplicação da biblioteca *CTranslate2*, a qual possui uma diversidade de técnicas de otimização de performance de modelos, como quantização de pesos, fusão de camadas, reordenação de *batch*, etc. Segundo dados do repositório oficial no GitHub, é quatro vezes mais rápida que o original, apresentando a mesma acurácia, usando menos memória. O Whisper, por si só, é um modelo da OpenAI de reconhecimento automático de fala (ASR) treinado em 680.000 horas de dados supervisionados multilíngues e multitarefas coletados da web, demonstrando robustez acentuada a sotaques, ruídos de fundo e linguagem técnica. Além disso, suporta a transcrição em múltiplos idiomas e a tradução desses idiomas para o inglês. No entanto, ao realizar testes empíricos no idioma Português-BR em ambos modelos, o *Faster Whisper* se saiu consideravelmente melhor, principalmente em falas mais curtas, além de possuir um tempo de processamento significativamente menor, justificando seu uso.

C. Assistente virtual

O coração do nosso assistente virtual é o GPT-4 [3], uma iteração avançada dos modelos de linguagem de grande escala (LLM) da OpenAI. Para personalizar a experiência do usuário, desenvolvemos uma arquitetura capaz de guardar e lembrar preferencias de cada usuário específico, criando uma persona virtual única para cada usuário. Isso é feito através da utilização de "Módulos" (Chamadas do GPT-4 com prompts e finalidades diferentes), orquestrados para realizar uma função mais complexa. No nosso caso temos os seguintes Módulos: Memorywrite e Answer. Memorywrite é o modulo cuja função é receber o histórico de conversas do usuário com a assistente (Chamamos de memória de curto prazo), e decidir quais informações são relevantes e devem ser guardadas (Chamamos de memória de longo prazo). Essa memória de longo prazo é então guardada em um dicionário do Python, diferenciada pelo nome do usuário. Answer é o módulo cuja função vai ser de uma assistente normal, recebendo o input do usuário e produzindo uma resposta. Para personalizar essa resposta para cada usuário, nós buscamos (De modo similar ao método Retrival Augmented Generation) a memória de longo prazo relativa a esse usuário, e colocamos esse conteúdo no módulo Answer. Isso resulta em respostas mais adaptadas, informativas e contextualizadas, melhorando significativamente a interação do usuário com o assistente.

D. Dataset de validação

No contexto do nosso projeto, optamos por utilizar primeira versão do dataset TED-LIUM [4] para a validação de nosso modelo de reconhecimento de locutores. Esta versão inicial do TED-LIUM se destaca por sua ampla gama de palestras TED em inglês, cada uma com a identificação de seus respectivos locutores. Este recurso é particularmente vantajoso para a validação de modelos de identificação de locutores, uma vez que oferece uma rica diversidade de vozes, condições de gravação e durações de áudio.

III. Metodologia

A metodologia utilizada neste projeto envolve a integração de diversas etapas, desde a coleta e processamento de áudio até a implementação de tecnologias avançadas de reconhecimento de voz e processamento de linguagem natural. O processo pode ser dividido nas seguintes fases:

A. Coleta e Preparação de Dados de Voz

Em nossa abordagem, armazenamos os nomes dos usuários e uma amostra de áudio para cada em um dicionário. Assim, dado que o fluxo do assistente virtual se inicia com um input de áudio, caso o falante responsável por esse input não seja reconhecido, provavelmente por não estar no dicionário, o usuário será redirecionado para um cadastro, onde digita seu nome, e o algoritmo fornece um exemplo de texto para leitura enquanto coleta a amostra de áudio, na intenção de resultar em aproximadamente um minuto de duração. O nome digitado também é utilizado para o cadastro do dicionário da memória de longo prazo.

B. Implementação do Reconhecimento de Voz com NeMo

O algoritmo itera cada par de chave (nome do usuário) e valor (respectiva amostra de áudio) do dicionário descrito na seção anterior e utiliza o modelo da biblioteca NeMo (específicado na seção de "Fundamentos Teóricos") para compará-los com o input de áudio e, assim que o modelo devolver um valor *True*, indicando que o usuário foi encontrado (isto é, as características do input são semelhantes às características da amostra cadastrada desse usuário), seu nome é retornado.

Utilizamos amostras de áudios mais "longas" para criar um perfil vocal detalhado na base de dados, visto que os comandos para o LLM provavelmente serão curtos e, em tempo real, a comparação entre áudios de curta duração apresenta pouca perfomance, mas, ao utilizar um mais longo como base de comparação, há um enorme aumento no desempenho do modelo.

C. Transcrição de Comandos de Voz com Whisper

Após o reconhecimento do falante, o áudio é inserido no *Faster Whisper*, o qual retorna uma transcrição do seu conteúdo.

D. Integração com GPT-4 - Assistente Virtual Personalizada

Por fim, tanto o nome quanto o conteúdo do input de áudio são inseridos no módulo "Answer" da arquitetura descrita e, durante a conversa, tanto esse conteúdo quanto as respostas retornadas são salvas na memória de curto prazo. Quando o usuário terminar o diálogo, a partir de um input contendo as palavras "Encerrar" e "Conversa", essa memória de curto praza é inserida no módulo "Memorywrite", utilizada para modificar a memória de longo prazo do usuário, e então é descartada.

O assistente virtual propriamente dita é a integração de todas as etapas descritas, onde são executadas em um loop que só se encerra com a ordem do usuário. Nessa versão

de implementação no ambiente *Jupyter Notebook*, a aplicação consiste em uma instância da classe *VirtualAssistent*, que possui a implementação de todos os processos descritos, portanto, a memória de longo prazo somente é "armazenada" na própria instância, assim, o diálogo sempre deve ser iniciado a partir dela, a não ser que a intenção seja reiniciar o assistente virtual.

E. Testes e Validação

Realizamos uma série de testes para validar a eficácia do sistema em diferentes cenários, avaliando a precisão do reconhecimento de voz, a qualidade da transcrição e a relevância das respostas do assistente virtual.

F. Validação com dataset maior

Por fim, empregamos uma metodologia rigorosa para avaliar a eficácia do modelo Nvidia NeMo em tarefas de reconhecimento de voz, utilizando o dataset Tedlium. Inicialmente, procedeu-se à importação das bibliotecas necessárias, incluindo o toolkit NeMo da Nvidia e a biblioteca datasets para manipulação de dados.

1) **Preparação do Dataset**: Carregou-se o *dataset Tedlium*, versão *Release 1*, conhecido por sua coleção de gravações de áudio de palestras TED e suas respectivas transcrições.
2) **Redução e Pré-processamento dos Dados**: Visando a eficiência no uso de recursos computacionais, o *dataset* foi reduzido para um tamanho gerenciável, selecionando-se um subconjunto de amostras para análise. Além disso, colunas desnecessárias foram removidas e os dados organizados para focar nas amostras de áudio e nos respectivos IDs dos locutores. As amostras de áudio foram salvas como arquivos WAV e associadas aos seus respectivos locutores.
3) **Teste de Verificação de Locutor**: O cerne da metodologia envolveu a verificação de locutores. Pares aleatórios de amostras de áudio foram selecionados, e o modelo NeMo foi utilizado para determinar se as duas amostras pertenciam ao mesmo locutor. Esse processo foi repetido para um número predefinido de pares, e os resultados foram categorizados em positivos verdadeiros, negativos verdadeiros, falsos positivos e falsos negativos.
4) **Testes Adicionais**: Nossos testes também incluem um código para adicionar novas amostras de áudio ao *dataset* e testar a capacidade do modelo de identificar o locutor de uma amostra de áudio com base no conjunto de dados existente.
5) **Avaliação de Desempenho**: Os resultados foram utilizados para construir uma matriz de confusão, oferecendo uma análise visual e quantitativa do desempenho do modelo na identificação correta de se duas amostras de áudio são do mesmo locutor.
6) **Conclusões**: Os resultados obtidos e representados na matriz de confusão da Figura 1 revelam um desempenho excelente do modelo Nvidia NeMo no reconhecimento de locutores utilizando o dataset Tedlium. Observa-se uma proeminente acurácia na distinção de vozes distintas, como indicado pelo alto número de Verdadeiros Negativos (99). Percebe-se, ainda, a precisão em identificar corretamente o locutor, apontada pelo valor verdadeiro positivo. Em conclusão, o modelo demonstra uma competência robusta para diferenciar locutores distintos e identificar corretamente o responsável pela fala, cumprindo com os requisitos do nosso projeto.

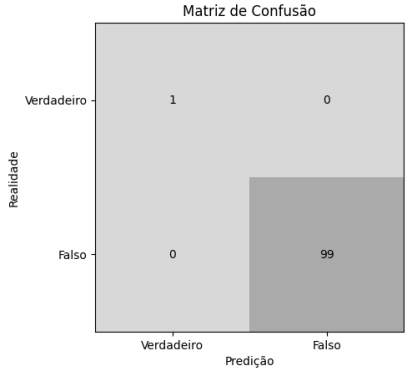

Fig. 1. Matriz de confusão do modelo de reconhecimento de voz para 70 locutores distintos.

IV. Resultados e Conclusões

Com a ascensão da IA, a personalização emergiu como um valor central, particularmente na forma como interagimos com a tecnologia. As técnicas de reconhecimento de fala, uma vez consideradas futuristas, são agora uma realidade tangível, impulsionando a personalização dos modelos de LLM. Esses avanços não são apenas teóricos, mas têm aplicações práticas significativas, refletindo uma tendência crescente no uso de ferramentas baseadas em IA para uma comunicação mais eficiente e personalizada.

Este trabalho examina como a integração de reconhecimento de fala com tecnologias de PLN e processamento de sinais e imagens pode criar soluções mais holísticas e adaptáveis. Através de uma análise aprofundada e do desenvolvimento de um projeto completo, nosso grupo demonstrou como essa interseção tecnológica pode resultar em sistemas mais eficazes e intuitivos. Ao fazê-lo, abrimos caminho para novas possibilidades em personalização, oferecendo insights valiosos para futuras inovações no campo da IA.

Para ilustrar a funcionalidade da nossa aplicação, produzimos um vídeo conciso. Assim, é possível obter uma visão clara do funcionamento da aplicação: "Clique Aqui"

References

[1] O. Kuchaiev, J. Li, H. Nguyen, O. Hrinchuk, R. Leary, B. Ginsburg, S. Kriman, S. Beliaev, V. Lavrukhin, J. Cook, P. Castonguay, M. Popova, J. Huang, and J. M. Cohen, "Nemo: a toolkit for building ai applications using neural modules," *arXiv preprint arXiv:1909.09577*, 2019, comments: 6 pages plus references. [Online]. Available: https://arxiv.org/abs/1909.09577

[2] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022, focus to learn more. [Online]. Available: https://arxiv.org/abs/2212.04356

[3] O. J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H. W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S. P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S. S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, and . additional authors not shown, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023, comments: 100 pages; updated authors list. [Online]. Available: https://arxiv.org/abs/2303.08774

[4] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Estève, "Ted-lium 3: twice as much data and corpus repartition for experiments on speaker adaptation," *arXiv preprint arXiv:1805.04699*, 2018, submitted to SPECOM 2018, 20th International Conference on Speech and Computer; TED-LIUM 3 corpus available on this https URL. [Online]. Available: https://arxiv.org/abs/1805.04699