UNIVERSIDADE FEDERAL DE GOIÁS / INSTITUTO DE INFORMÁTICA

Reconhecimento Automático de Fala (ASR)

Estudo sobre a Influência do Contexto no Desempenho e na Eficiência de Modelos

Daniel Ribeiro da Silva

UFG

UNIVERSIDADE
FEDERAL DE GOIÁS

UNIVERSIDADE FEDERAL DE GOIÁS (UFG)
INSTITUTO DE INFORMÁTICA (INF)

DANIEL RIBEIRO DA SILVA

Reconhecimento Automático de Fala (ASR)

Estudo sobre a Influência do Contexto no Desempenho e na Eficiência de Modelos

Goiânia
2025

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC no 1240/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei no 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo dos Trabalhos de Conclusão dos Cursos de Graduação disponibilizado no RI/UFG é de responsabilidade exclusiva dos autores. Ao encaminhar(em) o produto final, o(s) autor(a)(es)(as) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG)

Nome(s) completo(s) do(a)(s) autor(a)(es)(as): DANIEL RIBEIRO DA SILVA

Título do trabalho: Reconhecimento Automático de Fala (ASR)

Estudo sobre a Influência do Contexto no Desempenho e na Eficiência de Modelos

2. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador) Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante: a) consulta ao(à)(s) autor(a)(es)(as) e ao(à) orientador(a); b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo do TCCG. O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;

- Submissão de artigo em revista científica;

- Publicação como capítulo de livro.

Obs.: Este termo deve ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Daniel Ribeiro Da Silva**, **Discente**, em 10/01/2025, às 22:46, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Fernando Marques Federson**, **Professor do Magistério Superior**, em 15/01/2025, às 16:10, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020 .

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **5089541** e o código CRC **CD5C7261**.

Referência: Processo nº 23070.001552/2025-73 SEI nº 5089541

DANIEL RIBEIRO DA SILVA

Reconhecimento Automático de Fala (ASR)

Estudo sobre a Influência do Contexto no Desempenho e na Eficiência de Modelos

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Orientador: Prof. Dr. Fernando Marques Federson

Goiânia

2025

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

SILVA, DANIEL RIBEIRO DA
Reconhecimento Automático de Fala (ASR) [manuscrito] : Estudo sobre a Influência do Contexto no Desempenho e na Eficiência de Modelos / DANIEL RIBEIRO DA SILVA. - 2025.
90 f.

Orientador: Prof. Dr. Fernando Marques Federson.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Goiás, Instituto de Informática (INF), Inteligência Artificial, Goiânia, 2025.

1. inteligência artificial. 2. reconhecimento de fala. 3. contexto. I. Federson, Fernando Marques , orient. II. Título.

CDU 004

DANIEL RIBEIRO DA SILVA

Reconhecimento Automático de Fala (ASR)

Estudo sobre a Influência do Contexto no Desempenho e na Eficiência de Modelos

Relatório final de Trabalho de Conclusão de Curso, apresentado à Universidade Federal de Goiás, como parte das exigências para a obtenção do título de Bacharel em Inteligência Artificial.

Data da Aprovação: 17 de dezembro de 2024.

Prof. Dr. Fernando Marques Federson
Orientador (INF-UFG)

Prof. Dr. Aldo André Díaz Salazar
Coordenador de TCC do BIA (INF-UFG)

Prof. Dr. Anderson da Silva Soares
Coordenador do BIA (INF-UFG)

Me. Lucas Rafael Stefanel Gris
(CEIA-UFG)

DANIEL RIBEIRO DA SILVA

Reconhecimento Automático de Fala (ASR)

Estudo sobre a Influência do Contexto no Desempenho e na Eficiência de Modelos

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Reconhecimento Automático de Fala (ASR)**. Uma ilustração e sua narrativa descrevem os períodos de trabalho. Os Apêndices contêm os Termos de Aceite de Entrega e os resultados obtidos durante cada período de trabalho.

Palavras-chave: inteligência artificial, modelos grandes de linguagem, geração automática de datasets.

ABSTRACT

This Course Completion Report aims to bring together the results of my journey to become an expert in **Automatic Speech Recognition (ASR)**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

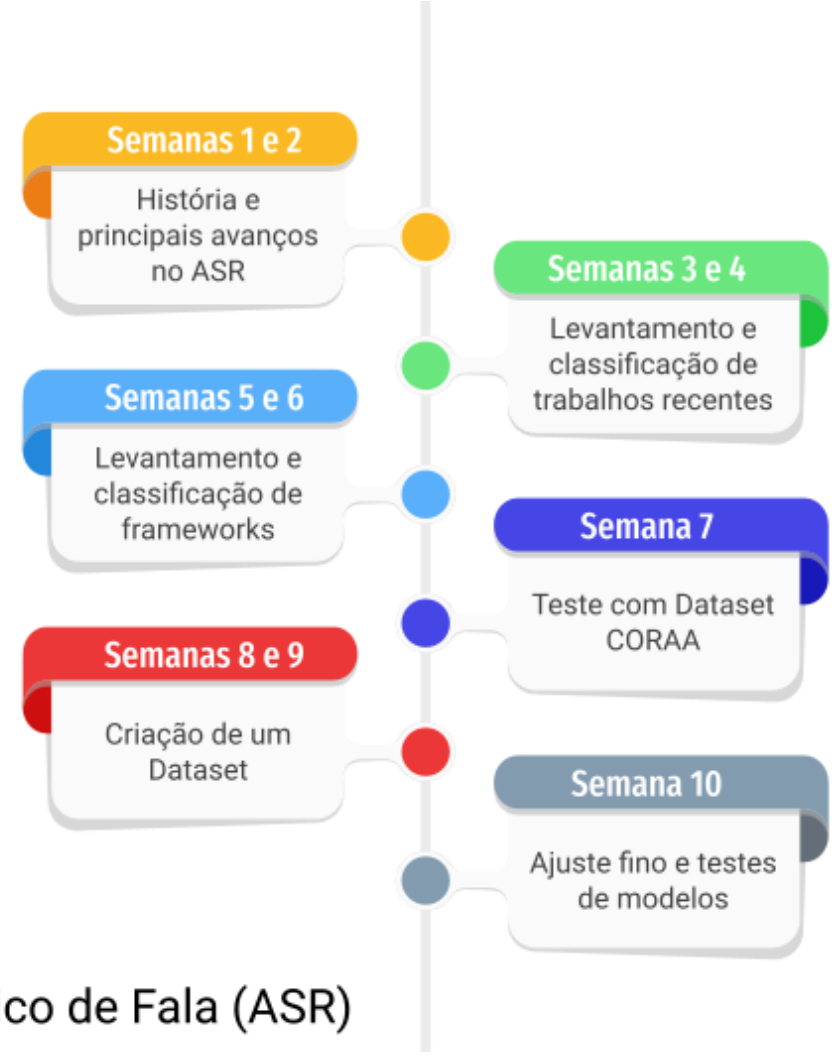
Keywords: artificial intelligence, large language models, automatic dataset generation.

Goiânia

2025

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Minha Jornada



Daniel Ribeiro da Silva
Especialista em: Reconhecimento Automático de Fala (ASR)

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

MINHA JORNADA

Nome: **Daniel Ribeiro da Silva**

Especialidade: **Reconhecimento Automático de Fala (ASR)**

Objetivo deste documento

Durante o processo da disciplina Residência em IA[1], foram gerados diversos resultados na construção da minha especialização. A cada semana, um conjunto de resultados foi formalizado por um Termo de Aceite de Entrega e avaliado por uma banca, considerando o planejado e o realizado para o período. Este documento tem como objetivo descrever esses resultados obtidos, fazendo referência aos Termos de Aceite de Entrega e seus documentos associados.

Minha Jornada

Minha jornada teve início na **Semana 1**, com atividades voltadas para a definição de uma área de conhecimento para minha especialização. Desde o princípio, a escolha por Processamento de Áudio e Voz se mostrou evidente. No entanto, foi necessário delimitar uma subárea que permitisse um desenvolvimento consistente ao longo das semanas. Para isso, dediquei-me ao estudo da história do campo e à análise de artigos de revisão que apresentassem uma visão abrangente sobre o tema. Após essa investigação, defini o Reconhecimento Automático de Fala (ASR) como minha área de especialização. Com essa escolha, finalizei a semana elaborando um resumo histórico e consolidando os artigos de revisão levantados, encerrando as atividades com uma base sólida para os próximos passos. Na **Semana 2**, realizei uma filtragem nos artigos previamente selecionados, escolhendo aprofundar-me no trabalho de Dhanjal e Singh[2]. Esse estudo apresenta um panorama sobre o Reconhecimento Automático de Fala no contexto das redes neurais

[1] Dez semanas, entre setembro de 2024 e dezembro de 2024.
[2] Dhanjal, A. S., Singh, W; A comprehensive survey on automatic speech recognition using neural networks; 2023.

artificiais. Para consolidar os principais pontos abordados no artigo, elaborei um resumo que auxiliou na compreensão dos conteúdos e na organização das ideias. Essa análise me proporcionou uma visão ampla do cenário atual na área de estudo, permitindo direcionar o foco para abordagens mais especializadas no campo de ASR. Os avanços realizados durante as duas primeiras semanas estão detalhados no **Apêndice 1**, registrando as etapas que fundamentam a continuidade do projeto.

Nas **Semanas 3 e 4**, o foco esteve no aprofundamento dos fundamentos e das técnicas atuais em Reconhecimento Automático de Fala. Para isso, foi realizado um levantamento dos estudos mais relevantes dos últimos anos, acompanhado de análises para compreender como as abordagens voltadas para o Português Brasileiro se situam nessa evolução temporal. Além disso, analisei a correlação entre autores de diferentes estudos, buscando identificar conexões e influências no campo. Como resultado dessa etapa, foram gerados três produtos principais: uma tabela com as principais informações sobre os artigos analisados, um gráfico que ilustra a evolução temporal dos estudos e um gráfico que visualiza a correlação entre os autores. Todo o desenvolvimento dessas duas semanas está detalhado no **Apêndice 2**, consolidando as bases para as próximas fases do trabalho.

Durante as **Semanas 5 e 6**, o foco esteve na identificação de ferramentas que possam auxiliar no desenvolvimento de aplicações para Reconhecimento Automático de Fala. Após uma análise criteriosa, foram selecionados 20 frameworks atualizados, que oferecem funcionalidades como manipulação de arquivos de áudio, pré-processamento, treinamento, ajuste fino e inferência com modelos pré-treinados. As ferramentas foram classificadas de acordo com suas características e organizadas em um texto descritivo, complementado por uma visualização gráfica que destaca seus principais aspectos. Todo o desenvolvimento relacionado aos frameworks está detalhado no **Apêndice 3**.

A partir do estudo de fundamentos, técnicas e ferramentas voltados para Reconhecimento Automático de Fala, foi possível, na **Semana 7**, iniciar testes práticos para a Residência em IA. O foco inicial foi analisar a influência da base de dados de treinamento

no desempenho de um modelo de ASR. Embora, em cenários generalistas, o Wav2vec 2.0[3] apresente acurácia inferior ao Whisper[4], o objetivo do teste foi realizar o ajuste fino de uma versão do Wav2vec 2.0 para o Português Brasileiro utilizando os dados do CORAA[5] e, a partir disso, comparar seu desempenho com o do Whisper. Após o ajuste fino, o Wav2vec 2.0 demonstrou desempenho superior ao Whisper, evidenciando a relevância de dados de treinamento adequados para a melhoria de modelos. Os testes realizados e os resultados obtidos estão detalhados no **Apêndice 4**.

Diante dos resultados promissores obtidos na **Semana 7**, o próximo passo foi avaliar como dados de treinamento específicos do domínio médico impactam o desempenho dos modelos. Assim, as **Semanas 8 e 9** foram dedicadas à construção de um conjunto de dados focado nesse cenário. Para isso, utilizei vídeos do YouTube com legendas anotadas, realizando a extração dos áudios e transcrições necessárias. Ao final da Residência em IA, a base de dados criada ultrapassou 17 horas de material e incluiu duas bases de teste distintas: uma composta por dados com os mesmos locutores da base de treino e outra por dados com novos locutores. As principais características do conjunto de dados, bem como os detalhes do processo de extração, estão descritos no **Apêndice 5**.

Na **Semana 10**, foram realizados os testes com o dataset criado nas **Semanas 8 e 9**. O modelo Wav2vec 2.0 passou por um ajuste fino, seguido pela avaliação de seu desempenho nas duas bases de teste. Os resultados mostraram que o contexto dos dados de treinamento tem um impacto significativo na qualidade das transcrições finais. O modelo ajustado apresentou melhorias em ambas as bases de teste, com destaque para o desempenho na base que continha os mesmos locutores dos dados de treinamento, onde superou os resultados obtidos pelo Whisper. A compilação completa dos resultados está disponível no **Apêndice 6**.

[3] Baevski, A. et al. Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations; 2020.

[4] Radford, A. et al. Robust Speech Recognition via Large-Scale Weak Supervision; 2022.

[5] Candido Junior, A. et al. CORAA: a large corpus of spontaneous and prepared speech manually validated for speech recognition in Brazilian Portuguese; 2021.

Por fim, considero que o desenvolvimento deste projeto ao longo das semanas proporcionou uma imersão no campo de Reconhecimento Automático de Fala (ASR), consolidando uma base teórica sólida e aprimorando habilidades práticas. As Semanas de estudo foram essenciais para meu desenvolvimento como especialista, permitindo uma compreensão aprofundada de técnicas e ferramentas, além de reforçar a importância do preparo de dados para o sucesso de modelos. Os testes práticos demonstraram a relevância do treinamento direcionado, com resultados que superaram expectativas e confirmaram o impacto de um trabalho bem fundamentado. Este percurso fortaleceu minha confiança e preparação para futuros desafios na área de Inteligência Artificial.

APÊNDICE 1

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 19 de set. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Daniel Ribeiro da Silva

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

- Foi gerado um resumo sobre alguns dos principais acontecimentos na área de processamento de áudio e voz, o resumo trata de assuntos com relevância histórica e que trazem uma noção temporal sobre o avanço das primeiras técnicas. Link do resumo: https://docs.google.com/document/d/1i5r2bfm5rHREw14PxeGbFQfTWsQg280p1uSZYTgWbC4/edit?usp=sharing
- Foram levantados artigos de revisão sobre processamento de áudio e voz, com certa preferência para ASR, os artigos foram selecionados a partir da leitura de seus títulos e resumos. Link para a seleção de artigos de revisão: https://docs.google.com/document/d/1-FC_kvrHVQxpIds0YdtTqTxkso7hRWrOvWSZtbF3Jv8/edit?usp=sharing

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Selecionar artigos para a leitura completa, com isso obter compreensão sobre o cenário atual do processamento de áudio e voz e ASR.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

Tenho dúvidas sobre como selecionar os artigos, se dou preferência para quantidade de citações, data de publicação, etc.

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

[Resumo sobre alguns dos principais acontecimentos na área de processamento de áudio e voz, citado no Termo de Aceite de Entrega de 19 de setembro de 2024]

História - Processamento de áudio e voz

Para realizar um estudo sobre a história do processamento de áudio e voz, este resumo estará dividido cronologicamente com os acontecimentos mais interessantes sobre a área. O objetivo aqui foi entender fatos mais primordiais dentro da área, não sendo um resumo técnico dos próximos campos de estudo durante o processo de residência.

1860 - Gravação de som mais antiga conhecida

Em 25 de março de 1857, Édouard-Léon Scott de Martinville, um bibliotecário francês, registrou a patente de um aparelho chamado fonoautógrafo. Juntamente com os documentos da patente, foram encontrados fononautogramas, gravações de som datadas de 1860, 28 anos antes de Thomas Edison desenvolver o fonógrafo [1].

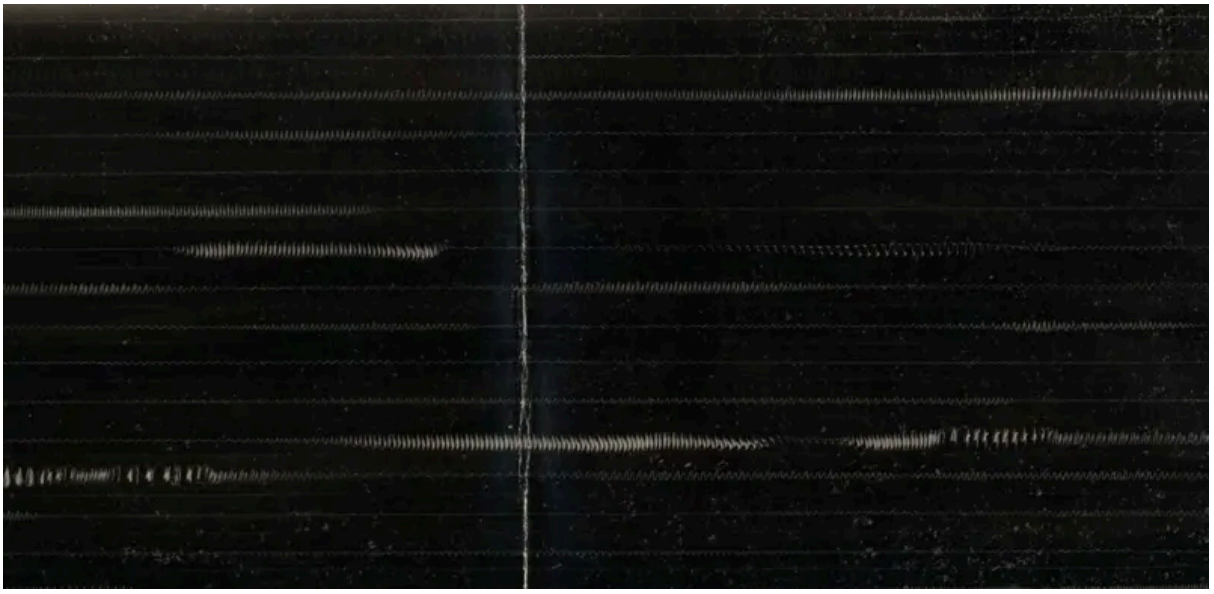


Figura 1: Marcas deixadas no papel fuliginoso pelo fonoautógrafo, retirada de [1].

1877 - Invenção do fonógrafo

Thomas Edison criou, em 1877, um aparelho capaz de gravar e reproduzir sons, o fonógrafo ganhou destaque nas décadas seguintes e é sempre lembrado como um marco notável no processamento de áudio [2].

1936 - PCM

O “Pulse Code Modulation” ou PCM é uma técnica que permite transformar um sinal analógico em digital. Sendo usado em computadores, Blu-ray, DVD e Discos Compactos, é um padrão para áudio digital.

1938 - Vocoder

O vocoder foi desenvolvido para fins militares, a fim de codificar e descodificar a voz humana e, assim, transmiti-la. Também foi e é utilizado por músicos para fins artísticos [3].

1965 - Transformada Rápida de Fourier

A Transformada Rápida de Fourier (FFT) é uma ferramenta central no processamento digital de áudio e, a partir da década de 60, a sua aplicação juntamente com o desenvolvimento dos computadores passou a ser frequente.

1969 - Linear Predictive Coding

Linear Predictive Coding (LPC) é uma técnica amplamente utilizada no processamento de sinais de áudio, principalmente na codificação e síntese de fala. A técnica foi introduzida na década de 1960 por Fumitada Itakura e Shuzo Saito, e avançada por Bishnu Atal e Manfred Schroeder. Nos anos 1970 e 1980, o LPC tornou-se base para tecnologias de compressão de voz.

1970-1980 - MP3

O MP3 foi desenvolvido ao longo das décadas de 1970 e 1980 por pesquisadores alemães, como Karlheinz Brandenburg, que aplicaram princípios de psicoacústica para criar um algoritmo de compressão de áudio altamente eficiente [4]. O MP3 ficou famoso a partir dos anos 90, com um aparelho que poderia ser usado como reprodutor de músicas.

1995 - Modelos Ocultos de Markov (HMM)

HMM (Hidden Markov Model) é um modelo probabilístico usado para representar sequências temporais. No contexto de reconhecimento de fala, por exemplo, o HMM modela a sequência de fonemas ou palavras com base em características extraídas do áudio [5].

2000 em diante - Redes neurais

Dentre as várias técnicas que surgiram nesse período, estão CNNs, RNNs, LSTMs, DNNs, DBNs, Hybrid networks, End-to-end recognition system, Transfer learning [6], dentre outros. Todos esses tópicos serão estudados de maneira mais detalhada nas próximas semanas.

Referências Bibliográficas

[1]: Dalia Ventura. O mistério das gravações de voz humana feitas 3 décadas antes de Thomas Edison. Disponível em: <https://www.bbc.com/portuguese/geral-56403105>. Acesso em 18 set. 24.

[2]: Soraia Simões de Andrade. Fonógrafo. Disponível em: <https://www.muralsonoro.com/mural-sonoro-pt/2014/2/15/fongrafo>. Acesso em 18 set. 24.

[3]: Eloy Caudet, Vocoders: Funcionalidade, história e melhores modelos. Disponível em: <https://woodandfirestudio.com/pt/vocoder/>. Acesso em 18 set. 24.

[4]: Nilton Kleina. A história do MP3, o formato que espalhou a música digital. Disponível em: <https://www.tecmundo.com.br/mercado/139452-historia-mp3-formato-espalhou-musica-digital-video.htm>. Acesso em 18 set. 24.

[5]: Modelos Ocultos de Markov. Disponível em: <http://leg.ufpr.br/~lucambio/MOM/MOM.html>. Acesso em 18 set. 24.

[6]: Amandeep Singh Dhanjal e Williamjeet Singh. A comprehensive survey on automatic speech recognition using neural networks. Disponível em: <https://link.springer.com/article/10.1007/s11042-023-16438-y?fromPaywallRec=true>. Acesso em 18 set. 24.

[Artigos de revisão sobre processamento de áudio e voz, com certa preferência para ASR, citado no Termo de Aceite de Entrega de 19 de setembro de 2024]

Referências encontradas:

Após uma pesquisa, foram selecionadas algumas referências que pareceram interessantes a princípio, as pesquisas se voltaram a artigos de revisão sobre processamento de áudio. Para selecionar os exemplos abaixo, foi realizada a leitura do título e do resumo de cada artigo, além de quais tópicos eram destacados nos mesmos. A intenção é realizar mais uma filtragem nesses artigos para selecionar os que serão lidos por completo, há a possibilidade de adição de novos artigos posteriormente.

[História][Fundamentos] Audio Signal Classification: History and Current Techniques (2003): https://www.uregina.ca/science/cs/assets/docs/techreports/2003-07.pdf

- Parece ter bastante coisa, não só classificação de áudio, bom para ver a história e ferramentas base para processamento de sinais.

[História][Fundamentos] A Review of Feature Extraction and Classification Techniques in Speech Recognition (2023): https://link.springer.com/article/10.1007/s42979-023-02158-5?fromPaywallRec=true

- Fala sobre algumas bases de processamento de áudio e ASR.

[História][Fundamentos] Speech Recognition by Machine: A Review (2009): https://arxiv.org/pdf/1001.2267

- Apresenta um resumo bem amplo sobre a história de ASR e alguns fundamentos.

[Fundamentos] Automatic Speech Recognition: A Review (2012): https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=aac912cbdb4eddc2ac5a62c0d8938ec2f5a7dc6b

- Focado em apresentar uma revisão dos fundamentos de ASR.

[Fundamentos] Audio Signal Processing: A Review of Audio Signal Classification Features (2017): https://www.researchgate.net/profile/Mittal-Darji/publication/318224324_Audio_Signal_Processing_A_Review_of_Audio_Signal_Classification_Features/links/595dd5ba0f7e9b3aefb62a45/Audio-Signal-Processing-A-Review-of-Audio-Signal-Classification-Features.pdf

- Revisão de alguns fundamentos para processamento de áudio, abordagens clássicas de processamento de sinais.

[Fundamentos] Computational Intelligence in Speech and Audio Processing: Recent Advances (2010):
https://link.springer.com/chapter/10.1007/978-3-642-11282-9_32

- Fala sobre algumas áreas dentro de processamento de voz e sobre avanços até 2010.

[Fundamentos] Audio Self-supervised Learning: A Survey (2022):
https://arxiv.org/pdf/2203.01205

- Focado em modelos SSL, faz um resumo da área.

[Fundamentos][Frameworks] A comprehensive survey on automatic speech recognition using neural networks (2023):
https://link.springer.com/article/10.1007/s11042-023-16438-y?fromPaywallRec=true
PDF: file:///home/daniel/Downloads/s11042-023-16438-y.pdf

- Parece muito bom, fala de quase tudo, técnicas, modelos, datasets, focado em ASR.

[Fundamentos][Frameworks] A Comprehensive Analysis of Speech Recognition Systems in Healthcare: Current Research Challenges and Future Prospects (2024):
https://link.springer.com/article/10.1007/s42979-023-02466-w?fromPaywallRec=false

- Focado na área médica, faz um review de ASR.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 25 de set. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Daniel Ribeiro da Silva

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

O objetivo para essa semana era:

- Dentre os artigos de revisão já selecionados, realizar mais um filtro e selecionar artigos para leitura completa.

Para isso:

- Foi selecionado o artigo intitulado "A comprehensive survey on automatic speech recognition using neural networks", disponível no link: https://drive.google.com/file/d/1EgP57tfMNqZGpS9FMGGG2xBO-HBFc-ts/view?usp=sharing
- Foi escrito um resumo sobre o artigo a fim de ter um compilado das principais ideias do artigo, resumo disponível no link: https://docs.google.com/document/d/1rVvtz-wIvBrF9VvFTIA7tEd2AMoUBE4JKgNVM1Qm_rA/edit?usp=sharing

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

O artigo selecionado para leitura foi publicado em 2023, escrito em 2022 e faz uma revisão de artigos até 2021, mesmo realizando citações sobre trabalhos mais recentes, existe uma lacuna de tempo importante o qual as referências lidas até o momento não contemplam. Para a próxima semana é necessária uma revisão de técnicas mais recentes no campo de ASR.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

LEONARDO ALVES: Go!

[Resumo sobre o artigo “A comprehensive survey on automatic speech recognition using neural networks”, citado no Termo de Aceite de Entrega de 25 de setembro de 2024]

A comprehensive survey on automatic speech recognition using neural networks

Abstract

O artigo mostra um panorama do estado da arte em ASR de 2015 a 2021, evidenciando os principais avanços na área e promovendo um conhecimento sobre os estudos mais recentes.

1. Introduction

Fala sobre a importância de sistemas de ASR para melhorar a vida das pessoas. Explica a forma que é produzida a fala humana e que algoritmos computacionais tentam capturar as informações de forma parecida com os humanos. Finaliza dizendo sobre a complexidade de se trabalhar com a voz e como o artigo deve apresentar as técnicas de deep learning, os datasets e as principais abordagens.

1.1. Motivation

Fala sobre a importância de se estudar os métodos de aprendizado de máquina voltados para o processamento de fala, principalmente deep learning. Explica um pouco sobre a pesquisa de artigos entre os anos de 2015 a 2021 que será detalhada nos próximos tópicos.

1.2. Our contribution

Explica como está estruturado o artigo e diz as vantagens de se abordar quatro tópicos principais: técnicas estado da arte, datasets, toolkits e métricas voltados para ASR. Cita artigos de revisão que não abordam esses tópicos por completo.

2. Research Method

Explica brevemente que foi realizado um trabalho de busca nos seguintes repositórios: Science Direct, Willey, Taylor and Francis, Elsevier, Springer, ACM, IEEE.

2.1. Research Question

Foram definidas algumas perguntas para apoio a pesquisa, como por exemplo: Quais métricas são geralmente aplicadas em reconhecimento de fala? ou Quais são os prós e os contras mais comuns usados na técnica de feature extraction?

2.2. Search strategy

Essa seção visa mostrar como foi a estratégia de busca nos portais, para pesquisar artigos de interesse foi necessário a construção de sentenças de buscas que continham palavras como ASR, Deep Learning, speech recognition ou Neural Network, por exemplo.

2.3. Study selection

Alguns resultados da pesquisa foram excluídos, já que apenas as sentenças não foram capazes de filtrar os artigos de interesse, então foi feita uma seleção pelos títulos dos artigos para verificar estudos que falavam sobre speech recognition, em especial com deep learning associado.

2.4. Study quality assessments

Mais alguns filtros foram aplicados nessa etapa, publicações pequenas e artigos que forneciam um resultado satisfatório foram retirados.

2.5. Data Extraction

Os artigos selecionados foram divididos em algumas categorias: study information, techniques, tools, datasets, evaluation metrics e quantitative results, cada qual com suas especificidades para selecionar os estudos para posterior análise.

2.6. Data synthesis and reporting the review

Os artigos foram divididos e analisados, a partir dos principais pontos abordados nos estudos foram definidos os tópicos que se seguem.

3. Preliminary

Essa seção vai mostrar uma visão geral de como funciona o processo de reconhecimento de voz utilizado atualmente e, após, passará pelas principais técnicas estado da arte na área de Speech Recognition.

3.1. Speech Recognition Process

Explica brevemente o reconhecimento de fala. Uma pessoa usa suas cordas vocais para produzir a fala, um microfone realizará a digitalização do sinal. Após ter a representação do sinal, podemos utilizar técnicas de extração de características para gerar representações do

conteúdo da fala, como MFCC, LPC, PLP e DWT. Depois podemos utilizar técnicas para decodificar essas características para o objetivo que desejamos, como reconhecimento de falante, identificação de sentimentos e tradução da fala, algumas técnicas serão mostradas nas próximas seções, mas podemos destacar Acoustic Models e Language Models como partes essenciais nesse processo.

3.2. Techniques

Resume brevemente como funciona a lógica de deep learning, como camadas de redes neurais interligadas pelos neurônios de cada camada, além de como o deep learning pode atuar em quase todos os processos após a amostragem do sinal.

3.2.1. Convolution neural network (CNN)

A seção traz um resumo geral sobre as arquiteturas CNN. As camadas convolucionais tem por objetivo extrair características de imagens através da operação de convolução. As camadas de pooling tem por objetivo diminuir o tamanho da imagem mantendo suas características principais, existem diversos tipos de pooling, como Max Pooling, Average Pooling, Min Polling, etc. A camada fully-connected tem por objetivo receber as representações das camadas anteriores e gerar a saída do modelo de acordo com a tarefa proposta. A rede também é composta por uma função de ativação e camadas de dropout, a função de ativação vai dizer quais tipos de informações serão passadas de camada em camada, já o dropout se refere a exclusão aleatória de neurônios durante o treinamento a fim de evitar overfitting.

3.2.2. Recurrent neural network (RNN)

As RNNs se tornaram famosas por aceitarem entradas de dados sequenciais para saídas sequenciais, muito utilizadas em diversos problemas de processamento de linguagem natural. Porém, esse método ainda sofre com problemas como vanishing (quando o gradiente da rede decresce muito) e explosão de gradiente (quando o gradiente da rede cresce muito).

3.2.3. Long Short-Term Memory (LSTM)

LSTM é uma variação de RNN. A rede busca resolver os problemas anteriores, ela implementa uma lógica com três dependências principais a fim de “armazenar” informações por longos períodos de tempo. A arquitetura já foi muito usada no processamento de linguagem natural.

3.2.4. Deep neural network (DNN)

Uma DNN, ou Rede Neural Profunda, caracteriza-se por ser uma rede capaz de crescer exponencialmente em tamanho, já que suas camadas podem ser enfileiradas indefinidamente, a depender do poder computacional disponível. A rede é composta basicamente por camada de entrada, camadas ocultas e camada de saída. Geralmente, quanto maior a rede, maior a acurácia. Porém, quanto mais camadas, mais complexo é a otimização da estrutura, chegando a um ponto que o crescimento da rede não apresenta melhora.

3.2.5. Deep belief network (DBN)

DBN pode ser considerada uma variação de DNN, ela utiliza de um aprendizado não supervisionado para aprender sobre a representação dos dados, posteriormente existe a necessidade de um ajuste fino supervisionado para a tarefa específica.

3.2.6. Hybrid networks

Hybrid networks se refere basicamente a uma abordagem que consiste em juntar mais de uma rede neural para gerar o resultado. Algumas junções em específico geram resultados satisfatórios em problemas como English speech recognition.

3.2.7. End-to-end recognition system

End-to-end recognition system se refere a uma abordagem que não necessita de processos intermediários entre a captura do sinal de áudio e sua respectiva transcrição. O artigo cita duas abordagens principais, a primeira baseada em atenção e a segunda baseada em programação dinâmica.

3.2.8. Transfer learning

Transfer learning se refere a uma técnica voltada para modelos não precisarem de treinamento inteiro de sua rede para novas tarefas, uma vez que o aprendizado pode ser transferido. O artigo cita aplicações em modelos multilinguais, uma vez que um algoritmo em uma determinada língua não precisa ser treinado novamente em outro idioma, apenas ajustado.

4. Analysis

Essa seção visa mostrar os trabalhos estado da arte (até 2021) em ASR, em específico mostrando as técnicas, toolkits, datasets e métricas mais utilizadas.

4.1. ASR Techniques

Essa seção inicia com comentários sobre o início dos métodos de ASR, em especial 1952 com o reconhecimento de dígitos falados, estudo de três pesquisadores do Bell Lab.

O artigo sumariza os estudos abordados na revisão em tabelas divididas em CNN, RNN, DNN, Hybrid, Transfer learning e outros. Essas tabelas podem ser consultadas futuramente para análise de estudos focados em áreas específicas.

O artigo resume em texto corrido um pouco sobre as técnicas usadas no âmbito de ASR.

CNNs não são exclusivamente para problemas de visão computacional, uma vez que é possível trabalhar com a geração de espectrogramas dos sinais de áudio e posterior treinamento de arquiteturas CNN. É possível o uso de diferentes métodos de pooling, com destaque para Max Pooling, Stochastic Pooling e Average Pooling, diferentes estudos trazem resultados diversos sobre o melhor uso dos métodos. Dois destaques em artigos sobre CNNs são a função de ativação ReLU e Dropout, ambos melhoram significativamente a performance dos modelos.

RNNs tiveram um desempenho considerável em tarefas de extração de features, já sua junção com LSTM possibilita bons resultados em classificações de áudio. Abordagens com GRU se mostraram melhores que LSTM para tarefas de ASR diversas.

HMMs são importantes em uma série de aplicações em áudio, principalmente no momento de decodificar a representação gerada por modelos. VAD é uma área extremamente importante em ASR, já que ela possui ligação com diversas outras e sua aplicação melhora as características de entrada para modelos maiores.

Modelos baseados em CTC obtiveram resultados melhores em muitas aplicações, porém possuem um treinamento bem mais custoso em relação aos outros métodos.

Aplicações de Transfer learning, end-to-end recognition system e redes híbridas também se mostraram eficientes em uma série de estudos. No geral, os métodos aqui descritos conseguem performar bem em cenários controlados, porém existem dificuldades em cenários estocásticos, ruidosos e com baixa qualidade de áudio.

4.2. ASR tools

Toolkits são coleções de trabalhos voltados para treinar e construir sistemas, nesse caso voltados para ASR. O artigo cita alguns casos:

- HTK: Voltado para HMMs, parece não ter atualizações desde 2016.
- ESPnet: Possui uma série de aplicações voltadas para speech recognition, text-to-speech, speech translation, speech enhancement, speaker diarization, spoken language understanding, etc. Mantêm-se atualizado.
- Kaldi: Possui aplicações com HMM, GMM e SGMMs implementadas. Mantêm-se atualizado.

- DeepSpeech: (link não informado no artigo) Possui algumas implementações voltadas para STT. Sem atualizações há 3 anos.
- wav2vec: No momento de análise do artigo, o wav2vec havia sido lançado a pouco tempo, foi lançado como um modelo capaz de aprender representações de diferentes línguas com poucos minutos de áudio.
- Nabu: Baseado em Tensorflow, oferece recursos de ASR. Não foram encontradas referências sobre as últimas atualizações.
- Espresso: Baseado no Pytorch e no Fairseq, promove algumas implementações em ASR. Mantêm-se atualizado.
- Athena: (link não informado no artigo) Possui aplicações para Automatic Speech Recognition, Speech Synthesis, Voice activity detection, Wake Word Spotting, etc. Sem atualizações há 2 anos.

4.3. ASR Datasets

Os dados são essenciais para qualquer tarefa de ML, o artigo reúne uma coletânea de datasets usados com frequência nos estudos entre 2015 e 2021:

- VoxForge: Dataset voltado para diferentes sotaques. Línguas: Italiano, Alemão, Francês, Espanhol e Português.
- TED-LIUM: áudios de “TED’s audio talks”.
- TIMIT: áudios com sentenças pré-definidas de norte-americanos.
- LibriSpeech: áudio livros em inglês.
- WSJ: áudios do “Wall Street Journal”.
- Common Voice: Projeto da Mozilla, qualquer pessoa pode gravar uma sentença pré-definida na sua língua e disponibilizar no dataset. Contém mais de 50 línguas

O artigo oferece uma tabela com outros 20 datasets, pode ser revisitada posteriormente para verificação de possíveis usos.

4.4. ASR evaluation metrics

Essa seção reúne as principais métricas de análise de resultados em ASR encontradas nos artigos selecionados:

- Observação: Variáveis contidas nas fórmulas abaixo (a depender da fórmula, ela considera número de caracteres, palavras, etc):
  - S: número de substituições;
  - D: número de deleções;
  - I: número de inserções;
  - C: número de correções;
  - E: número de sentenças não reconhecidas;
  - N: número total.

- WER (Word Error Rate): WER=(S+D+I)/N. Verifica o erro em relação às palavras;
- WRR (Word Recognition Rate): WRR=1-WER. Variação do WER;
- CER (Character Error Rate): CER=(I+D+S)/(C+D+S). Verifica o erro em relação aos caracteres;
- SER (Sentence Error Rate): SER=E/R. Verifica o erro em relação às sentenças;
- PER (Phone Error Rate): PER=(S+D+I)/N. Verifica o erro em relação aos fonemas;
- EER (Equal Error Rate): EER=(FRR+FAR) -> FAR=(NumberoffalseAcceptance/TotalnumberofimposterAcceptenceNumber)*100 -> FRR=(Numberoffalserejection/TotalnumberofgenuineAttempts)*100. Comum em tarefas de Reconhecimento de falante.
- Outras métricas usadas são FER (Frame Error Rate), MAE (Mean Absolute Error), RMSE (Root Mean Squared Error) e Matriz de Confusão.

5. Discussion

Nessa seção são respondidas as perguntas levantadas na seção 2.1. No geral, são discutidos os resultados de forma quantitativa, como por exemplo: o WER foi a métrica mais usada nos estudos, presente em 45% deles, as estatísticas que considero mais importantes e algumas observações serão colocadas abaixo:

- WER é a principal métrica;
- Kaldi foi o toolkit mais usado;
- São enumerados repositórios no github com mais de 2000 estrelas que não foram citados anteriormente, eles estão descritos na tabela 12, pode ser verificada posteriormente a depender dos objetivos que serão traçados;
- Praticamente todas as técnicas citadas ao longo do artigo foram disruptivas no momento de suas primeiras análises, todas mostraram resultados promissores nos estudos e o artigo cita um panorama de como foi esse avanço;
- A figura 7 do artigo mostra as palavras mais comuns nos estudos sobre ASR;
- A figura 8 do artigo mostra os autores com mais publicações em ASR;
- A tabela 13 mostra os prós e contras das abordagens de feature extraction;
- O artigo cita que quantidade de ruído é muito prejudicial para as tarefas de ASR;
- Variedade de línguas nos datasets são bem escassas;
- Processamento em tempo real é difícil e existem poucas abordagens;
- Fala espontânea possui dificuldade maior de análise de modelos, muito por conta de sotaque, pronúncias erradas, etc.
- São necessários datasets grandes para obtenção de bons resultados;

6. Conclusion

Existem uma gama de aplicações para ASR, já existem diversas ferramentas extremamente eficazes em alguns cenários. Porém, ainda existem contextos com desafios a serem explorados.

APÊNDICE 2

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 2 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Daniel Ribeiro da Silva

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

O objetivo para essa semana era:

- Realizar uma pesquisa sobre as técnicas mais usadas nos últimos anos dentro do campo de ASR.

A pesquisa foi realizada e buscou referências para as técnicas e modelos mais comuns em implementações atuais no campo de ASR, também houve um foco em abordagens que utilizam português como campo de estudo, como resultado da pesquisa, foi feita a criação da seguinte tabela:

- Principais técnicas usadas nos últimos anos no campo de ASR

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Definir um escopo para uso das técnicas aprendidas de ASR dentro da Residência.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

[Tabela “Artigos Gerais”, dentro de “Principais técnicas usadas nos últimos anos no campo de ASR”, citado no Termo de Aceite de Entrega de 2 de outubro de 2024]

Artigo	Link	Ano	Citações	Técnicas	Observações	Busca
Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition	link	2020	362	Wav2vec 2.0, Conformer, aumento de dados com ruído	Bateu alguns benchmarks	arxiv
W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training	link	2021	380	W2v-BERT	-	arxiv
Conformer: Convolution-augmented Transformer for Speech Recognition	link	2020	3111	Conformer	-	arxiv
Fast Conformer with Linearly Scalable Attention for Efficient Speech Recognition	link	2023	51	parakeet-rnnt-1.1b (Fast Conformer)	-	arxiv
Self-Training for End-to-End Speech Recognition	link	2020	253	Treinamento auto-supervisionado	-	arxiv
Self-training and Pre-training are Complementary for Speech Recognition	link	2020	185	Conv + Transformer + wav2vec2.0 + pseudo labeling	-	arxiv
Wav2vec: Unsupervised Pre-Training For Speech Recognition	link	2019	1565	Wav2vec	-	arxiv
Vq-Wav2vec: Self-Supervised Learning Of Discrete Speech Representations	link	2020	720	Vq-Wav2vec	-	arxiv
wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations	link	2020	5412	Wav2vec 2.0	-	arxiv

HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units	link	2021	2495	HuBERT	-	arxiv
ContentVec: An Improved Self-Supervised Speech Representation by Disentangling Speakers	link	2022	106	ContentVec	-	arxiv
WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing	link	2022	1490	WavLM	-	arxiv
ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context	link	2020	305	ContextNet	-	arxiv
Deep Speech: Scaling up end-to-end speech recognition	link	2014	2695	Deep Speech	-	arxiv
Deep Speech 2: End-to-End Speech Recognition in English and Mandarin	link	2015	3773	Deep Speech 2	-	arxiv
Wav2Letter: an End-to-End ConvNet-based Speech Recognition System	link	2016	362	Wav2Letter	-	arxiv
Jasper: An End-to-End Convolutional Neural Acoustic Model	link	2019	284	Jasper	-	arxiv
Listen, Attend and Spell	link	2015	605	LAS	-	arxiv
Robust Speech Recognition via Large-Scale Weak Supervision	link	2022	2772	Whisper	-	arxiv
XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale	link	2021	603	XLS-R (baseado no wav2vec 2.0)	-	arxiv

[Tabela “Artigos focados em português”, dentro de “Principais técnicas usadas nos últimos anos no campo de ASR”, citado no Termo de Aceite de Entrega de 2 de outubro de 2024]

Artigo	Link	Ano	Citações	Técnicas	Observações	Busca
A survey on automatic speech recognition systems for Portuguese language and its variations	link	2020	52	Artigo de revisão	-	Google Acadêmico
Brazilian Portuguese Speech Recognition Using Wav2vec 2.0	link	2022	15	Wav2vec 2.0	-	Google Acadêmico (arxiv)
Desenvolvimento de um modelo de reconhecimento de voz para o Portugues Brasileiro com poucos dados utilizando o Wav2vec 2.0	link	2021	5	Wav2vec 2.0	-	Repositório USP
An open-source end-to-end ASR system forBrazilian Portuguese using DNNs built from newlyassembled corpora	link	2020	19	DeepSpeech 2, languange model	Trabalho interessante	Referências
CORAA: a large corpus of spontaneous and prepared speech manually validated for speech recognition in Brazilian Portuguese	link	2021	14	Dataset, Wav2vec, Wav2vec 2.0	-	Google Acadêmico
A Data-Centric Approach for Portuguese Speech Recognition: Language Model And Its Implications	link	2023	0	Wav2vec 2.0, language model	-	ieeexplore

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 9 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Daniel Ribeiro da Silva

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Com a finalidade de finalizar os estudos a respeito de História, Fundamentos e Técnicas, foi gerada algumas formas de visualizar todas as principais informações obtidas nas últimas semanas:

- **Tabela:** Tabela base para reunião de técnicas, fundamentos, frameworks e datasets
    - 33 artigos com as principais técnicas e fundamentos desenvolvidos nos últimos anos;
    - 6 artigos voltados para o Português Brasileiro;
    - 10 frameworks já mapeados;
    - 4 datasets usados como benchmarks comuns nos artigos mapeados.
- **Linha do tempo:** Canva, PDF
    - Oferece uma visualização temporal da evolução das técnicas reunidas nas últimas semanas, destaque para como as técnicas para o Português Brasileiro se encaixam em meio aos outros estudos.
- **Correlação entre autores:** Research Rabbit
    - Oferece uma visualização em relação a correlação de autores nos trabalhos desenvolvidos, trabalhos internacionais possuem ampla correlação, já os trabalhos para o Português Brasileiro possuem autores menos correlacionados com os demais.

Com a reunião das principais técnicas e fundamentos e o desenvolvimento de visualizações em linha temporal e correlação entre autores, foi possível fechar uma grande base teórica e entender como os estudos se relacionam.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Os frameworks já selecionados oferecem muitas abordagens interessantes, porém ainda não abordam processamentos mais simples para áudio e nem todos estão atualizados. Então, o

planejamento para a semana é continuar os estudos a respeito dos principais frameworks voltados a ASR e iniciar testes práticos para o entendimento aprofundado de cada um.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

[Tabela “Artigos Gerais”, dentro de “Tabela base para reunião de técnicas, fundamentos, frameworks e datasets”, citado no Termo de Aceite de Entrega de 9 de outubro de 2024]

Artigo	Link	Ano	Citações	Técnicas	Observações
Towards Robust Speech Representation Learning for Thousands of Languages	link	2024	4	XEUS	Auto-supervisionado, 4057 idiomas, treinado em mais de 1 milhão de horas de áudio.
Less is More: Accurate Speech Recognition & Translation without Web-Scale Data	link	2024	5	Canary	Baseado em Fast Conformer, treinamento com poucas horas de áudio (86 mil)
Speech Slytherin: Examining the Performance and Efficiency of Mamba for Speech Separation, Recognition, and Synthesis	link	2024	2	ConMamba	Mais eficiência em velocidade e memória para discursos longos.
Fast Conformer with Linearly Scalable Attention for Efficient Speech Recognition	link	2023	51	parakeet-rnnt-1.1b (Fast Conformer)	2,8 vezes mais rápido que o Conformer, supera o Conformer em velocidade e precisão.
Distil-Whisper: Robust Knowledge Distillation via Large-Scale Pseudo Labelling	link	2023	23	Distil-Whisper	Reduz o tamanho do Whisper, destilação de conhecimento, 5,8 vezes mais rápido que o Whisper e com redução de 51% dos parâmetros.

ContentVec: An Improved Self-Supervised Speech Representation by Disentangling Speakers	link	2022	106	ContentVec	Baseado no framework do Hubert, mecanismo de separação de locutor.
WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing	link	2022	1490	WavLM	Produzido para diversas tarefas, como diarização, ASR, verificação de falantes, etc. Robusto a ruído.
Robust Speech Recognition via Large-Scale Weak Supervision	link	2022	2772	Whisper	Modelo robusto, treinado em 680 mil horas de áudio, Transformer, robusto fora do domínio do treinamento, pouco processamento nos dados.
HuBERT-EE: Early Exiting HuBERT for Efficient Speech Recognition	link	2022	10	HuBERT-EE	Melhora em relação ao Hubert. Reduz a latência, mantêm um equilíbrio entre desempenho e velocidade.
W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training	link	2021	380	W2v-BERT	Aprendizado contrastivo, masked language modeling, CNNs, Transformer. É composto por um codificador de características, um módulo de aprendizado contrastivo e um módulo de MLM.
HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units	link	2021	2495	HuBERT	Entropia cruzada, targets são construídos por meio de clusterização,
XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale	link	2021	603	XLS-R (baseado no wav2vec 2.0)	Pré-treinado em 436 mil horas de dados multilinguais, desempenho bom em speech translation.

Robust wav2vec 2.0: Analyzing Domain Shift in Self-Supervised Pre-Training	link	2021	244	Wav2vec 2.0	Faz análises no Wav2vec 2.0, o domínio dos dados de pré-treinamento é bastante impactante.
Audio Albert: A Lite Bert for Self-Supervised Learning of Audio Representation	link	2021	91	Audio Albert	Foca na eficiência do modelo, uma vez que diminui em 91% a quantidade de parâmetros em relação a modelos mais robustos.
Performance-Efficiency Trade-offs in Unsupervised Pre-training for Speech Recognition	link	2021	40	SEW	Implementação reduzida do Wav2vec 2.0, rede de contexto menor, computação distribuída em diferentes componentes.
Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition	link	2020	362	Wav2vec 2.0, Conformer, aumento de dados com ruído	Utilização do encoder do Wav2vec 2.0, do conformer em versões XL e XXL, aumento de dados com SpecAugment e treinamento "Noisy Student". Treinado e testado no conjunto de dados do Librispeech.
Improved Noisy Student Training for Automatic Speech Recognition	link	2020	265	Noisy Student	Originalmente eficaz para classificação de imagens, envolve auto-treinamento iterativo, onde modelos sucessivos aprendem a partir de dados não rotulados. O paper traz uma perspectiva voltada à ASR, com aumentos de dados específicos.
Conformer: Convolution-augmented Transformer for Speech Recognition	link	2020	3111	Conformer	CNNs e Transformers. Possui algumas versões de modelos com tamanhos diferentes.
Self-Training for End-to-End Speech Recognition	link	2020	253	Treinamento auto-supervisi	Estuda o uso de treinamento auto-supervisionado em tarefas de ASR e fala em aumento significativo de precisão em relação a modelos treinados apenas com dados rotulados.

				onado	
Self-training and Pre-training are Complementary for Speech Recognition	link	2020	185	Conv + Transformer + wav2vec 2.0 + pseudo labeling	Estuda o uso de treinamento auto-supervisionado e não supervisionado em tarefas de ASR. Afirmar que a combinação dessas duas abordagens melhora significativamente o desempenho dos modelos.
Vq-Wav2vec: Self-Supervised Learning Of Discrete Speech Representations	link	2020	720	Vq-Wav2vec	Adiciona um método de quantização na arquitetura do Wav2vec, onde é gerado uma representação discretizada do áudio, útil para predizer o contexto.
wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations	link	2020	5412	Wav2vec 2.0	CNNs, Transformers, aprendizado contrastivo, método de quantização nas saídas do encoder. Demonstrou a viabilidade de pré-treinamento auto-supervisionado com dados não rotulados e posterior fine-tune em dados rotulados.
ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context	link	2020	305	ContextNet	CNNs com "squeeze-and-excitation", função de ativação Swish.
Transformer Transducer: A Streamable Speech Recognition Model with Transformer Encoders and RNN-T Loss	link	2020	527	Transformer Transducer	Codificadores Transformers, streaming, RNN-T Loss.
Wav2vec: Unsupervised Pre-Training For Speech Recognition	link	2019	1565	Wav2vec	Pré-treinamento não supervisionado, CNNs, classificação binária contrastiva.

Jasper: An End-to-End Convolutional Neural Acoustic Model	link	2019	284	Jasper	CNNs, otimizador NovoGrad, "simples e eficaz".
SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition	link	2019	2046	Aumento de dados	Técnica de aumento de dados, envolve distorção temporal, mascaramento de frequências e mascaramento de tempo. Melhorou os resultados sem a utilização da técnica.
wav2letter++: A Fast Open-source Speech Recognition System	link	2018	139	Wav2Letter em C++	Implementação do wav2letter em C++, paper não disponível para leitura completa.
Wav2Letter: an End-to-End ConvNet-based Speech Recognition System	link	2017	362	Wav2Letter	CNNs, MFCC, treinado e testado no conjunto Librispeech.
Listen, Attend and Spell	link	2016	605	LAS	Possui dois componentes principais: um listener e um speller. O listener é uma rede recorrente piramidal que processa sinais acústicos, enquanto o speller é uma rede baseada em atenção que gera caracteres de saída.
Deep Speech 2: End-to-End Speech Recognition in English and Mandarin	link	2015	3773	Deep Speech 2	RNNs e CNNs, loss CTC, 11.000 horas de áudio em inglês e 9.000 horas em mandarim.
Deep Speech: Scaling up end-to-end speech recognition	link	2014	2695	Deep Speech	RNNs, language model na saída, 5000 horas de dados de treinamento com 9600 locutores.

[Tabela “Artigos focados em português”, dentro de “Tabela base para reunião de técnicas, fundamentos, frameworks e datasets”, citado no Termo de Aceite de Entrega de 9 de outubro de 2024]

Artigo	Link	Ano	Citações	Técnicas	Observações
A Data-Centric Approach for Portuguese Speech Recognition: Language Model And Its Implications	link	2023	0	Wav2vec 2.0, language model	Estuda o impacto dos dados no treinamento de modelos de linguagem usados em conjunto com modelos de ASR.
Brazilian Portuguese Speech Recognition Using Wav2vec 2.0	link	2022	15	Wav2vec 2.0	Treinamento do Wav2vec 2.0 em um conjunto de dados de 470 horas de fala em português.
Desenvolvimento de um modelo de reconhecimento de voz para o Portugues Brasileiro com poucos dados utilizando o Wav2vec 2.0	link	2021	5	Wav2vec 2.0	Fine-tune do Wav2vec 2.0 com 1 hora de fala em português brasileiro.
CORAA: a large corpus of spontaneous and prepared speech manually validated for speech recognition in Brazilian Portuguese	link	2021	14	Dataset, Wav2vec 2.0	Apresentação do dataset CORAA, possui fala espontânea e preparada. Foi realizado o fine-tune do Wav2vec 2.0 com os dados reunidos e obteve-se resultados promissores.
A survey on automatic speech recognition systems for Portuguese	link	2020	52	Artigo de revisão	Aborda técnicas mais antigas, como HMM, MLP, DNN, SVM e LSTM.

language and its variations					
An open-source end-to-end ASR system forBrazilian Portuguese using DNNs built from newlyassembled corpora	link	2020	19	DeepSpeech 2, language model	Treinamento do DeepSpeech 2 com um conjunto de 158 horas de dados.

[Tabela “Frameworks”, dentro de “Tabela base para reunião de técnicas, fundamentos, frameworks e datasets”, citado no Termo de Aceite de Entrega de 9 de outubro de 2024]

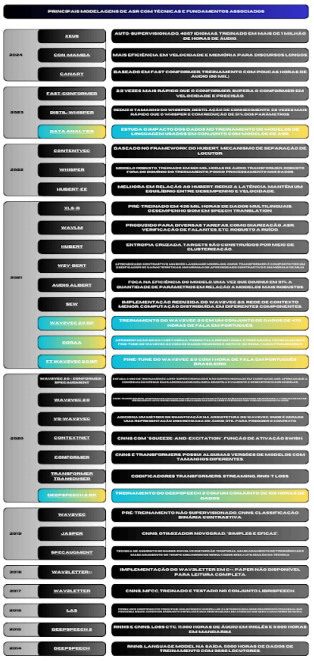
Framework	Possuir artigo?	Link artigo	Link repositório	Ano	Citações	Observações
FAIRSEQ: A Fast, Extensible Toolkit for Sequence Modeling	Sim	link	link	2019	1684	Possui modelos e implementações para translation, summarization, language modeling and other text generation tasks. Mantêm-se atualizado
EESEN: End-to-End Speech Recognition using Deep RNN Models and WFST-based Decoding	Sim	link	link	2015	968	Possui implementações para Hidden Markov models (HMMs), Gaussian mixture models (GMMs), Decision trees and phonetic questions, sem atualizações a 5 anos.
OpenSeq2Seq: Extensible Toolkit for Distributed and Mixed Precision Training of Sequence-to-Sequence Models	Sim	link	link	2018	43	Possui implementações para Neural Machine Translation, Automatic Speech Recognition, Speech Synthesis, Language Modeling, NLP tasks (sentiment analysis). Sem atualizações a 4 anos.
HTK	-		link	-	-	Voltado para HMMs, parece não ter atualizações desde 2016.
ESPnet	-		link	-	-	Possui uma série de aplicações voltadas para speech recognition, text-to-speech, speech translation, speech enhancement, speaker diarization, spoken language understanding, etc. Mantêm-se atualizado.
Kaldi	-		link	-	-	Possui aplicações com HMM, GMM e SGMMs implementadas. Mantêm-se atualizado.

DeepSpeech	Sim	link	link	2014	2695	Possui algumas implementações voltadas para STT. Sem atualizações há 3 anos.
Nabu	-		link	-	-	Baseado em Tensorflow, oferece recursos de ASR. Não foram encontradas referências sobre as últimas atualizações
Espresso	-		link	-	-	Baseado no Pytorch e no Fairseq, promove algumas implementações em ASR. Mantêm-se atualizado
Athena	-		link	-	-	Possui aplicações para Automatic Speech Recognition, Speech Synthesis, Voice activity detection, Wake Word Spotting, etc. Sem atualizações há 2 anos.

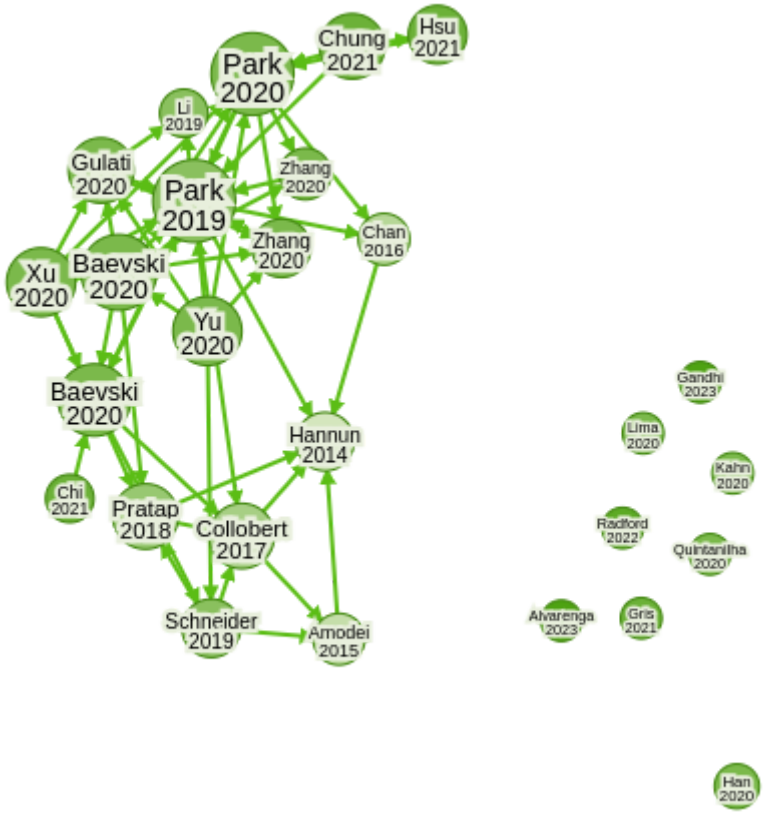
[Tabela “Datasets”, dentro de “Tabela base para reunião de técnicas, fundamentos, frameworks e datasets”, citado no Termo de Aceite de Entrega de 9 de outubro de 2024]

Artigo	Possui artigo?	Link artigo	Link dataset	Ano	Citações	Observações
CORAA: a large corpus of spontaneous and prepared speech manually validated for speech recognition in Brazilian Portuguese	Sim	link		2021	14	
VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation	Sim	link		2021	420	
Librispeech: An ASR corpus based on public domain audio books	Sim	link		2020	3343	
Common Voice: A Massively-Multilingual Speech Corpus	Sim	link		2019	152	

[Linha do Tempo “Principais Modelagens de ASR com Técnicas e Fundamentos Associados”, citado no Termo de Aceite de Entrega de 9 de outubro de 2024]



[Gráfico “Correlação entre autores”, citado no Termo de Aceite de Entrega de 9 de outubro de 2024]



APÊNDICE 3

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 16 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Daniel Ribeiro da Silva

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Foi realizado um levantamento e análise inicial dos frameworks disponíveis e atualizados para o campo de ASR, um panorama das aplicações levantadas está presente em Stage 5 - Principais técnicas e frameworks usados nos últimos anos no campo de ASR na tabela intitulada Frameworks.

- 27 frameworks encontrados;
- 8 frameworks já descartados;
- 19 frameworks selecionados: Fairseq, ESPnet, Kaldi, Espresso, Transformers, Speech Brain, NeMo, Whisper, Coqui STT, Librosa, Pydub, TorchAudio, HuggingSound, Lingvo, KenLM, PaddleSpeech, CMU Sphinx, Vosk e Sherpa.

Dentre algumas características dos frameworks, pode-se destacar exemplos de aplicações já voltadas a algumas funções:

- Manipulação de arquivos de áudio: Pydub;
- Manipulação de características básicas em Processamento de áudio: Librosa e TorchAudio;
- Suporte para finetune de modelos: Fairseq, ESPnet, NeMo e HuggingSound;
- Modelos abertos disponíveis: Transformers, NeMo e Whisper;
- Dentre outros.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Para cada framework selecionado, gastar pelo menos alguns minutos explorando suas funcionalidades internas e, assim, realizar uma nova filtragem.
Após a nova filtragem, aprofundar no estudo dos principais frameworks e, com isso, obter domínio no seu uso.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

[Tabela “Frameworks”, dentro de “Stage 5 - Principais técnicas e frameworks usados nos últimos anos no campo de ASR”, citado no Termo de Aceite de Entrega de 16 de outubro de 2024]

Framework	Possui artigo?	Link artigo	Link repositório	Ano	Citações	Desenvolvedor	Observações
FAIRSEQ: A Fast, Extensible Toolkit for Sequence Modeling	Sim	link	link	2019	1684	Meta	Possui modelos e implementações para translation, summarization, language modeling and other text generation tasks. Mantêm-se atualizado
~~EESEN: End-to-End Speech Recognition using Deep RNN Models and WFST-based Decoding~~	~~Sim~~	link	link	~~2015~~	~~968~~	~~Autoria própria~~	~~Possui implementações para Hidden Markov models (HMMs), Gaussian mixture models (GMMs), Decision trees and phonetic questions, sem atualizações a 5 anos.~~
~~OpenSeq2Seq: Extensible Toolkit for Distributed and Mixed Precision Training of Sequence-to-Sequence Models~~	~~Sim~~	link	link	~~2018~~	~~43~~	~~Nvidia~~	~~Possui implementações para Neural Machine Translation, Automatic Speech Recognition, Speech Synthesis, Language Modeling, NLP tasks (sentiment analysis). Sem atualizações a 4 anos.~~
~~HTK~~	~~Não~~	-	link	-	-	~~Autoria própria~~	~~Voltado para HMMs, parece não ter atualizações desde 2016.~~
ESPnet	Sim	link	link	2018	1641	Autoria própria	Possui uma série de aplicações voltadas para speech recognition, text-to-speech, speech translation, speech enhancement, speaker diarization, spoken language understanding, etc. Mantêm-se atualizado.

Kaldi	Sim	link	link	2011	7509	Autoria própria	Possui aplicações com HMM, GMM e SGMMs implementadas. Mantêm-se atualizado.
~~DeepSpeech~~	~~Sim~~	link	link	~~2014~~	~~2695~~	~~Mozilla~~	~~Possui algumas implementações voltadas para STT. Sem atualizações há 3 anos.~~
~~Nabu~~	~~Não~~	-	link	-	-	~~Autoria própria~~	~~Baseado em Tensorflow, oferece recursos de ASR. Não foram encontradas referências sobre as últimas atualizações~~
Espresso	Sim	link	link	2019	89	Autoria própria	Baseado no Pytorch e no Fairseq, promove algumas implementações em ASR. Mantêm-se atualizado.
~~Athena~~	~~Não~~	-	link	-	-	~~Autoria própria~~	~~Possui aplicações para Automatic Speech Recognition, Speech Synthesis, Voice activity detection, Wake Word Spotting, etc. Sem atualizações há 2 anos.~~
Transformers	Não	-	link	-	-	Hugging Face	Oferece alguns modelos pré-treinados para ASR. Mantêm-se atualizado.
Speech Brain	Sim	link	link	2021	654	Autoria própria	Modelos pré-treinados, datasets, tutoriais e outras coisas. Mantêm-se atualizado.
Nemo	Sim	link	link	2019	271	nvidia	Suporte aos modelos da nvidia. Mantêm-se atualizado.

Whisper	Sim	link	link	2022	2889	OpenAI	Suporte ao modelo Whisper e suas diferentes implementações. Mantêm-se atualizado.
~~Julius~~	~~Sim~~	link	link	~~2001~~	~~618~~	~~Autoria própria~~	~~Modelos mais antigos como HMMs. Sem atualizações há 6 meses.~~
Coqui STT	Não	-	link	-	-	Coqui	Oferece modelos pré-treinados, inferência em tempo real e scripts de treinamento. Sem atualizações há 1 ano.
Librosa	Não	-	link	-	-	Autoria própria	Biblioteca para funções de processamento de áudio mais básico. Mantêm-se atualizado.
Pydub	Não	-	link	-	-	Autoria própria	Biblioteca para manipulação de arquivos de áudio. Sem atualizações há 2 anos.
TorchAudio	Sim	link	link	2021	195	Pytorch	Funções mais básicas de procesamento de áudio. Mantêm-se atualizado.
HuggingSound	Não	-	link	-	-	Autoria própria	Suporte a modelos de ASR do HuggingFace. Mantêm-se atualizado.
Lingvo	Sim	link	link	2019	209	Tensorflow	Baseado em tensorflow, oferece alguns modelos para análise. Mantêm-se atualizado.
KenLM	Sim	link	link	2011	1673	Autoria	Implementação de recursos possibilitados pelo KenLM. Mantêm-se atualizado.

						própria	
PaddleSpeech	Sim	link	link	2022	23	Autoria própria	Oferece algumas funcionalidades de ASR. Mantêm-se atualizado.
~~OpenVINO~~	~~Não~~	-	link	-	-	~~Intel~~	~~Funcionalidades para IA no geral, pouca coisa para ASR. Mantêm-se atualizado.~~
CMU Sphinx (PocketSphinx)	Sim	link	link	2006	631	Autoria própria	Funcionalidades para ASR. Mantêm-se atualizado.
Vosk	Não	-	link	-	-	Autoria própria	Voltado a implementação de sistemas de ASR offlines. Mantêm-se atualizado.
Sherpa	Não	-	link	-	-	Autoria própria	Focado em modelos E2E. Mantêm-se atualizado.

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 31 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Daniel Ribeiro da Silva

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Na Semana anterior, foram selecionados 19 frameworks para estudo mais aprofundado e posterior descarte dos menos relevantes, uma sugestão para essa Semana foi a classificação dos frameworks para obtenção de um levantamento geral de suas funcionalidades.

A classificação sugerida foi realizada e está disponível de forma textual em:

- Stage 6 - Classificação e visão geral dos frameworks selecionados

Também disponível com uma visualização mais simples em:

- Visualização da classificação dos frameworks.jpeg

Durante a Semana foi descartado 1 framework e adicionados 2 para a classificação. Totalizando 20 frameworks. A atualização da planilha base para reunião dos principais pontos está disponível em:

- Stage 6 - Principais técnicas e frameworks usados nos últimos anos no campo de ASR

Obviamente, nem todos os frameworks serão usados nas próximas Semanas da Residência em IA, então considero que os principais, dado o planejamento, sejam:

- Pydub, TorchAudio, HuggingSound, Fairseq e KenLM.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Realizar a implementação do Paper
Brazilian Portuguese Speech Recognition Using Wav2vec 2.0.pdf e tentar utilizar os estudos desenvolvidos em A Data-Centric Approach for Portuguese Speech Recognition: Language M… em junção com a implementação.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

[Documento “Stage 6 - Classificação e visão geral dos frameworks selecionados”, citado no Termo de Aceite de Entrega de 31 de outubro de 2024]

Classificação de frameworks:

- Manipulação de arquivos de áudio;
- Pré-processamento de áudios;
- Treinamento e/ou finetune de modelos;
- Inferência com modelos pré-treinados.

FAIRSEQ - https://github.com/facebookresearch/fairseq

- Foco em aplicações de modelagem de sequência;
- Suporte a modelos de tradução, sumarização, modelagem de linguagem e outras tarefas de geração de texto;
- Desenvolvido em Pytorch;
- Possui implementações de modelos descritos em papers, como redes neurais convolucionais (CNN), redes LSTM, Transformers, RoBERTa, wav2vec, mBART, entre outros;
- Oferece suporte a técnicas de pré-treinamento e ajuste fino;
- Permite a reprodução de experimentos;
- Permite treinamentos especializados;

ESPnet - https://espnet.github.io/espnet/

- Desenvolvido em Pytorch;
- Oferece suporte para ASR, TTS, ST, VC, entre outros;
- Oferece modelos pré-treinados;
- Oferece suporte ao treinamento e finetune de modelos;
- Possui implementações para inferências em tempo real e em arquivos de áudio.

Kaldi - https://kaldi-asr.org/

- Projetado principalmente em C++, com algumas implementações em python;
- Oferece recursos voltados a RNNs e GMM-HMM, não foram encontrados referências a métodos mais recentes;
- Oferece modelos para treinamento e inferência;
- Oferece alguns métodos de extração de características, com MFCC e PLP.

Espresso - https://github.com/freewym/espresso

- Baseado no FAIRSEQ;
- A ideia do framework é realizar as mesmas implementações com processamento de GPU mais eficiente;
- Rapidez;

Transformers - https://github.com/huggingface/transformers

- A biblioteca com um todo oferece modelos para uso em NLP, Visão Computacional e Processamento de áudio;
- Modelos pré-treinados;
- Suporte a treinamento e finetune de modelos;
- Suporte a inferência;
- Suporte a modificações nos pesos dos modelos para eventuais testes;

Speech Brain - https://github.com/speechbrain/speechbrain

- Desenvolvido em Pytorch;
- Voltado a tecnologias que facilitem o diálogo, como voz e texto, mas também tem aplicações para EEG;
- Disponibiliza Modelos pré-treinados e possibilita treinamento e finetune de modelos;
- Atua em tarefas como reconhecimento de fala, identificação de locutores, separação e aprimoramento de fala, entre outras.

Whisper - https://github.com/openai/whisper

- Oferece todo o suporte para inferência com o modelo Whisper.

Librosa - https://github.com/librosa/librosa/tree/main

- Manipulação de arquivos de áudio;
- Extração de características do áudio;
- Transformações de sinais;
- Pré-processamento de áudio no geral.

Pydub - https://github.com/jiaaro/pydub

- Carrega arquivos de áudio;
- Corta e concatena áudios;
- Ajusta características como volume;

- Aplica efeitos;
- Exporta.

TorchAudio - https://github.com/pytorch/audio

- Extensão do Pytorch voltado ao processamento de áudio;
- Suporte a GPU;
- Carregar e exportar arquivos de áudio;
- Funções de processamento de áudio;
- Extração de características;
- Transformações.

HuggingSound - https://github.com/jonatasgrosman/huggingsound

- Oferece suporte a modelos CTC;
- Possibilidade de realizar avaliação e ajuste fino de modelos;
- Suporte a adição de um modelo de linguagem para auxílio na transcrição;

Lingvo - https://github.com/tensorflow/lingvo

- Baseado em TensorFlow;
- Foco em modelos de sequência, como Reconhecimento Automático de Fala (ASR), Modelagem de Linguagem, Tradução Automática e Detecção de Objetos em 3D;
- Possui algumas implementações de modelos;
- Suporte a treinamento e ajuste fino de modelos.

KenLM - https://github.com/kpu/kenlm

- Focado em velocidade e eficiência na criação e inferência de modelos de linguagem n-grama;
- Em ASR, é útil para modelar saídas dos modelos de transcrição de fala;
- HuggingSound a utiliza em paralelo.

PaddleSpeech - https://github.com/PaddlePaddle/PaddleSpeech

- Possui implementações de modelos;
- Roda modelos em streaming;
- Suporte a grafemas em chinês;
- Treinamento e finetune de modelos.

PocketSphinx - https://github.com/cmusphinx/pocketsphinx

- Voltado a dispositivos móveis ou sistemas embarcados;
- Não é atual em relação ao estado-da-arte;
- Compatível com C e Python;
- Usado para aplicações simples de reconhecimento de fala;
- ASR para streaming e áudios prontos.

Vosk - https://github.com/alphacep/vosk-api

- Voltado a soluções mais simples, possui modelos de aproximadamente 50 mb;
- Suporta treinamento juntamente com o Kaldi;
- Suporte a várias linguagens de programação, como Python, C, C#, Ruby, entre outras;
- Implementação para várias línguas.

Sherpa - https://github.com/k2-fsa/sherpa

- Suporte a C++ e Python;
- Usado para colocar em deploy os modelos de ASR;
- Permite inferência com diversos modelos.

SoundFile - https://github.com/bastibe/python-soundfile

- Lê e escreve arquivos de áudio;
- Utiliza Numpy para representação do áudio;
- É possível acessar arquivos grandes em blocos, o que facilita a manipulação.

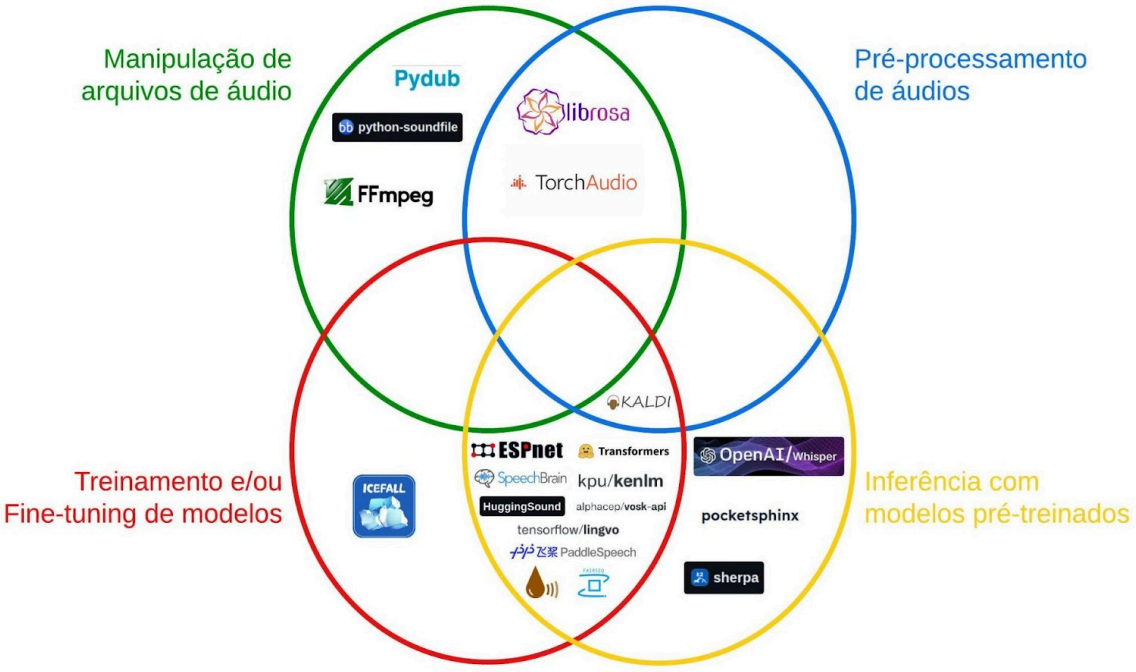
FFmpeg - https://github.com/FFmpeg/FFmpeg

- Manipulação de arquivos de vídeo e áudio;
- Conversão de formatos;
- Compressão e descompressão de áudio;
- Suporte edições, como corte, junção, redimensionamento, entre outras;
- Funciona em aplicações streaming.

Icefall - https://github.com/k2-fsa/icefall

- Suporte ao treinamento e finetune de vários modelos;
- Uso em paralelo com o Sherpa, no Icefall realiza-se o treinamento dos modelos e no Sherpa o deploy.

[Gráfico “Visualização da classificação dos frameworks”, citado no Termo de Aceite de Entrega de 31 de outubro de 2024]



[Tabela “Frameworks”, dentro de “Stage 6 - Principais técnicas e frameworks usados nos últimos anos no campo de ASR”, citado no Termo de Aceite de Entrega de 31 de outubro de 2024]

Framework	Possui artigo?	Link artigo	Link repositório	Ano	Citações	Desenvolvedor	Observações
FAIRSEQ: A Fast, Extensible Toolkit for Sequence Modeling	Sim	link	link	2019	1684	Meta	Possui modelos e implementações para translation, summarization, language modeling and other text generation tasks. Mantêm-se atualizado
~~EESEN: End-to-End Speech Recognition using Deep RNN Models and WFST-based Decoding~~	~~Sim~~	link	link	~~2015~~	~~968~~	~~Autoria própria~~	~~Possui implementações para Hidden Markov models (HMMs), Gaussian mixture models (GMMs), Decision trees and phonetic questions, sem atualizações a 5 anos.~~
~~OpenSeq2Seq: Extensible Toolkit for Distributed and Mixed Precision Training of Sequence-to-Sequence Models~~	~~Sim~~	link	link	~~2018~~	~~43~~	~~Nvidia~~	~~Possui implementações para Neural Machine Translation, Automatic Speech Recognition, Speech Synthesis, Language Modeling, NLP tasks (sentiment analysis). Sem atualizações a 4 anos.~~
~~HTK~~	~~Não~~	-	link	-	-	~~Autoria própria~~	~~Voltado para HMMs, parece não ter atualizações desde 2016.~~
ESPnet	Sim	link	link	2018	1641	Autoria própria	Possui uma série de aplicações voltadas para speech recognition, text-to-speech, speech translation, speech enhancement, speaker diarization, spoken language understanding, etc. Mantêm-se atualizado.
Kaldi	Sim	link	link	2011	7509	Autoria	Possui aplicações com HMM, GMM e SGMMs implementadas. Mantêm-se atualizado.

						própria	
~~DeepSpeech~~	~~Sim~~	~~link~~	~~link~~	~~2014~~	~~2695~~	~~Mozilla~~	~~Possui algumas implementações voltadas para STT. Sem atualizações há 3 anos.~~
~~Nabu~~	~~Não~~	-	~~link~~	-	-	~~Autoria própria~~	~~Baseado em Tensorflow, oferece recursos de ASR. Não foram encontradas referências sobre as últimas atualizações~~
Espresso	Sim	link	link	2019	89	Autoria própria	Baseado no Pytorch e no Fairseq, promove algumas implementações em ASR. Mantêm-se atualizado.
~~Athena~~	~~Não~~	-	~~link~~	-	-	~~Autoria própria~~	~~Possui aplicações para Automatic Speech Recognition, Speech Synthesis, Voice activity detection, Wake Word Spotting, etc. Sem atualizações há 2 anos.~~
Transformers	Não	-	link	-	-	Hugging Face	Oferece alguns modelos pré-treinados para ASR. Mantêm-se atualizado.
Speech Brain	Sim	link	link	2021	654	Autoria própria	Modelos pré-treinados, datasets, tutoriais e outras coisas. Mantêm-se atualizado.
Nemo	Sim	link	link	2019	271	nvidia	Suporte aos modelos da nvidia. Mantêm-se atualizado.

Whisper	Sim	link	link	2022	2889	OpenAI	Suporte ao modelo Whisper e suas diferentes implementações. Mantêm-se atualizado.
~~Julius~~	~~Sim~~	link	link	~~2001~~	~~618~~	~~Autoria própria~~	~~Modelos mais antigos como HMMs. Sem atualizações há 6 meses.~~
~~Coqui STT~~	~~Não~~	-	link	-	-	~~Coqui~~	~~Oferece modelos pré treinados, inferência em tempo real e scripts de treinamento. Sem atualizações há 1 ano.~~
Librosa	Não	-	link	-	-	Autoria própria	Biblioteca para funções de processamento de áudio mais básico. Mantêm-se atualizado.
Pydub	Não	-	link	-	-	Autoria própria	Biblioteca para manipulação de arquivos de áudio. Sem atualizações há 2 anos.
TorchAudio	Sim	link	link	2021	195	Pytorch	Funções mais básicas de procesamento de áudio. Mantêm-se atualizado.
HuggingSound	Não	-	link	-	-	Autoria própria	Suporte a modelos de ASR do HuggingFace. Mantêm-se atualizado.
Lingvo	Sim	link	link	2019	209	Tensorflow	Baseado em tensorflow, oferece alguns modelos para análise. Mantêm-se atualizado.
KenLM	Sim	link	link	2011	1673	Autoria	Implementação de recursos possibilitados pelo KenLM. Mantêm-se atualizado.

						própria	
PaddleSpeech	Sim	link	link	2022	23	Autoria própria	Oferece algumas funcionalidades de ASR. Mantêm-se atualizado.
~~OpenVINO~~	~~Não~~	-	link	-	-	~~Intel~~	~~Funcionalidades para IA no geral, pouca coisa para ASR. Mantêm-se atualizado.~~
CMU Sphinx (PocketSphinx)	Sim	link	link	2006	631	Autoria própria	Funcionalidades para ASR. Mantêm-se atualizado.
Vosk	Não	-	link	-	-	Autoria própria	Voltado a implementação de sistemas de ASR offlines. Mantêm-se atualizado.
Sherpa	Não	-	link	-	-	Autoria própria	Focado em modelos E2E. Mantêm-se atualizado.
soundfile	Não	-	link	-	-	Autoria própria	Manipulação de arquivos de áudio.
ffmpeg	Não	-	link	-	-	Autoria própria	Manipulação de arquivos de áudio.

Icefall	Não	-	link	-	-	Autoria própria	Treinamento e finetune de vários modelos.

APÊNDICE 4

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 7 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Daniel Ribeiro da Silva

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

A entrega prevista para essa Semana seria a implementação completa de um paper, porém ao analisar melhor o escopo do paper e os testes que pretendo fazer nas próximas Semanas, realizar a implementação completa do trabalho desenvolvido iria gastar um tempo desnecessário com testes que fogem dos objetivos da Residência. Então, para essa Semana, o teste principal foi o finetuning do modelo Wav2vec 2.0 em um dataset que ele não havia visto anteriormente (CORAA) para análise de quão boa é a sua melhora quando adicionado um domínio e como é o resultado em comparação com o Whisper (um modelo maior multilingual desenvolvido pela OpenAI). A métrica usada para avaliação foi o WER e seguem os resultados abaixo:

- Whisper Large v3: 31,46;
- Wav2vec 2.0 pré-treinado: 51,84;
- Wav2vec 2.0 com finetuning: 28,63.

Como comparação, segue os tempos de inferência totais na base usada para teste:

- Whisper Large v3: 149 min;
- Wav2vec 2.0: 5,95 min.

Os códigos desenvolvidos durante a semana estão disponíveis em
Stage 7 - Códigos desenvolvidos.ipynb

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Encontrar ou desenvolver através de Web Scraping uma base de dados voltada para um contexto de aplicação, como saúde. A base de dados servirá para ajuste fino de modelos, para posterior desenvolvimento de um sistema de ASR voltado a contextos específicos em contrapartida ao uso de grandes modelos generalistas.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

[Código retirado de “Stage 7 - Códigos desenvolvidos.ipynb”, citado no Termo de Aceite de Entrega de 7 de novembro de 2024]

```
# -*- coding: utf-8 -*-
"""Stage 7 - Códigos desenvolvidos.ipynb

Automatically generated by Colab.

Original file is located at

https://colab.research.google.com/drive/1TGBzqU611tHFtPPJlsOtA4rPyhvGPF7f

# Stage 7 - Códigos desenvolvidos

Os códigos presentes neste notebook foram executados em uma máquina pessoal, então o ambiente do Colab está sendo usado apenas como repositório para compartilhamento.
"""

import pandas as pd
import librosa
import random
import torch
import time
import numpy as np
from IPython.display import Audio
from torch.utils.data import DataLoader
from huggingsound import TrainingArguments, ModelArguments, SpeechRecognitionModel, TokenSet
from jiwer import wer, cer
from transformers import (
```

```
    Wav2Vec2ForCTC,
    Wav2Vec2ProcessorWithLM,
    Wav2Vec2Processor,
    WhisperProcessor,
    WhisperForConditionalGeneration
)
import re
import string
import logging
from num2words import num2words
from tqdm import tqdm
from copy_script_normalizacao_modificado import MarkPreprocessing

df_train =
pd.read_csv('../CORAA/_metadata_/metadata_train_final.csv')
df_test = pd.read_csv('../CORAA/_metadata_/metadata_test_final.csv')
df_test = df_test[['file_path', 'text']]
df_test.to_csv('metadata_test_coraa.csv', index=False)

"""## Seleciona 10 horas para treino e 5 horas para validação"""

# Carrega o CSV original
df = pd.read_csv("../CORAA/_metadata_/metadata_train_final.csv")

# Lista para armazenar os dados dos áudios selecionados
selected_audios = []
total_duration = 0  # Duração total dos áudios selecionados (em
segundos)
target_duration = 15 * 60 * 60  # 15 horas em segundos

# Itera pelas linhas do DataFrame
```

```
for index, row in df.iterrows():
    audio_path = '../CORAA/' + row['file_path']

    try:
        # Carrega o áudio e obtém a duração
        duration = librosa.get_duration(filename=audio_path)

         # Verifica se o áudio está entre 5 e 8 segundos e se ainda não atingimos o limite
           if 5 <= duration <= 8 and total_duration + duration <= target_duration:
            selected_audios.append(row)
            total_duration += duration

        # Para se já atingimos o total desejado de 15 horas
        if total_duration >= target_duration:
            break

    except Exception as e:
        print(f"Erro ao processar {audio_path}: {e}")

# Cria um DataFrame com os áudios selecionados
filtered_df = pd.DataFrame(selected_audios)

# Embaralha os dados aleatoriamente
filtered_df = filtered_df.sample(frac=1, random_state=42).reset_index(drop=True)

# Variáveis para controle das durações de treino e validação
train_duration = 10 * 60 * 60  # 10 horas em segundos
train_audios = []
```

```
train_total_duration = 0

# Separa 10 horas para treino
for index, row in filtered_df.iterrows():
    audio_path = '../CORAA/' + row['file_path']

    try:
        duration = librosa.get_duration(filename=audio_path)

        if train_total_duration + duration <= train_duration:
            train_audios.append(row)
            train_total_duration += duration
        else:
            break

    except Exception as e:
        print(f"Erro ao processar {audio_path}: {e}")

# Cria DataFrames para treino e validação
train_df = pd.DataFrame(train_audios)
val_df = filtered_df.drop(train_df.index).reset_index(drop=True)

# Salva os resultados em arquivos CSV
train_df.to_csv("audios_treino_10horas.csv", index=False)
val_df.to_csv("audios_validacao_5horas.csv", index=False)

print(f"Total de áudios para treino: {len(train_df)}")
print(f"Duração total dos áudios de treino: {train_total_duration /
3600:.2f} horas")
print(f"Total de áudios para validação: {len(val_df)}")
```

```
print(f"Duração total dos áudios de validação: {(total_duration - train_total_duration) / 3600:.2f} horas")

"""## Análise CORAA"""

df_train = pd.read_csv('audios_treino_10horas.csv')
df_val = pd.read_csv('audios_validacao_5horas.csv')
df_train

print(df_train['variety'].value_counts())
print(df_train['task'].value_counts())
print(df_val['variety'].value_counts())
print(df_val['task'].value_counts())

df_train = df_train[['file_path', 'text']]
df_val = df_val[['file_path', 'text']]
df_train

def ler_audio(nome_arquivo):
    audio = Audio(nome_arquivo)
    return audio

index = 62
print(df_train['text'][index])
ler_audio('../CORAA/' + df_train['file_path'][index])

df_train = pd.read_csv('metadata_train_10_horas_coraa.csv')
df_val = pd.read_csv('metadata_val_5_horas_coraa.csv')

df_train[df_train['text'].str.contains('ũ', na=False)]
```

```
df_train['text'] = df_train['text'].str.replace('ũ', 'ão',
regex=False)
df_val['text'] = df_val['text'].str.replace('ũ', 'ão', regex=False)

df_train['text'][504]

# Concatena todos os textos da coluna 'text' em uma única string
all_text = ''.join(df_train['text'].astype(str))

# Obtém os caracteres únicos
unique_chars = sorted(set(all_text))

print("Caracteres únicos na coluna 'text':")
print(unique_chars)

# Concatena todos os textos da coluna 'text' em uma única string
all_text = ''.join(df_val['text'].astype(str))

# Obtém os caracteres únicos
unique_chars = sorted(set(all_text))

print("Caracteres únicos na coluna 'text':")
print(unique_chars)

# Conjunto para armazenar taxas de amostragem únicas
unique_sample_rates = set()

# Itera sobre cada áudio no CSV
for index, row in df_train.iterrows():
    audio_path = '../CORAA/' + row['file_path']
```

```
    try:
        # Carrega o áudio e obtém a taxa de amostragem
            _, sr = librosa.load(audio_path, sr=None)   # `sr=None`
mantém a taxa de amostragem original
        unique_sample_rates.add(sr)
    except Exception as e:
        print(f"Erro ao carregar {audio_path}: {e}")

# Exibe as taxas de amostragem únicas
print("Taxas de amostragem presentes no dataset:")
for sr in unique_sample_rates:
    print(f"{sr} Hz")

# Conjunto para armazenar taxas de amostragem únicas
unique_sample_rates = set()

# Itera sobre cada áudio no CSV
for index, row in df_val.iterrows():
    audio_path = '../CORAA/' + row['file_path']

    try:
        # Carrega o áudio e obtém a taxa de amostragem
            _, sr = librosa.load(audio_path, sr=None)   # `sr=None`
mantém a taxa de amostragem original
        unique_sample_rates.add(sr)
    except Exception as e:
        print(f"Erro ao carregar {audio_path}: {e}")

# Exibe as taxas de amostragem únicas
print("Taxas de amostragem presentes no dataset:")
for sr in unique_sample_rates:
```

```
    print(f"{sr} Hz")

df_train.to_csv('metadata_train_10_horas_coraa.csv', index=False)
df_val.to_csv('metadata_val_5_horas_coraa.csv', index=False)

"""## Treinamento"""

def fine_tuning_w2v(pasta_saida, qtd_steps):
    df_train = pd.read_csv('metadata_train_10_horas_coraa.csv')
    df_val = pd.read_csv('metadata_val_5_horas_coraa.csv')

    train_data = []
    eval_data = []

    device = "cuda" if torch.cuda.is_available() else "cpu"
    print(f"\nDevice: {device}\n")

    for index, row in df_train.iterrows():
        data_point = {
            "path": str('../CORAA/' + row['file_path']),
            "transcription": str(row['text'])
        }
        train_data.append(data_point)

    for index, row in df_val.iterrows():
        data_point = {
            "path": str('../CORAA/' + row['file_path']),
            "transcription": str(row['text'])
        }
        eval_data.append(data_point)
```

```
                                                    model              =
SpeechRecognitionModel("lgris/wav2vec2-large-xlsr-open-brazilian-por
tuguese", device=device)
    output_dir = pasta_saida

     # tokens = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j',
'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w',
'x', 'y', 'z', 'à', 'á', 'â', 'ã', 'ç', 'é', 'ê', 'í', 'ó', 'ô',
'õ', 'ú']
    # token_set = TokenSet(tokens)

    training_args = TrainingArguments(
        learning_rate=3e-4,
        max_steps=qtd_steps,
        eval_steps=2000,
        per_device_train_batch_size=2,
        per_device_eval_batch_size=2,
        save_total_limit=3,
    )

    model_args = ModelArguments(
        activation_dropout=0.1,
        hidden_dropout=0.1,
    )

    model.finetune(
        output_dir,
        train_data=train_data,
        eval_data=eval_data,
        training_args=training_args,
        model_args=model_args,
```

```
        # token_set=token_set,
    )

    print('\n')
    print('===========================================')
    print("Fine Tuning Finalizado")
    print('===========================================')
    print('\n')


if __name__ == '__main__':

    fine_tuning_w2v(pasta_saida='models/teste4', qtd_steps=10000)
    torch.cuda.empty_cache()

"""## Teste"""

def teste_w2v_sem_lm_gpu(modelo, csv_saida='teste.csv',
device='cuda'):

     device = torch.device("cuda" if torch.cuda.is_available() else
"cpu")

    print("Device: ", device)

    df = pd.read_csv('metadata_test_coraa.csv')

    df['predicao'] = ''
    df['tempo_inferencia'] = ''

    model = Wav2Vec2ForCTC.from_pretrained(modelo).to(device)
```

```
    processor = Wav2Vec2Processor.from_pretrained(modelo)

    for i in range(len(df)):
        inicio = time.time()

        caminho_audio = '../CORAA/' + df['file_path'][i]

            audio, taxa_de_amostragem = librosa.load(caminho_audio,
sr=16000)
                  inputs = processor(audio, sampling_rate=16000,
return_tensors="pt").to(device)

        with torch.no_grad():
            logits = model(**inputs).logits

        predicted_ids = torch.argmax(logits, dim=-1)
        transcription = processor.batch_decode(predicted_ids)
                                          #     transcription     =
processor.batch_decode(logits.cpu().numpy()).text

        fim = time.time()

        print(f'{i}: {transcription[0]}')
        df.loc[i, 'predicao'] = str(transcription[0])
        df.loc[i, 'tempo_inferencia'] = fim - inicio

    df.to_csv(f'results/{csv_saida}', index=False)

def teste_whisper_gpu(modelo, csv_saida='teste.csv', device='cuda'):
    # Verifica o dispositivo disponível
```

```
    device = torch.device("cuda" if torch.cuda.is_available() else
"cpu")
    print("Device: ", device)

    # Carrega o arquivo CSV
    df = pd.read_csv('metadata_test_coraa.csv')
    df['predicao'] = ''
    df['tempo_inferencia'] = ''

    # Carrega o modelo e o processador Whisper
                                                    model            =
WhisperForConditionalGeneration.from_pretrained(modelo).to(device)
    processor = WhisperProcessor.from_pretrained(modelo)

    # Define o idioma para português do Brasil
    processor.tokenizer.lang = "pt"
     processor.feature_extractor.sampling_rate = 16000  # Garante a
taxa de amostragem correta

    for i in range(len(df)):
        inicio = time.time()

         # Carrega o áudio e ajusta a taxa de amostragem para 16kHz,
se necessário
        caminho_audio = '../CORAA/' + df['file_path'][i]
            audio, taxa_de_amostragem = librosa.load(caminho_audio,
sr=16000)

        # Processa o áudio
                    inputs = processor(audio, sampling_rate=16000,
return_tensors="pt").to(device)
```

```
        # Define as configurações de geração, incluindo o idioma
        with torch.no_grad():
             predicted_ids = model.generate(inputs["input_features"],
forced_decoder_ids=processor.get_decoder_prompt_ids(language="pt",
task="transcribe"))

        # Decodifica a transcrição
              transcription  =  processor.batch_decode(predicted_ids,
skip_special_tokens=True)[0]

        fim = time.time()

        print(f'{i}: {transcription}')
        df.loc[i, 'predicao'] = transcription
        df.loc[i, 'tempo_inferencia'] = fim - inicio

    # Salva o resultado no CSV
    df.to_csv(f'results/{csv_saida}', index=False)

teste_w2v_sem_lm_gpu(modelo='models/teste4',
                     device='cuda',
                     csv_saida='finetuning.csv')
torch.cuda.empty_cache()

teste_w2v_sem_lm_gpu(modelo='lgris/wav2vec2-large-xlsr-open-brazilia
n-portuguese',
                     device='cuda',
                     csv_saida='base.csv')
torch.cuda.empty_cache()
```

```
teste_whisper_gpu("openai/whisper-large-v3",
csv_saida='teste_whisper.csv')

def calculate_error(csv):
    """
       Recebe df de um dos resultados e adiciona os wer e cer
calculados
    """
    df = pd.read_csv(csv)
    error = {'wer':[], 'cer':[]}

    # calculando o wer e cer e guardando valores com tqdm
      for predicao, transcricao in tqdm(zip(df.predicao, df.text),
total=len(df)):
        try:
            print(MarkPreprocessing.normalize(str(predicao)))
                                wer_calculado = wer(transcricao,
MarkPreprocessing.normalize(str(predicao)))
                                cer_calculado = cer(transcricao,
MarkPreprocessing.normalize(str(predicao)))
            # wer_calculado = wer(transcricao, str(predicao))
            # cer_calculado = cer(transcricao, str(predicao))
        except OverflowError:
            wer_calculado = wer(transcricao, predicao)
            cer_calculado = cer(transcricao, predicao)

        error['wer'].append(wer_calculado)
        error['cer'].append(cer_calculado)

    # armazenando erros em df do resultado
    df['wer'] = error['wer']
```

```
    df['cer'] = error['cer']

    print('==============================')
    print(f'{csv}')
    print(f'WER: {np.mean(error["wer"])}')
    print(f'CER: {np.mean(error["cer"])}')


    with open('results.txt', 'a') as arquivo:

arquivo.write(f'================================================
=====\n')
        arquivo.write(f'Teste finalizado: {csv}\n')
        arquivo.write(f'WER: {np.mean(error["wer"])}\n')
        arquivo.write(f'CER: {np.mean(error["cer"])}\n')

arquivo.write(f'================================================
=====\n')

calculate_error('results/finetuning.csv')
calculate_error('results/base.csv')
calculate_error('results/teste_whisper.csv')
```

APÊNDICE 5

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 13 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Daniel Ribeiro da Silva

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

O objetivo para essa semana era a busca de uma base de dados voltada a algum contexto de aplicação específico e, assim, possibilitar o ajuste fino de modelos como o Wav2vec 2.0 para cenários específicos, com foco em resultados melhores e mais rápidos que modelos grandes como o Whisper.

Foram selecionados canais no Youtube que possuem vídeos com legendas anotadas, o que garante uma robustez na qualidade dos dados que serão usados. O contexto escolhido foi a área médica. A lista de canais segue abaixo:

- https://www.youtube.com/@institutoamato
- https://www.youtube.com/@RegeneratiNeurologia
- https://www.youtube.com/c/M%C3%A9diconaPr%C3%A1tica
- https://www.youtube.com/@oncoguia
- https://www.youtube.com/c/M%C3%A9dicoparatodavida

Para a extração das informações necessárias (áudio e legenda), será utilizado o repositório https://github.com/GO0108/katube. Uma característica importante ao buscar pelos canais foi a variabilidade de locutores, presença de diferentes sotaques e presença de vocabulário de palavras grandes, já que tais características são reportadas nos papers como importantes para a generalização dos modelos.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Realizar o ajuste fino de modelos com os dados coletados e comparar os resultados de acurácia e tempo de inferência com outras aplicações.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Instituto de Informática (INF) - UFG
Alameda Palmeiras, Quadra D, Câmpus Samambaia
Fone: (62) 3521-1181 / Fax: (62) 3521-1182

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 27 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Daniel Ribeiro da Silva

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Geração do dataset que será utilizado nos testes finais, o dataset foi gerado a partir de cinco canais do youtube sobre medicina, o objetivo foi reunir uma coletânea de áudios e transcrições que possuem um vocabulário especializado em um domínio de aplicação. Para isso, algumas características do dataset precisaram ser levadas em consideração:

- Vídeos mais recentes, uma vez que farei comparações com versões do whisper treinadas com áudios do youtube;
- Vídeos com legenda anotada, já que existe a necessidade de transcrições condizentes com a realidade para garantir um resultado mais robusto;
- Diversificação de locutores, áudios com vozes diferentes são importantes para garantirmos uma robustez na avaliação e garantir que o modelo não vai aprender características intrínsecas a um determinado falante.

Algumas características referentes ao dataset:

- 100 vídeos selecionados;
- 3939 arquivos de áudio extraídos;
- + de 4 horas de tempo de áudio.

O dataset está disponível no seguinte link:
https://drive.google.com/file/d/1aXjDkL0-HEkkmxBtR3sE7IeFHk673FBB/view?usp=sharing

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Utilizar o dataset montado para realizar os testes finais da Residência, como fine-tuning de modelos e comparações com abordagens como o Whisper.

Observação: [caso precise fazer alguma observação, de qualquer "natureza"]

ACEITE DA ENTREGA:

LEONARDO ANTÔNIO ALVES: Go!

APÊNDICE 6

Termo de Aceite de Entrega

Objetivo deste documento

Este documento faz parte do Processo da disciplina Residência em IA e tem como objetivo formalizar o aceite da entrega considerando o planejado e o realizado para o período.

Data da Reunião ("gate") de aprovação: 4 de dez. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Daniel Ribeiro da Silva

Entrega: [descrever a ENTREGA: requisitos e produtos gerados: links para textos, códigos, vídeos etc.]

Geração de um novo dataset para realização dos experimentos, o dataset anterior estava com poucos áudios em comparação aos datasets públicos descritos em papers. Os novos dados seguem a mesma lógica anterior, áudios providos de canais no Youtube com legenda anotada, recentes e com locutores diversos.

Algumas características em relação ao dataset:

- 70 vídeos selecionados;
- 10277 arquivos de áudio extraídos;
- 14 horas e 40 minutos de tempo de áudio.

O dataset está disponível no seguinte link:
https://drive.google.com/file/d/1Rakk30mCEJEYxJ8qUyWnZzDN2lis0egm/view?usp=sharing

Também, foram realizados ajustes finos no modelo Wav2vec 2.0 pré-treinado em português brasileiro, abaixo segue o melhor resultado em comparação com o benchmark do Whisper Large v3:

	WER	CER
- Wav2vec 2.0 pré-treinado:	33,85	17,36
- Whisper Large v3:	16,4	10,73
- Wav2vec 2.0 ajustado:	18,9	8,61

O resultado do Wav2vec 2.0 ajustado melhorou em comparação a sua versão pré-treinada e ficou equivalente ao resultado do Whisper, isso mostra que poucas horas de áudios conseguem tornar um modelo menor equivalente a um grande modelo como o Whisper em um determinado contexto.

Planejamento: [descrever o que pretende fazer para realizar a próxima ENTREGA]

Observação: [caso precise fazer alguma observação, de qualquer “natureza”]

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

[Ampliação do dataset citado no Termo de Aceite de Entrega de 4 de dezembro de 2024]

Foi realizada uma ampliação no dataset citado no Termo de Aceite de Entrega de 4 de dezembro de 2024, a adição visou a adição de áudios de locutores fora do domínio do treino para posterior avaliação dos modelos nessa base de dados.

A ampliação do dataset está disponível no seguinte link:
https://drive.google.com/file/d/1Cp4hcfZF_7OKvVki8nn6kCRY-w6ibo0S/view?usp=sharing

Informações sobre a ampliação do dataset:

- 21 vídeos;
- 2454 arquivos de áudio extraídos;
- 3 horas e 8 minutos de tempo de áudio.

[Testes realizados na ampliação do dataset citado no Termo de Aceite de Entrega de 4 de dezembro de 2024]

Após a ampliação do dataset, foram realizados testes com os novos dados e avaliado a melhora do Wav2vec 2.0 após a realização do ajuste fino.

	WER	CER
- Wav2vec 2.0 pré-treinado:	38,73	19,78
- Whisper large v3:	18,49	11,49
- Wav2vec 2.0 ajustado:	32,01	14,41

Apesar do Whisper large v3 possuir um resultado melhor que a versão ajustada do Wav2vec 2.0, nota-se que o contexto dos áudios de treinamento melhorou o modelo em um cenário com locutores diferentes da base de treino, mostrando a influência do domínio de aplicação no ajuste do Wav2vec 2.0.

[Modelo Wav2vec 2.0 ajustado, citado no Termo de Aceite de Entrega de 4 de dezembro de 2024]

O modelo final desenvolvido no Processo da Residência em IA está disponível no seguinte link:
https://huggingface.co/danielribeiro/wav2vec2_residencia