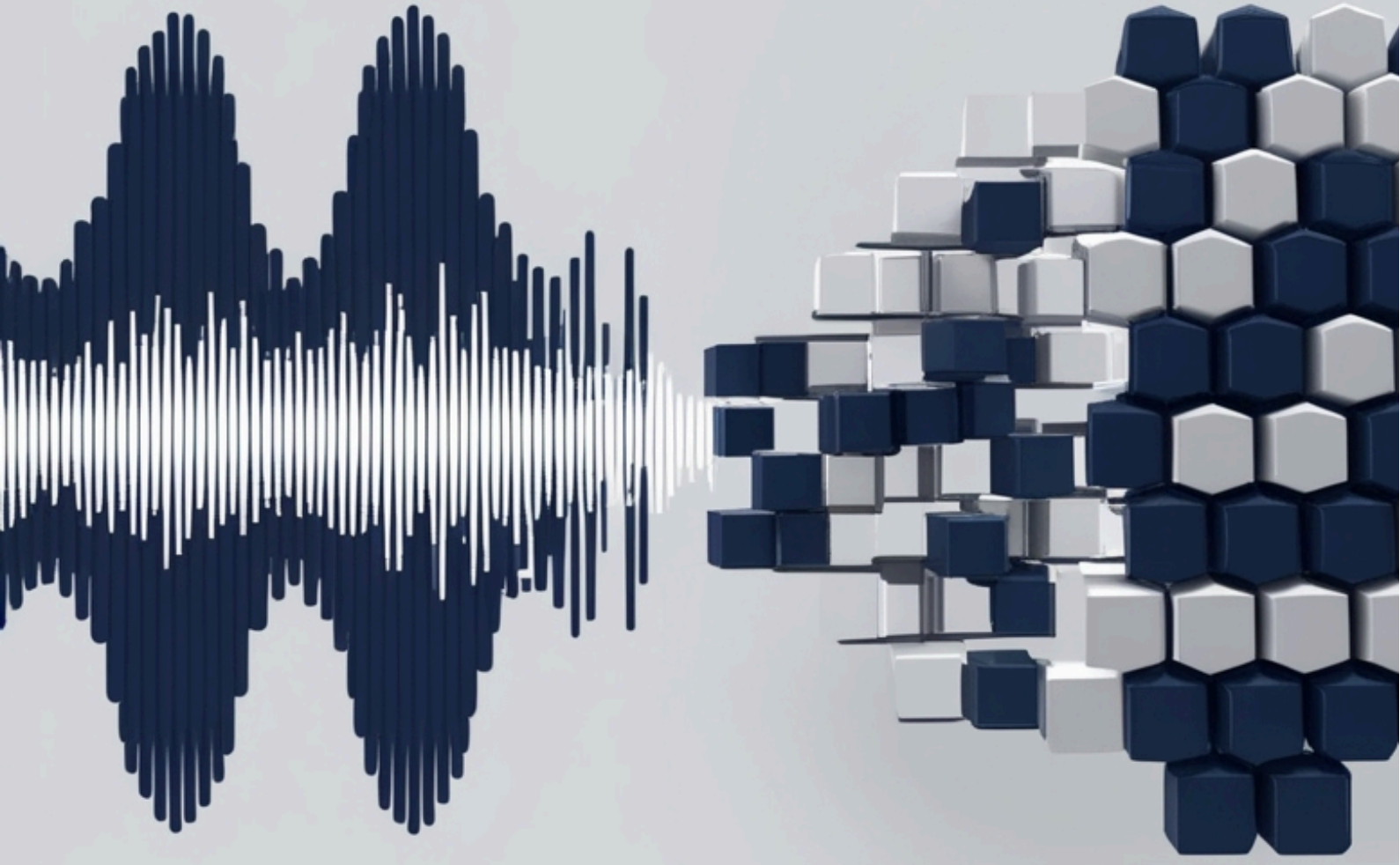
UNIVERSIDADE FEDERAL DE GOIÁS / INSTITUTO DE INFORMÁTICA

Explorando Codecs Neurais de Áudio e suas Inovações

Estudo sobre os Fundamentos e as Aplicações na Representação de Áudio

Alexandre Costa Ferro Filho



UFG
UNIVERSIDADE FEDERAL DE GOIÁS

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

ALEXANDRE COSTA FERRO FILHO

Explorando Codecs Neurais de Áudio e suas Inovações

Estudo sobre os Fundamentos e as Aplicações na Representação de Áudio

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): ALEXANDRE COSTA FERRO FILHO

Título do trabalho: Explorando Codecs Neurais de Áudio e suas Inovações

Estudo sobre os Fundamentos e as Aplicações na Representação de Áudio

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 **Alexandre Costa Ferro Filho**, **Discente**, em 16/01/2025, às 15:43, 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 16/01/2025, às 18:25, 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 **5078623** e o código CRC **AB5C8A8E**.

Referência: Processo nº 23070.000483/2025-81 SEI nº 5078623

ALEXANDRE COSTA FERRO FILHO

Explorando Codecs Neurais de Áudio e suas Inovações

Estudo sobre os Fundamentos e as Aplicações na Representação de Áudio

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.

FILHO, ALEXANDRE COSTA FERRO
Explorando Codecs Neurais de Áudio e suas Inovações [manuscrito] : Estudo sobre os Fundamentos e as Aplicações na Representação de Áudio / ALEXANDRE COSTA FERRO FILHO. - 2025.
117 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. representação de aúdio. 3. codecs de áudio neurais. I. Federson, Fernando Marques , orient. II. Título.

CDU 004

ALEXANDRE COSTA FERRO FILHO

Explorando Codecs Neurais de Áudio e suas Inovações

Estudo sobre os Fundamentos e as Aplicações na Representação de Áudio

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)

ALEXANDRE COSTA FERRO FILHO

Explorando Codecs Neurais de Áudio e suas Inovações

Estudo sobre os Fundamentos e as Aplicações na Representação de Áudio

RESUMO

Este Relatório de Conclusão de Curso tem como objetivo reunir os resultados da minha jornada para me tornar um especialista em **Codecs Neurais de Áudio**. 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 **Neural Audio Codecs**. An illustration and its narrative describe the work periods. The Appendices contain the Delivery Acceptance Terms and the results obtained during each work period.

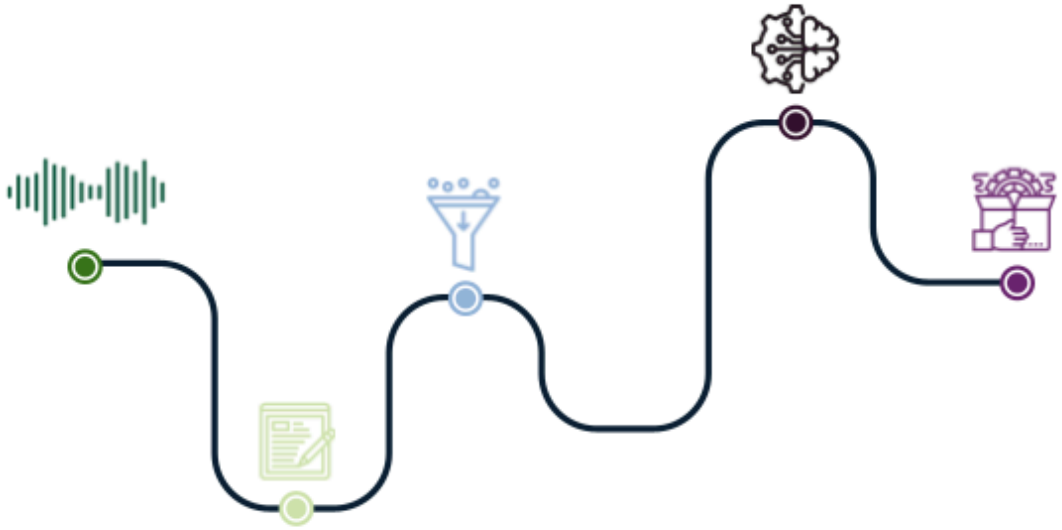
Keywords: artificial intelligence, large language models, automatic dataset generation.

Goiânia

2025

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

Minha Jornada



Alexandre Costa Ferro Filho

Especialista em: Codecs Neurais de Áudio

Semana 1

Revisão/Introdução de conceitos da Área de Processamento de Áudio e Voz.

Semanas 2-4

Revisão bibliográfica da Área e definição do escopo de estudo para a representação discreta de áudios.

Semana 5-6

Estudo aprofundado de conceitos e ferramentas no contexto de algoritmos de Codecs Neurais.

Semanas 7-8

Exploração de aplicações dentro da Área de Codecs Neurais de Áudio.

Semana 9-10

Desenvolvimento de uma aplicação e realização de testes.

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

MINHA JORNADA

Nome: Alexandre Costa Ferro Filho

Especialidade: Codecs Neurais de Áudio

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 começou na **Semana 1** com atividades para definir a área de conhecimento da minha especialização. A escolha por atuar em processamento de áudio e voz foi motivada pela afinidade com o tema e pela perspectiva de explorar diversas subáreas dentro deste campo. A formação obtida na disciplina de Processamento de Áudio e Voz do Bacharelado, assim como minha participação no núcleo de robótica Pequi Mecânico da UFG [2], foram fundamentais para minha decisão. Este último me proporcionou um contato mais prático com diferentes partes da Inteligência Artificial, ampliando minha visão sobre o tema. Para dar início à minha especialização, realizei um estudo abrangente, revisando conceitos fundamentais e explorando referências teóricas importantes. Um material de destaque foi uma revisão que contextualizava a área de forma geral[3], abordando desde algoritmos e técnicas mais antigas até métodos mais recentes, além de explicar as diferentes tarefas que podem ser exploradas no campo do processamento de áudio e voz. Complementando, a

[1] Dez semanas, entre setembro de 2024 e dezembro de 2024.
[2] Núcleo de robótica com foco em competições e desenvolvimento de soluções práticas.
[3] "A Review of Deep Learning Techniques for Speech Processing."

playlist do canal de Valerio Velardo[4] também foi essencial para consolidar conceitos básicos e estabelecer uma base sólida para o desenvolvimento da pesquisa. Mais detalhes sobre as referências utilizadas podem ser encontrados no **Apêndice 1**.

A partir da definição da área, foi possível, na **Semana 2**, realizar um levantamento geral dos modelos autorregressivos utilizados recentemente no campo do processamento de áudio e voz, com o objetivo de mapear o desenvolvimento da área como um todo, além de pesquisar as tarefas mais relevantes e interessantes que poderiam ser abordadas na pesquisa. Na **Semana 3**, dei continuidade ao levantamento bibliográfico, desta vez com foco nos modelos da área de transcrição da fala, além de delimitar o escopo do estudo para as novas abordagens aplicadas na geração de síntese de voz. Já na **Semana 4**, completei meu estudo abrangente explorando técnicas utilizadas em áreas como biometria, classificação e detecção de atividade vocal (VAD), e iniciei os estudos sobre as formas mais recentes de representação de áudio. Foi nessa etapa que fui introduzido aos codecs neurais de áudio, observando seu impacto significativo em aplicações como zero-shot TTS e SpeechLMs. Todo o material revisado e as observações feitas durante essas semanas foram organizados para consulta detalhada no **Apêndice 2**, auxiliando na continuidade e estruturação da pesquisa.

As **Semanas 5 e 6** foram períodos cruciais na Minha Jornada, marcados por um aprofundamento significativo nos estudos sobre codecs neurais de áudio e tokenizadores de fala. Na **Semana 5**, concentrei-me em entender as diferentes formas de representação e avaliar as vantagens e aplicações de cada abordagem. Os codecs neurais, utilizados como tokens acústicos, destacaram-se por seu potencial promissor, motivando um estudo aprofundado nesse campo. Para embasar essa análise, realizei uma leitura detalhada de um artigo da área[5], que forneceu uma visão abrangente sobre o estado atual, as aplicações, os potenciais e as limitações desses algoritmos. Paralelamente, iniciei a documentação dos principais frameworks de deep learning para áudio, com foco em identificar cenários ideais para sua aplicação e em como estruturar suas funções de maneira eficiente. Já na **Semana**

[4] Canal no YouTube especializado em processamento de áudio e voz.
[5] "Low Frame-rate Speech Codec: a Codec Designed for Fast High-quality Speech LLM Training and Inference"

6, aprofundei ainda mais esses estudos, com destaque para os codecs neurais pioneiros, como SoundStream e EnCodec. Revisei suas arquiteturas, técnicas de treinamento e contribuições para o campo, enquanto documentava as observações mais relevantes. Para consolidar o aprendizado, elaborei uma apresentação detalhada sobre tokenizadores de fala, abordando tanto os fundamentos teóricos quanto exemplos práticos, com ênfase nos codecs neurais. Além disso, realizei testes iniciais com frameworks aplicados aos codecs, explorando como suas implementações refletem a teoria estudada e avaliando os processos de treinamento e inferência. Todo o material levantado, incluindo as anotações sobre os frameworks e os resultados dos testes, foi organizado no **Apêndice 3** e oferece uma visão estruturada e prática que servirá como referência para etapas futuras da pesquisa.

Dando continuidade, as **Semanas 7 e 8** marcaram uma nova etapa no desenvolvimento da pesquisa, com foco na definição de possíveis aplicações práticas a serem exploradas. Na **Semana 7**, comecei a identificar abordagens que poderiam ser trabalhadas, considerando diferentes níveis de complexidade e demandas de tempo. Três possibilidades foram analisadas: refinamento de modelos de síntese de fala, otimização no transporte de dados comprimidos e seus benefícios e uso de representações comprimidas para classificação de áudio. Após breve levantamento, o Low Frame-rate Speech Codec destacou-se como a opção mais promissora, sendo priorizado nos testes práticos subsequentes. Dessa forma, na **Semana 8**, iniciei os testes práticos, priorizando as abordagens de transmissão de dados comprimidos e classificação, consideradas mais viáveis dentro do período restante da residência. Os testes relacionados à transmissão de dados comprimidos visaram avaliar o potencial ganho de desempenho e a viabilidade de aplicação em cenários de áudio em streaming, enquanto os testes de classificação focaram no desenvolvimento inicial de um código para treinar classificadores de Deepfake a partir de áudio. Embora os testes iniciais com os classificadores não tenham apresentado bons resultados, ambos os experimentos foram documentados no **Apêndice 4** e serviram de base para a construção dos códigos mais elaborados.

Na **Semana 9**, embora esperasse alcançar avanços significativos no desenvolvimento dos classificadores de Deepfake em áudio, encontrei diversas dificuldades que impediram o progresso. Entre os desafios estavam a baixa qualidade e o desbalanceamento dos datasets

iniciais, limitações na arquitetura dos classificadores que ainda precisavam ser ajustadas e a utilização de um tokenizador com desempenho inferior nos testes preliminares. Por conta disso, o foco principal dessa semana voltou-se para o desenvolvimento da aplicação de transmissão de fluxo de dados comprimidos em streaming. Nessa aplicação, fluxos capturados por microfone eram coletados em blocos de dados, comprimidos, enviados para um servidor e descomprimidos. Para testar o ganho prático do pipeline, foram simuladas condições de redes adversas, como baixa largura de banda e inconsistência. O estudo completo, incluindo os resultados e a aplicação desenvolvida, foi documentado no **Apêndice 5**, juntamente com o código disponível no GitHub[6]. Na **Semana 10**, a abordagem dos classificadores foi intensificada. Realizei testes com diversas arquiteturas neurais, incluindo modelos neurais simples de classificação e arquiteturas mais complexas, como BERT. Paralelamente, elaborei datasets mais robustos e balanceados para treino, validação e teste, além de testar tokenizadores mais promissores para a tarefa. Após vários ajustes, o melhor modelo treinado, uma MLP, alcançou 96% de acurácia com datasets da mesma origem dos dados de treino e 90% com um dataset externo, demonstrando resultados satisfatórios. Além disso, foi conduzido um teste de degradação dos áudios reconstruídos pelos codecs, avaliando o impacto de sua utilização em tarefas relacionadas ao aprendizado de máquina. Por fim, revisitei a área de síntese de voz e implementei um fine-tuning simples de um modelo zero-shot TTS (xTTS) para adaptação a um locutor específico. Essa abordagem resultou em uma síntese de alta qualidade, indicando uma aplicação promissora dentro do contexto e tempo da Residência.

Em função de tudo que vivi nesta Jornada, gostaria de deixar registrado que o processo vivenciado foi fundamental para minha especialização em codecs neurais de áudio, uma área pela qual desenvolvi grande afinidade e interesse. A oportunidade de explorar subáreas promissoras, realizar pesquisas aprofundadas e desenvolver soluções inovadoras foi extremamente enriquecedora. Agradeço por essa Jornada que não só ampliou meu conhecimento teórico, mas também aprimorou minhas habilidades práticas em desenvolvimento de códigos e técnicas específicas da área. Esse período foi crucial para

[6] https://github.com/alexandreacff/Audio_Codecs_Streaming.git

meu crescimento acadêmico e profissional, proporcionando uma base sólida para futuras pesquisas e aplicações no campo da Inteligência Artificial e processamento de áudio.

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: 18 de set. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Alexandre Costa Ferro Filho

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

As atividades desta semana foram divididas entre estudos teóricos e análises técnicas, focados na área de interesse previamente definida, **Processamento de áudio e voz**:

- **Estudo Histórico:** Realizada uma revisão sobre a evolução do processamento de áudio e voz, com ênfase nos principais marcos e avanços tecnológicos.
  - A short history of speech recognition
  - State of audio processing
- **Conteúdo Didático:** Visualização de vídeos introdutórios e teóricos, abordando fundamentos de processamento de sinal de áudio aplicado a machine learning.
  - Parte da playlist do canal do Valério Velardo.
- **Levantamento Técnico/Histórico:** Pesquisa e levantamento das principais técnicas de deep learning empregadas no processamento de áudio e voz, destacando as metodologias mais utilizadas.
  - Leitura parcial do artigo A Review of Deep Learning Techniques for Speech Processing.
  - Resumo do artigo lido:
    Resumo_A_Review_of_Deep_Learning_Techniques_for_Speech

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

Para próxima entrega planejo:

- Finalizar estudos dos fundamentos de processamento de sinal de áudio aplicado a machine learning.

- Fazer levantamento bibliográficos dos modelos autorregressivos para speech e suas limitações.
- Buscar ideias de pesquisas que podem ser desenvolvidas com outros da área.

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Resumo do Artigo - A Review of Deep Learning Techniques for Speech Processing

Introdução ao Processamento de Fala:

O artigo discute a evolução do processamento de fala, desde os modelos estatísticos tradicionais, como os Modelos Ocultos de Markov (HMMs), até as técnicas modernas de deep learning. As tarefas de processamento de fala incluem reconhecimento automático de fala (ASR), síntese de fala, reconhecimento de locutor, reconhecimento de emoções, entre outras.

Dentre essas técnicas modernas, o campo de processamento de fala foi revolucionado pelo uso do deep learning, com redes neurais profundas, convolucionais e recorrentes. Essas técnicas, que extraem padrões diretamente dos sinais de áudio, trouxeram avanços notáveis, como maior precisão e robustez em cenários com ruído ou variações de sotaque. O uso de mecanismos de atenção e modelos Transformers também contribuíram para melhorar o desempenho em tarefas de processamento de fala. No entanto, ainda há desafios a serem superados, como a necessidade de **grandes quantidades de dados rotulados** e a **interpretação dos modelos**.

Contexto:

Antes de abordar as arquiteturas neurais profundas, são discutidos termos básicos em processamento de fala, representações de baixo nível dos sinais de fala e modelos tradicionais utilizados na área.

1. **O processamento de sinais** é uma disciplina que estuda quantidades que variam no espaço ou no tempo. Sinais sonoros são variações na pressão do ar, e sinais de fala são variações de pressão geradas por nós para comunicação. Transdutores convertem esses sinais em formas elétricas. A forma de onda de um sinal periódico determina seu timbre. Para processamento, os sinais de fala são digitalizados, transformando em valores numéricos.

2. **As características da fala** são representações numéricas utilizadas para análise, reconhecimento e síntese. Elas se dividem em características no domínio do tempo, como **energia, taxa de cruzamento, pitch e LPC**, e no domínio da frequência, como **mel-espectrograma e MFCCs.**

3. Os algoritmos tradicionais utilizam **modelos simples** para extrair características de sinais de fala, que servem como entrada para modelos de classificação ou regressão. Alguns modelos citados são: Modelos de Mistura Gaussiana, Máquinas de Vetores de Suporte, Modelos Ocultos de Markov e Árvores de decisão.
   Esses algoritmos têm aplicações em **reconhecimento de fala, identificação de falantes e síntese de fala**, mas têm sido superados por redes neurais profundas devido a sua capacidade avançada.

Arquiteturas de Deep Learning:

- **Redes Neurais Recorrentes (RNNs)** e suas variantes (como LSTMs e GRUs) foram as primeiras opções para tarefas sequenciais, como reconhecimento de fala.
- **Redes Neurais Convolucionais (CNNs)** tornaram-se populares para extração de características de espectrogramas, contribuindo para tarefas como síntese de fala.
- **Transformers** e **mecanismos de autoatenção** são agora mais utilizados para lidar com dependências de longo alcance em dados de fala, especialmente em ASR e TTS.
- **Modelos Conformer**, que combinam CNNs e transformers, oferecem desempenho de última geração em tarefas de reconhecimento de fala.
- **Modelos de Difusão Probabilística** estão emergindo para tarefas como aprimoramento de fala.

Aprendizado de Representação de fala:

O artigo aborda métodos de aprendizado supervisionado, não supervisionado, semi-supervisionado e auto-supervisionado para extrair representações úteis de sinais de fala, e mostra evolução desses tipos de aprendizados.

1. **Aprendizado supervisionado:** é um processo em que o modelo é treinado com dados rotulados, ajustando seus parâmetros para minimizar a diferença entre as previsões e os rótulos verdadeiros. No processamento de fala, redes neurais como **CNNs** aprendem a identificar padrões em espectrogramas para reconhecer fonemas ou palavras, enquanto **RNNs** podem trabalhar diretamente com sinais de áudio brutos para extrair características relevantes.
2. **Aprendizado não supervisionado:** é o processamento de fala que busca aprender representações úteis de áudio sem utilizar dados anotados. Normalmente, o modelo

é pré-treinado com grandes quantidades de dados disponíveis e, depois, ajustado para tarefas com poucos dados.
    a. **Modelos probabilísticos de variáveis latentes** (PLVM) permitem aprender representações estruturais ricas, relacionando variáveis observadas e não observadas.
    b. **Modelos como autoencoders variacionais** (VAE) são usados nesse contexto para capturar padrões complexos em dados de fala, sem necessidade de supervisão explícita.
3. **Aprendizado semi-supervisionado:** otimiza modelos utilizando tanto dados rotulados quanto não rotulados. Ele combina uma perda supervisionada, calculada sobre os dados rotulados, e uma perda não supervisionada, que aproveita os dados não rotulados para aprender representações significativas. Essa abordagem melhora a **generalização do modelo**, especialmente em **cenários com poucos dados rotulados.**
4. **Aprendizado auto-supervisionado:** é uma abordagem de aprendizado de máquina que permite ao modelo aprender características profundas e robustas de dados sem a necessidade de grandes quantidades de dados rotulados. Em vez de depender de rótulos fornecidos por humanos, o SSRL utiliza tarefas pré-textuais para gerar rótulos automáticos, ou pseudo-rótulos, diretamente dos próprios dados. Isso reduz a dependência de conjuntos de dados anotados manualmente, superando limitações comuns do aprendizado supervisionado, e pode, em alguns casos, alcançar ou até superar a eficácia desses métodos.

Técnicas como **Wav2Vec 2.0** e **HuBERT** fizeram grandes avanços no pré-treinamento não supervisionado.

Tarefas de Processamento de Fala:

Estas incluem principalmente tarefas como:

- **ASR**: Converter fala em texto automaticamente.
- **Síntese de Fala**: Gerar fala a partir de texto.
- **Reconhecimento de Locutor**: Identificar ou verificar locutores a partir de áudio.
- **Aprimoramento e Separação de Fala**: Melhorar a qualidade da fala em ambientes ruidosos ou separar vozes sobrepostas.
- **Detecção de Atividade de Voz (VAD)** e **Avaliação da Qualidade da Fala**.

Detalhando mais:

- **Reconhecimento Automático de Fala (ASR)**: Usando redes neurais profundas e técnicas como os Transformers, sistemas modernos de ASR melhoraram significativamente, lidando melhor com sotaques e ambientes ruidosos. Modelos como Wav2Vec 2.0 e Whisper estão entre os mais avançados e podem lidar com tarefas multilíngues e ambientes adversos, dependendo se o ASR será utilizado em domínio específico ou não. OBS: Teste pessoais mostraram que o Conformer tem um bom desempenho com latência muito menor.

- **Síntese de Fala**: O processo inclui análise de texto, modelos acústicos e vocoders, sendo que modelos como HiFi-GAN e frameworks como FastSpeech2 permitem manipulação de voz, altura e velocidade. Os modelos são classificados em autoregressivos, que geram fala sequencialmente, e não-autoregressivos, que geram todos os elementos em paralelo. A avaliação da qualidade de fala gerada é feita por métricas como **MOS (Mean Opinion Score),MCD (Mel Cepstral Distortion) e WER (Word Error Rate).**

- **Reconhecimento de Falantes**: Focado na **identificação e verificação de falantes** com base em características vocais, este campo tem grande aplicação em autenticação biométrica. Ambas as tarefas requerem a extração de vetores de representação do falante, a partir de amostras de fala. Modelos avançados, baseados em redes neurais profundas (DNNs), como **x-vectors** e **ECAPA-TDNN**, mostraram resultados superiores na identificação e verificação de falantes. O **ECAPA-TDNN** introduz melhorias como conexões de resíduo e blocos de "Squeeze-and-Excitation" para realçar os canais mais informativos. Além disso, o uso de **mecanismos de atenção** e arquiteturas **Transformer** tem se mostrado eficaz para capturar características discriminativas, melhorando o desempenho de tarefas de reconhecimento de falantes

- **Detecção de Atividade de Voz (VAD)**: O VAD diferencia entre fala e ruído em um sinal de áudio. Sistemas desenvolvidos, reconhecendo a presença ou ausência de fala. Modelos recentes de VAD, como **NAS-VAD** e **self-attentive VAD**, utilizam deep learning para melhorar a precisão, especialmente em ambientes ruidosos. Além disso, arquiteturas como **CNNs** têm demonstrado melhor desempenho em detecção de fala em comparação com outras redes neurais

Técnicas Avançadas de Transferência de Aprendizado

O uso de adaptação de domínio, meta-aprendizado e modelos eficientes em parâmetros é destacado para melhorar o desempenho em ambientes com poucos recursos.

Desafios e Direções Futuras:

Os principais desafios incluem escassez de dados, interpretabilidade dos modelos e a robustez dos sistemas de deep learning em ambientes diversos. O artigo sugere melhorias contínuas no processamento multimodal e na transferência de aprendizado para futuras pes

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: 25 de set. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Alexandre Costa Ferro Filho

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

As atividades realizadas nesta semana abrangeram tanto a continuidade de tarefas iniciadas na semana anterior quanto a realização de novos estudos e análises, com foco na área de **Processamento de Áudio e Voz**:

- **Interações com colegas da área:** Discussão sobre os principais desafios e aplicações do campo, além de sugestões de inovações e possíveis abordagens futuras.
  - **Estudos em áreas de maior familiaridade:** Continuação de pesquisas em tópicos já conhecidos.
  - **Exploração de ideias inovadoras:** Foco em possíveis avanços na área de síntese e conversão de fala, considerando riscos mais elevados, pois é uma área mais complexa e menos conhecida por mim.

- **Conteúdo Didático:** Visualização de vídeos introdutórios e teóricos, abordando fundamentos de processamento de sinal de áudio aplicado a machine learning.
  - Parte da playlist do canal do Valério Velardo.

- **Levantamento bibliográfico:** Pesquisa de modelos auto-regressivos recentes voltados para síntese de fala e modelos multimodais, com ênfase em trabalhos publicados a partir de 2020.
  Autoregressive models - 16 modelos.

  Cada modelo inclui:

  - Introdução/Resumo do que foi proposto.
  - Destaque nas aplicações.
  - Artigo de referência do modelo.

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

Para próxima entrega planejo:

- Delimitar e especificar mais minha linha de estudo seguindo uma das abordagens comentadas.
- Realizar o levantamento de modelos autoregressivos com um foco maior em ASR (Reconhecimento automático de fala).

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

ACEITE DA ENTREGA:

LEONARDO ALVES: Go!

Modelos autoregressivos de Fala

Definição: são um tipo de rede neural onde a previsão da saída em um determinado passo depende das saídas anteriores. Esses modelos são amplamente utilizados em tarefas de **geração de fala, síntese de voz e transcrição automática de áudio**.

Características dos Modelos Autoregressivos

1. **Previsão Sequencial**: A natureza autoregressiva significa que esses modelos preveem cada próximo passo da sequência de fala com base nas previsões anteriores.
2. **Dependência Temporal**: São eficientes em capturar dependências temporais de longa distância, uma vez que a fala é sequencial e as palavras ou fonemas dependem de suas precedentes.

Principais Aplicações:

- **Síntese de Fala**: Geração de fala natural e expressiva a partir de texto (TTS).
- **Reconhecimento Automático de Fala**: Transcrição de fala em texto.
- **Codificação de Voz**: Compressão de sinais de fala para transmissão eficiente.

Modelos Autoregressivos Recentes na Área de Fala:

Método	Text-To-Speech	Vocoder
Modelos Autoregressivos	Flowtron, RobuTrans, Devicetts, Wave-Tacotron, Apple TTS, VALL-E, VALL-E 2, MELEE, XTTS, TorToise Mega-TTS 2.	MultiBand-WaveRNN, ImporvedLPCNet, Bunched LPCNet2

1. Flowtron (2020)

- Modelo de síntese de fala baseado em normalizing flows que aprende uma distribuição probabilística da fala e pode gerar fala de alta fidelidade e diversidade. O

uso de normalizing flows permite que o modelo seja mais flexível ao gerar fala com variações controláveis de expressividade e estilo.

- **Aplicações**: Fala controlável, ajustes finos de prosódia e estilo de fala.
- **Referência**: Flowtron: an Autoregressive Flow-based Generative Network for Text-to-Speech Synthesis.

2. RobuTrans(2020)

- Utiliza uma arquitetura transformer para síntese de fala com foco na geração de fala estável e robustas a entradas ruidosas ou com erros. Ele aplica aprendizado autorregressivo combinado com mecanismos de atenção para produzir síntese de fala clara.
- **Aplicações**: Fala assistiva, síntese robusta em dispositivos com baixa capacidade de processamento.
- **Referência**: RobuTrans: A Robust Transformer-Based Text-to-Speech Model.

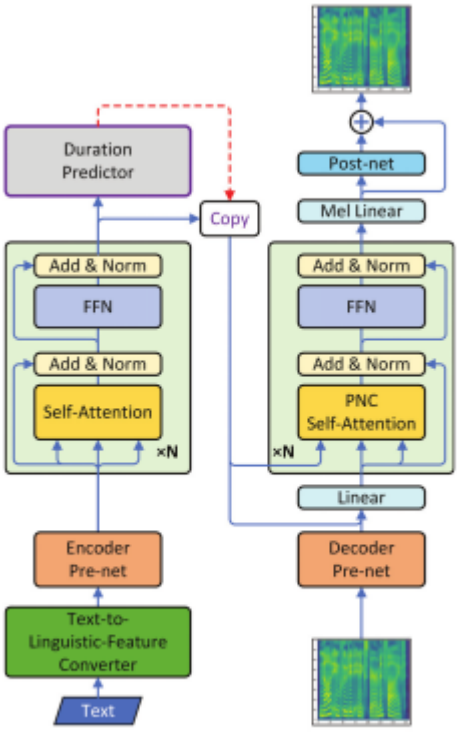
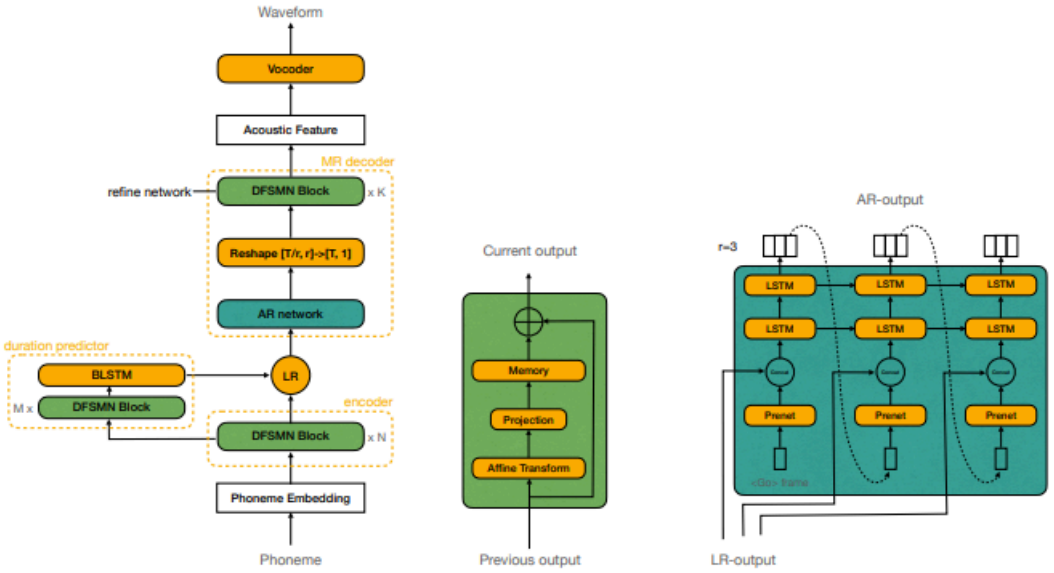


Figure 3: Architecture of RobuTrans.

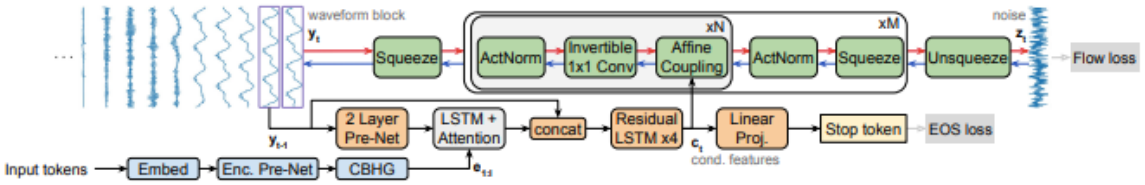
3. Devicetts(2021)

- Otimizado para rodar em dispositivos móveis e embarcados, utilizando técnicas de quantização e compactação de modelos para oferecer síntese de fala em tempo real, com uso eficiente de memória e processamento. Ele prevê frames de mel-espectrograma com alta precisão, mantendo baixa latência.
- **Aplicações**: Assistentes de voz em dispositivos de baixo consumo de energia, sintetizadores de fala portáteis.
- **Referência**: Devicetts: A small-footprint, fast, stable network for on-device text-to-speech.



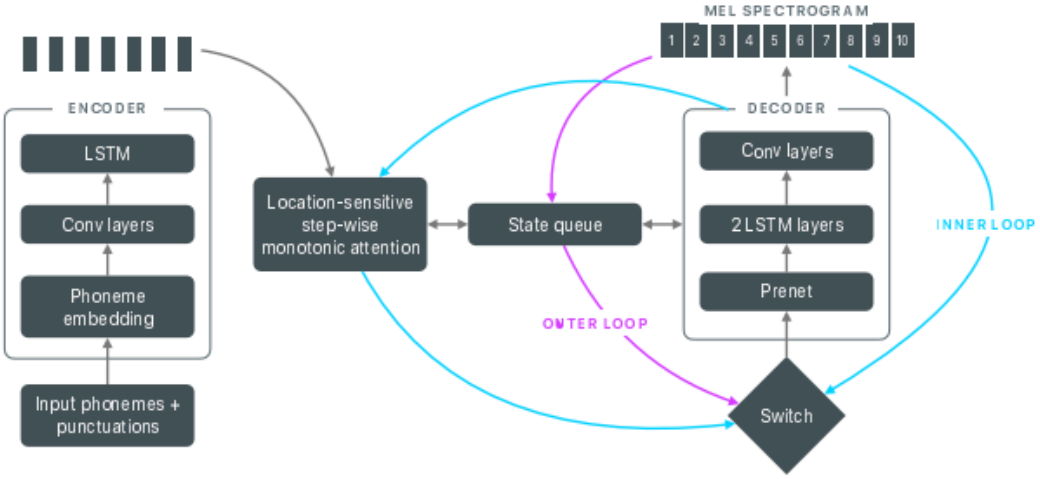
4. Wave-Tacotron(2021)

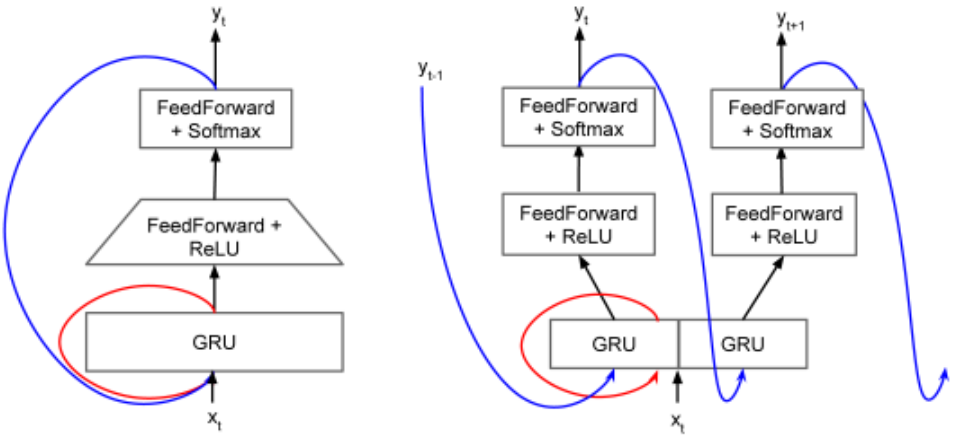
- É uma abordagem de síntese de fala end-to-end que unifica as tarefas de predição de mel-espectrograma e vocoder em uma única rede neural. Ele utiliza uma rede neural convolucional para gerar diretamente as formas de onda.
- **Aplicações**: síntese de fala altamente eficiente, síntese de fala de alta qualidade em pipelines compactos.
- **Referência**: Wave-Tacotron: Spectrogram-free end-to-end text-to-speech synthesis.



5. Apple TTS (2021)

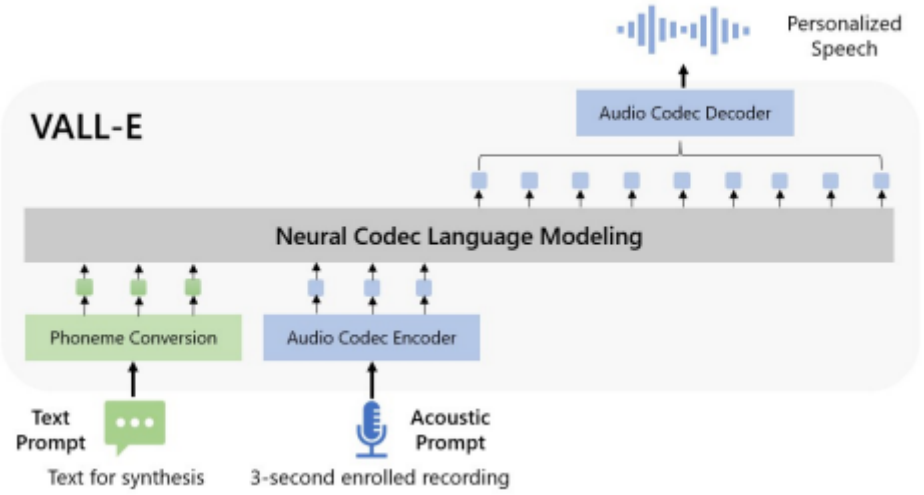
- Apresenta melhorias no modelo -Tacotron e o WaveRNN - e estratégias de otimização que permitem a implantação desses modelos em servidores GPU e em dispositivos móveis. O sistema proposto consegue gerar fala de alta qualidade a 24 kHz, operando 5 vezes mais rápido que o tempo real em servidores e 3 vezes mais rápido em dispositivos móveis.
- **Aplicações**: Assistentes virtuais, acessibilidade em dispositivos Apple, integração com Siri.
- **Referência**: [On-device neural speech synthesis.]





6. VALL-E (2023)

- Modelo de síntese de fala zero-shot que utiliza aprendizado de representação latente para prever diretamente os mel-espectrogramas com base em uma pequena amostra de voz. Ele é capaz de imitar a prosódia e o estilo de fala de um locutor a partir de poucos segundos de áudio, mantendo a naturalidade da fala gerada.
- **Aplicações**: Imitação de voz zero-shot, síntese de fala personalizada.
- **Referência**: [Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers.]



7. TorToise (2023)

- O campo da geração de imagens foi revolucionado pela aplicação de transformadores autoregressivos e Modelos de Difusão Denoising (DDPMs), que modelam o processo de geração de imagens como um processo probabilístico passo a passo, utilizando grandes quantidades de dados e poder computacional para aprender a distribuição de imagens. Apresenta uma maneira de aplicar esses avanços na geração de imagens à síntese de fala, resultando no TorToise.
- **Aplicações**: Sistema de texto para fala (TTS) multi-voz e expressivo.
- **Referência**: Better speech synthesis through scaling

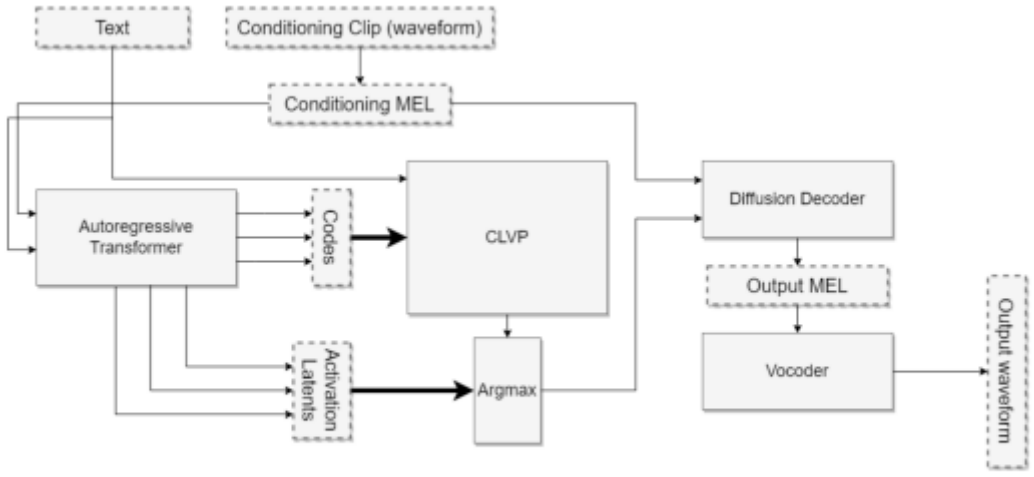
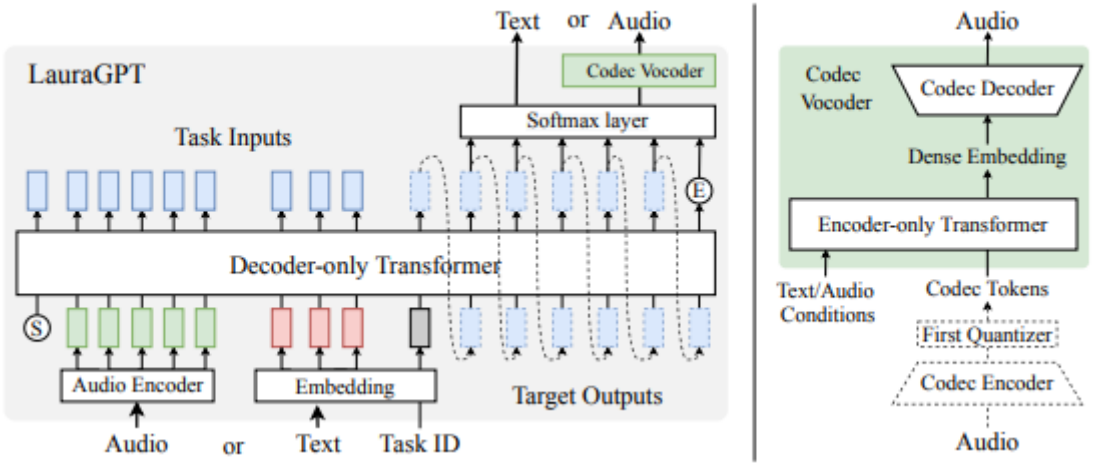


Figure 1: TorToise-v2 architectural design diagram. Inputs of text and a reference audio clip (for speaker cloning) flow through a series of decoding and filtering networks to produce high-quality speech.

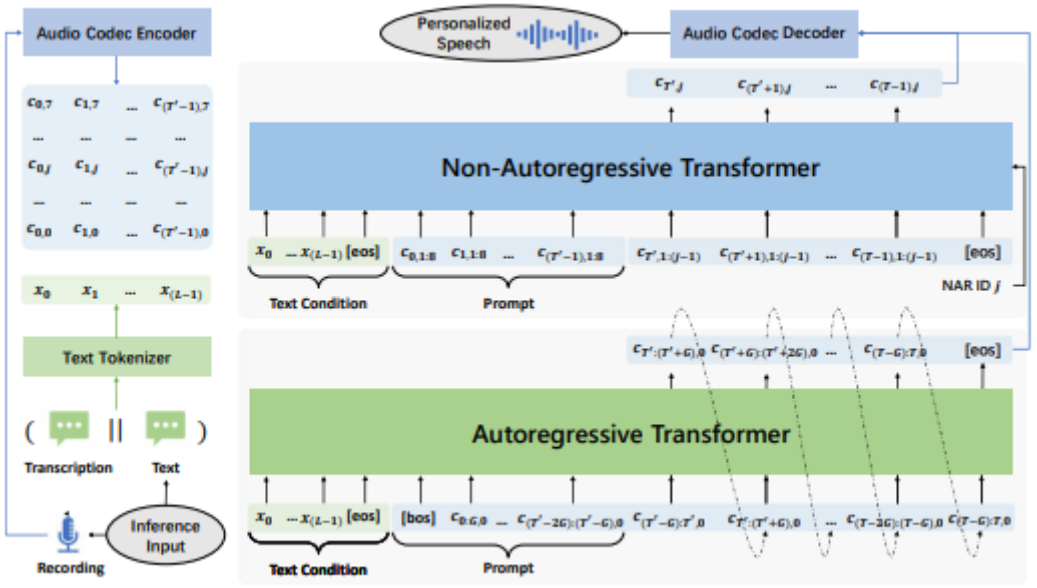
8. LauraGPT (2023)

- Modelo unificado de LLM baseado em GPT para reconhecimento, compreensão e geração de áudio. O LauraGPT combina representações contínuas e discretas, utilizando um codificador de áudio para gerar representações contínuas e um vocoder de um passo para converter essas representações em códigos discretos de áudio. Após o ajuste fino com aprendizado supervisionado de múltiplas tarefas, LauraGPT demonstra desempenho superior ou equivalente a modelos de base em uma ampla gama de tarefas de áudio, como reconhecimento automático de fala, tradução fala-para-texto, síntese de texto-para-fala, e outras tarefas relacionadas a análise de sinais de áudio.
- **Aplicações**: Reconhecimento automático de fala, Síntese de fala, Melhoria de fala.
- **Referência**: LauraGPT: Listen, Attend, Understand, and Regenerate Audio with GPT



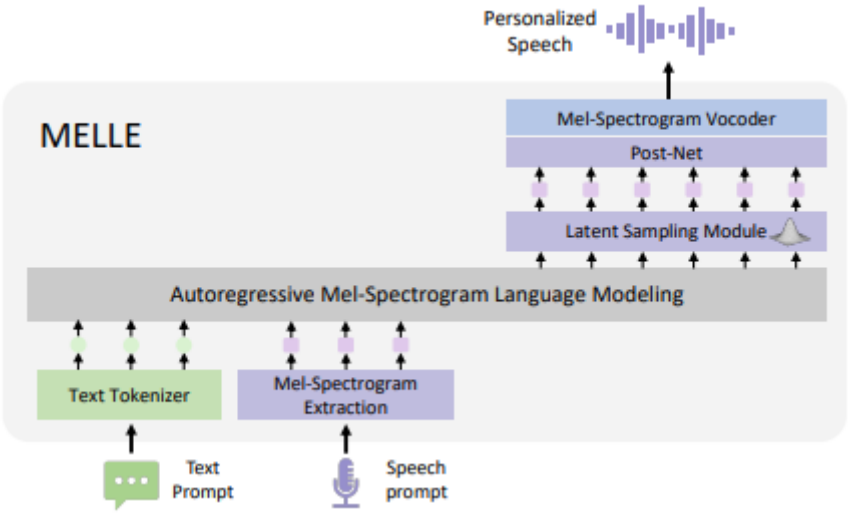
9. VALL-E 2 (2024)

- O VALL-E 2 é uma evolução do VALL-E, com melhorias no desempenho de síntese de fala zero-shot.
- **Aplicações**: Imitação de voz zero-shot, síntese de fala personalizada.
- **Referência**: VALL-E 2: Neural Codec Language Models are Human Parity Zero-Shot Text to Speech Synthesizers



10. MELLE (2024)

- Ao invés de prever códigos de codecs neurais comprimidos, o MELLE prevê diretamente os frames de mel-espectrograma, otimizados para maximizar a probabilidade de gerar fala de alta qualidade. Ele também usa amostragem latente e técnicas de refinamento para melhorar a fidelidade e naturalidade da fala gerada, integrando um vocoder para converter os mel-espectrogramas em ondas sonoras finais.
- **Aplicações**: Síntese de fala zero-shot,
- **Referência**: Autoregressive Speech Synthesis without Vector Quantization.



11. XTTS (2024)

- Baseia-se no modelo Tortoise e inclui várias modificações para permitir treinamento multilíngue, melhorar o clonagem de voz e acelerar o treinamento e a inferência. O XTTS foi treinado em 16 idiomas, alcançando resultados de estado da arte na maioria deles.
- **Aplicações**: Síntese de fala zero-shot.
- **Referência**: [XTTS: a Massively Multilingual Zero-Shot Text-to-Speech Model]

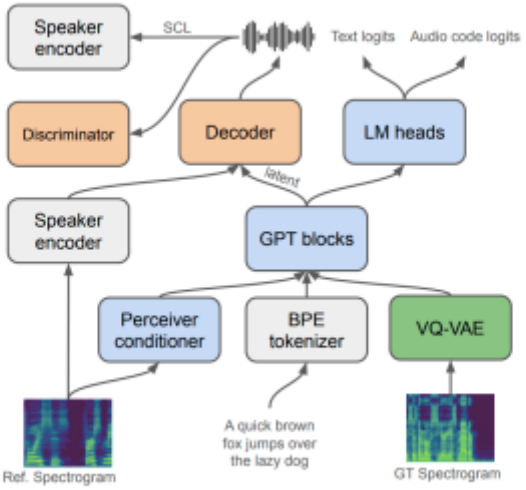


Figure 1: *XTTS training architecture overview.*

12. MEGA-TTS 2 (2024)

- Mecanismo de prompting genérico que aborda problemas (utilização de prompts de frase única que limitam o desempenho e a dificuldade em transferir informações prosódicas). A solução inclui um autoencodificador acústico que separa informações de prosódia e timbre, além de um codificador de timbre de múltiplas referências e um modelo de linguagem prosódica (P-LLM) que extrai informações de prompts de múltiplas frases.
- **Aplicações**: Síntese de fala zero-shot.
- **Referência**: MEGA-TTS 2: BOOSTING PROMPTING MECHANISMS FOR ZERO-SHOT SPEECH SYNTHESIS

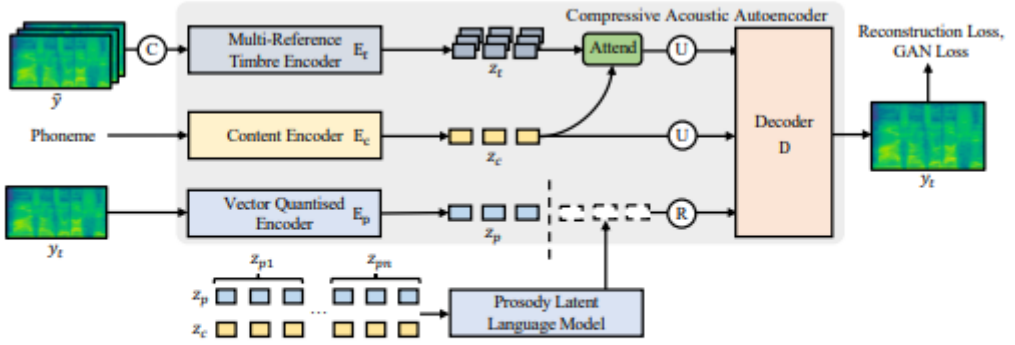


Figure 1: The overall architecture of Mega-TTS 2. Ⓒ, Ⓤ, Ⓡ denotes the concatenation, upsampling, and repeating operations, respectively. $\tilde{y}$ is concatenated along the time axis. "Attend" means the attention operation.

13. SpearTTS(2023)

- Sistema de síntese de fala (TTS) multi-falante que requer supervisão mínima para ser treinado. Ele divide o processo de TTS em duas tarefas de sequência para sequência: transformar texto em tokens semânticos de alto nível ("ler") e, em seguida, transformar esses tokens em tokens acústicos de baixo nível ("falar"). Essa separação permite treinar o módulo de "falar" com dados de áudio não rotulados, reduzindo a necessidade de dados paralelos.
- **Aplicações**: Síntese de fala multi-falante.

- **Referência**: Speak, Read and Prompt: High-Fidelity Text-to-Speech with Minimal Supervision

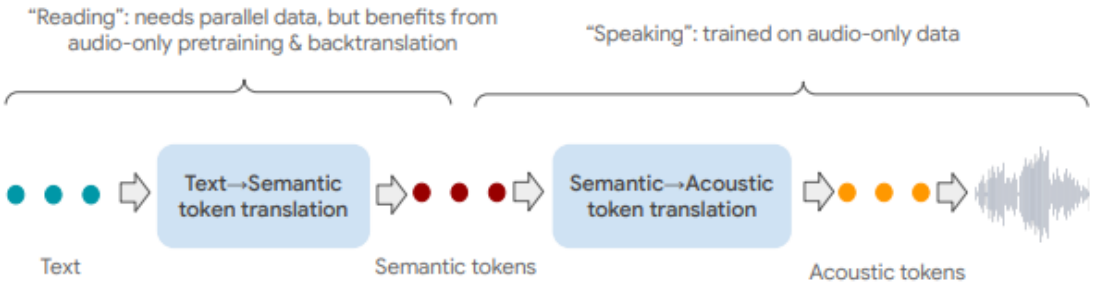
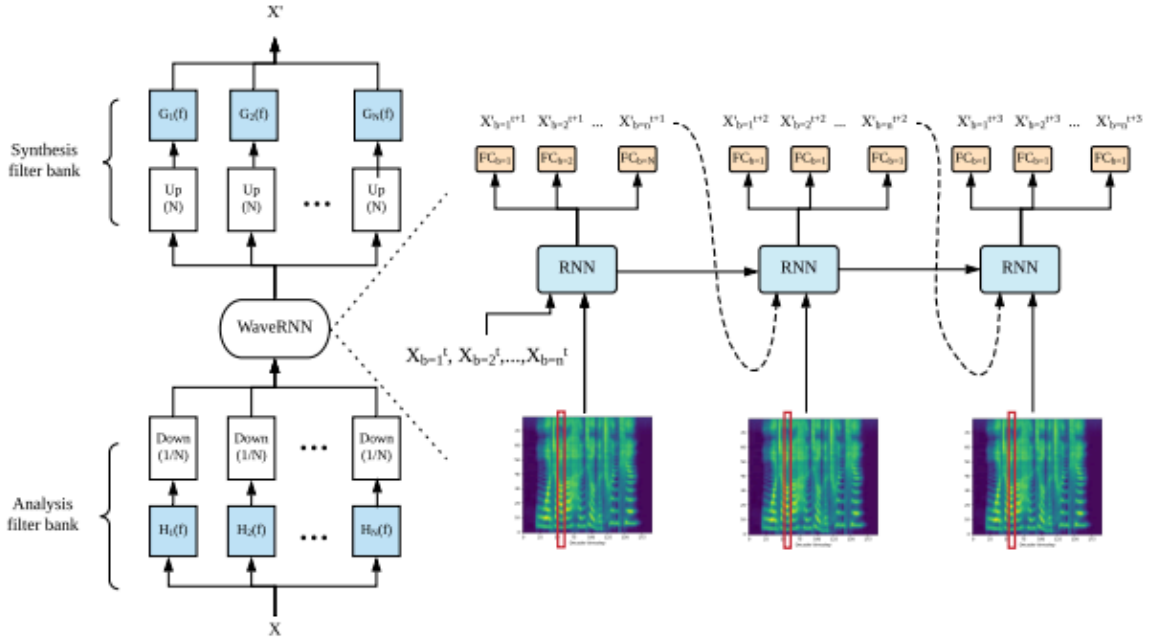


Figure 1: **SPEAR-TTS.** The first stage $\mathcal{S}_1$ ("reading") maps tokenized text to semantic tokens. The second stage $\mathcal{S}_2$ ("speaking") maps semantic tokens to acoustic tokens. Acoustic tokens are decoded to audio waveforms.

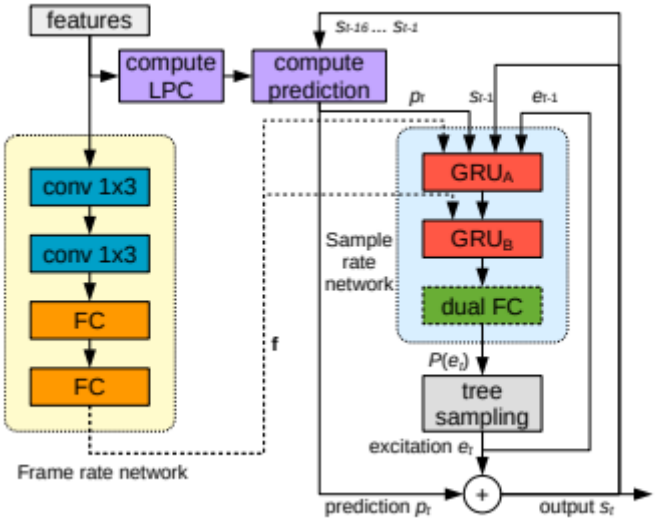
14. MULTIBAND-WAVERNN (2021)

- Uma arquitetura WaveRNN de bit fino-grosseiro para modelagem de forma de onda mu-law de 10 bits e a implementação de uma unidade recorrente gated (GRU) esparsa com um grande número de unidades ocultas para melhorar o desempenho.
- **Aplicações**: Síntese de fala, aplicações em baixa latência em tempo real:
- **Referência**: High-Fidelity and Low-Latency Universal Neural Vocoder based on Multiband WaveRNN with Data-Driven Linear Prediction for Discrete Waveform Modeling.



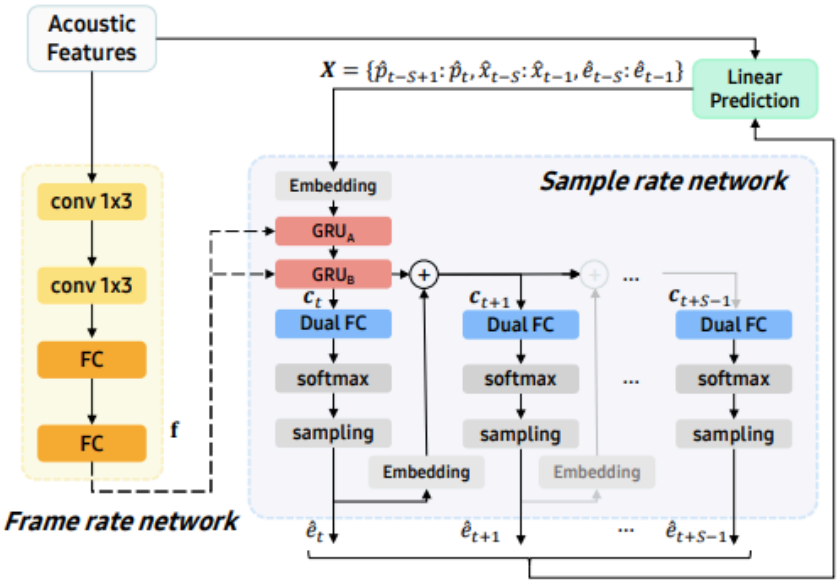
15. ImporvedLPCNet (2022)

- Apresenta melhorias no modelo LPCNet, um vocoder neural que utiliza predição linear para reduzir a complexidade computacional da síntese de fala, tornando-o mais acessível a uma variedade de dispositivos. As novas otimizações visam melhorar a eficiência algorítmica e computacional, permitindo que o LPCNet opere 2,5 vezes mais rápido e com qualidade de síntese aprimorada.
- **Aplicações**: Síntese de fala, aplicações em baixa latência em tempo real.
- **Referência**: Neural speech synthesis on a shoestring: Improving the efficiency of lpcnet.



16. Bunched LPCNet2 (2022)

- Uma arquitetura aprimorada do LPCNet, que visa oferecer desempenho eficiente em alta qualidade para servidores em nuvem e baixa complexidade para dispositivos de recursos limitados. Essa nova abordagem utiliza uma distribuição logística única para melhorar a eficiência computacional e técnicas que reduzem o tamanho do modelo sem comprometer a qualidade da fala. Três principais contribuições: uma camada de saída logística única, uma abordagem de taxa dupla e técnicas que minimizam a pegada do modelo.
- **Aplicações**: Síntese de fala, aplicações em baixa latência em tempo real.
- **Referência**: Bunched LPCNet2: Efficient Neural Vocoders Covering Devices from Cloud to Edge



Desafios dos Modelos Autoregressivos:

- **Latência**: Como cada passo depende dos anteriores, esses modelos podem ter um tempo de inferência maior, o que os torna menos adequados para aplicações em tempo real.
- **Instabilidade**: Erros pequenos em previsões iniciais podem se amplificar, levando a uma degradação da qualidade do áudio final.

Falta leitura:
AudioLM: a Language Modeling Approach to Audio Generation

AUDIOGEN: TEXTUALLY GUIDED AUDIO GENERATION

Speak Foreign Languages with Your Own Voice: Cross-Lingual Neural Codec Language Modeling - Vall-e x

VioLA: Unified Codec Language Models for Speech Recognition, Synthesis, and Translation

SpeechX: Neural Codec Language Model as a Versatile Speech Transformer

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: 3 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Alexandre Costa Ferro Filho

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

As atividades realizadas nesta semana abrangeram a realização de novos estudos e análises, com foco na área de **Processamento de Áudio e Voz**:

- **Área foco dos próximos estudos:** Delimitação dos principal foco dos meus estudos dentro da área de PAV:
  - Síntese e conversão de voz - áreas pouco estudadas por mim e que eu percebo cada vez inovações e diferentes abordagens, logo maior área para estudo.

- **Levantamento bibliográfico:** Pesquisa de modelos recentes voltados para Reconhecimento Automático da Fala, com ênfase em trabalhos publicados a partir de 2019. ASR - Models - 14 modelos.

  Cada modelo inclui:

  - Introdução/Resumo do que foi proposto.
  - Destaque nas aplicações.
  - Artigo de referência do modelo.

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

Para próxima Entrega planejo:

- Realizar estudo mais aprofundado das diferentes formas de representação do áudio e texto nos principais modelos levantados nesse e no Stage anterior.
- Fazer levantamento superficial e mais recente nas áreas que faltaram: reconhecimento de falante, classificação e reconhecimento de atividade vocal.

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Modelos Reconhecimento Automático de Fala

Definição: São sistemas de inteligência artificial que convertem a fala em texto. Eles processam sinais de áudio e utilizam técnicas de aprendizado de máquina, como redes neurais, para transcrever palavras faladas. Esses modelos são amplamente usados em assistentes de voz, sistemas de transcrição e interfaces controladas por fala.

Características dos Modelos Autoregressivos

1. **Processamento Sequencial:** A fala é um sinal temporal contínuo, e os modelos de ASR precisam segmentar e transcrever em unidades discretas, como palavras ou fonemas.
2. **Robustez em Cenários Variados:** Modelos de ASR devem ser capazes de lidar com variações na pronúncia, sotaques, ruídos de fundo e diferentes estilos de fala.
3. **Treinamento com Grandes Quantidades de Dados:** Eles são geralmente treinados com grandes volumes de dados de áudio para melhorar sua precisão e adaptação a diferentes contextos e linguagens.

Principais Aplicações:

- **Processamento Sequencial:** A fala é um sinal temporal contínuo, e os modelos de ASR precisam segmentar e transcrever em unidades discretas, como palavras ou fonemas.
- **Robustez em Cenários Variados:** Modelos de ASR devem ser capazes de lidar com variações na pronúncia, sotaques, ruídos de fundo e diferentes estilos de fala.
- **Treinamento com Grandes Quantidades de Dados:** Eles são geralmente treinados com grandes volumes de dados de áudio para melhorar sua precisão e adaptação a diferentes contextos e linguagens.

Modelos Recentes na Área de ASR:

Método	ASR
Modelos Autoregressivos	Whisper, Distil-Whisper, Transformer Transducer, XEUS
Modelos Não-Autoregressivos	Conformer, Wav2vec 2.0, ContextNet, ConMamba, Fast Conformer, Canary, SEW, HuBERT, WavLM, Citrinet, Jasper

1. Transformer Transducer (2020)

- Baseado em codificadores Transformer que pode ser utilizado em sistemas de reconhecimento de fala em tempo real. Ele combina informações de áudio e rótulos com autoatenção e uma camada feed-forward. O modelo foi treinado para ser adequado ao streaming, utilizando uma perda RNN-T que mantém a computação constante por quadro, tornando a decodificação prática.
- **Aplicações**: Transcrição de áudio em tempo real ou de alta precisão.
- **Referência**: Transformer Transducer: A Streamable Speech Recognition Model with Transformer Encoders and RNN-T Loss.

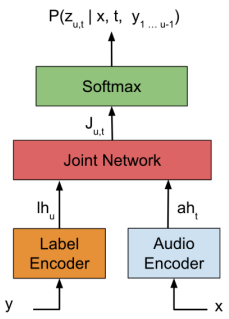


Fig. 1. RNN/Transformer Transducer architecture.

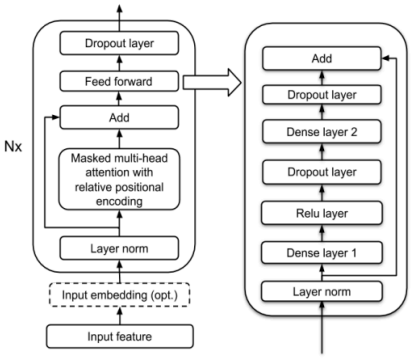
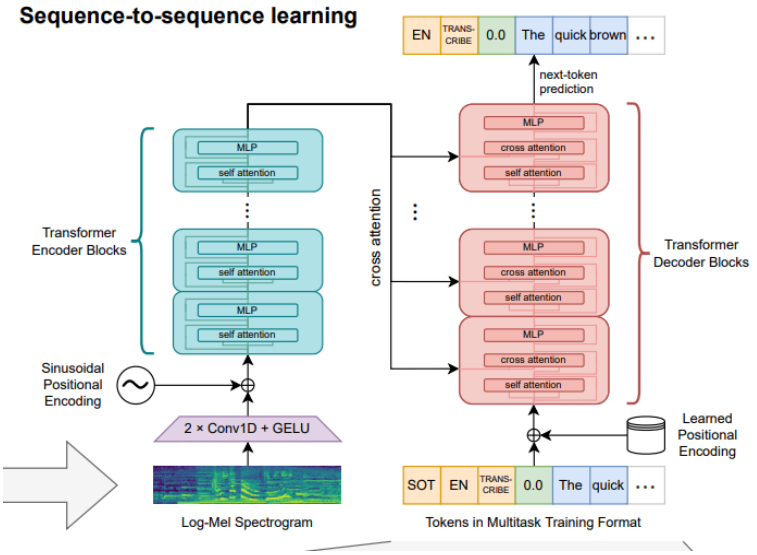


Fig. 2. Transformer encoder architecture.

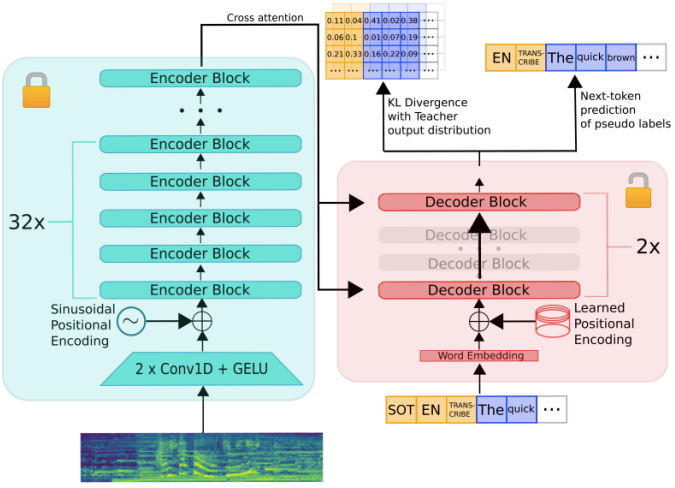
2. Whisper (2022)

- Com 680.000 horas de supervisão multilíngue e multitarefa, os modelos resultantes generalizam bem para benchmarks padrão e, em muitos casos, são competitivos com resultados supervisionados anteriores, mas sem necessidade de ajuste fino. Em comparação aos humanos, os modelos alcançam precisão e robustez similares. O trabalho foca em ampliar a escala de pré-treinamento fraco e multilíngue para reconhecimento de fala, sem utilizar técnicas de auto-supervisão. Modelo segue a arquitetura encoder-decoder.
- **Aplicações**: Transcrição de fala para texto com altíssima qualidade
- **Referência**: Robust Speech Recognition via Large-Scale Weak Supervision.



3. Distil-Whisper (2023)

- Uma versão reduzida do modelo Whisper, visando solucionar os desafios de implantar grandes modelos de reconhecimento de fala em ambientes com restrições de recursos e baixa latência. Utilizando pseudo-rotulagem e um critério baseado em WER, o modelo destilado tem 51% menos parâmetros e é 5,8 vezes mais rápido, mantendo até 1% de WER em dados fora da distribuição. Ele é mais robusto a condições acústicas adversas e menos suscetível a erros de alucinação em áudios longos.
- **Aplicações**: Transcrição de fala para texto com altíssima qualidade com um tempo de latência reduzido.
- **Referência**: Distil-Whisper: Robust Knowledge Distillation via Large-Scale Pseudo Labelling.



4. XEUS (2024)

- Cross-lingual Encoder for Universal Speech é um modelo de aprendizado auto-supervisionado projetado para expandir a cobertura linguística em tecnologias de fala, treinado em mais de 1 milhão de horas de dados em 4.057 idiomas. Esse modelo combina um grande conjunto de dados existente com um novo corpus de 7.400 horas, abrangendo uma variedade de estilos de fala e condições de gravação. Para aumentar a robustez do XEUS, foi introduzido um novo objetivo de aprendizado que visa a desreverberação acústica, ajudando o modelo a lidar com gravações em ambientes desafiadores. O XEUS supera ou iguala o desempenho de modelos SSL de última geração em várias tarefas, incluindo reconhecimento de fala, tradução de fala e resíntese de fala.
- **Aplicações**: Transcrição de fala para texto, Cobertura Linguística Ampliada, Auto-supervisionado.
- **Referência**: Towards Robust Speech Representation Learning for Thousands of Languages.

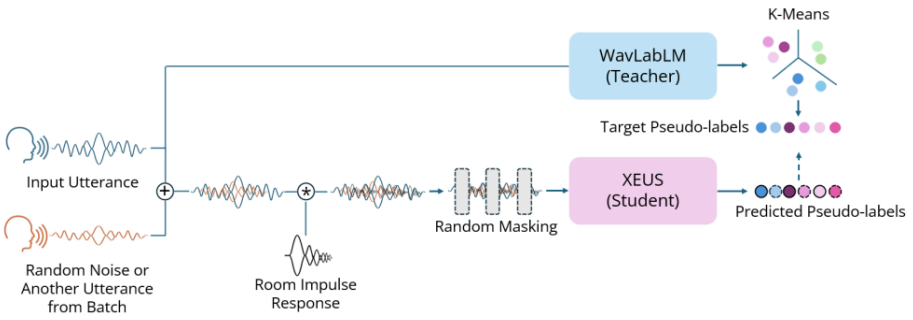


Figure 2: Overview of XEUS' pre-training. The teacher encoder generates phonetic pseudo-labels from clean speech, while the student must predict those pseudo-labels after masking, random noise and/or reverberation is applied to the input waveform.

5. Jasper (2019)

- O modelo utiliza apenas convoluções 1D, normalização em lote, ReLU, dropout e conexões residuais. Para melhorar o treinamento, foi introduzido o otimizador **NovoGrad**. A versão mais profunda do Jasper, com 54 camadas de convolução, obteve uma taxa de erro de palavras (WER) de 2,95% com um decodificador de busca em feixe e 3,86% com um decodificador guloso nos dados de teste do LibriSpeech.
- **Aplicações**: Transcrição de fala para texto End-to-End, flexibilidade.
- **Referência**: Jasper: An End-to-End Convolutional Neural Acoustic Model

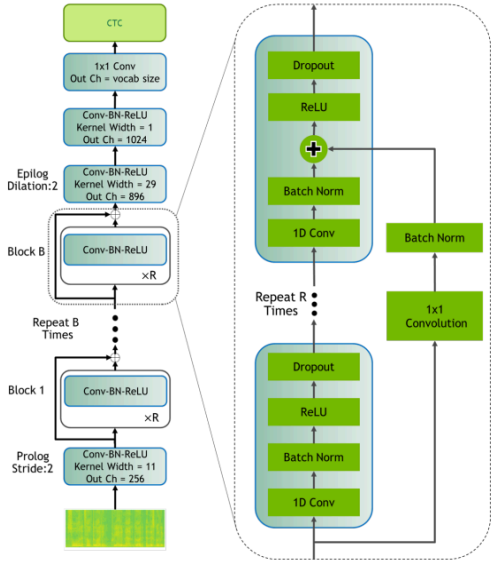


Figure 1: *Jasper BxR model: B - number of blocks, R - number of sub-blocks.*

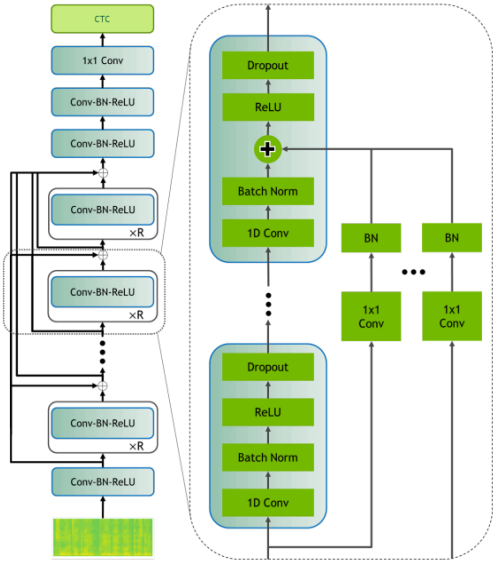


Figure 2: *Jasper Dense Residual*

6. ContextNet (2020)

- Combina redes neurais convolucionais (CNN) e redes recorrentes (RNN), visando superar a performance de modelos baseados em RNN/transformers. Sua principal inovação é a incorporação de módulos de **squeeze-and-excitation (SE)**, que permitem que as camadas convolucionais acessem informações de contexto global, melhorando a capacidade de compreensão do modelo.
- **Aplicações**: Transcrição de fala para texto, Acesso a Contexto Global, Escalabilidade.
- **Referência**: [ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context.]

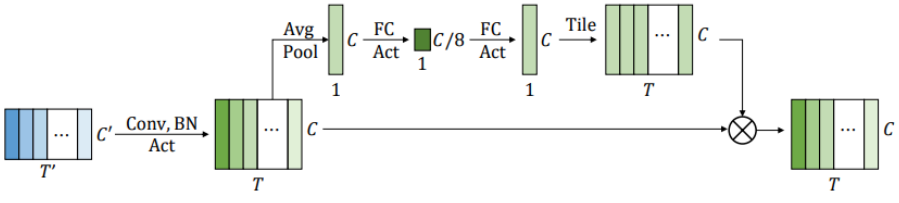


Figure 2: *1D Squeeze-and-excitation module. The input first goes through a convolution layer followed by batch normalization and activation. Then average pooling is applied to condense the conv result into a 1D vector, which is then processed by a bottleneck structure formed by two fully connected (FC) layers with activation functions. The output goes through a Sigmoid function to be mapped to* $(0, 1)$*, and then tiled and applied on the conv output using pointwise multiplications.*

7. Conformer (2020)

- Modelo proposto que combina redes neurais convolucionais (CNN) e Transformers para capturar tanto dependências locais quanto globais de sequências de áudio de maneira eficiente em termos de parâmetros. Este modelo supera significativamente modelos anteriores baseados em Transformers e CNNs, alcançando a precisão de estado da arte no benchmark LibriSpeech.
- **Aplicações**: Transcrição de fala para texto com boa performance e bom tempo de inferência.
- **Referência**: Conformer: Convolution-augmented Transformer for Speech Recognition.

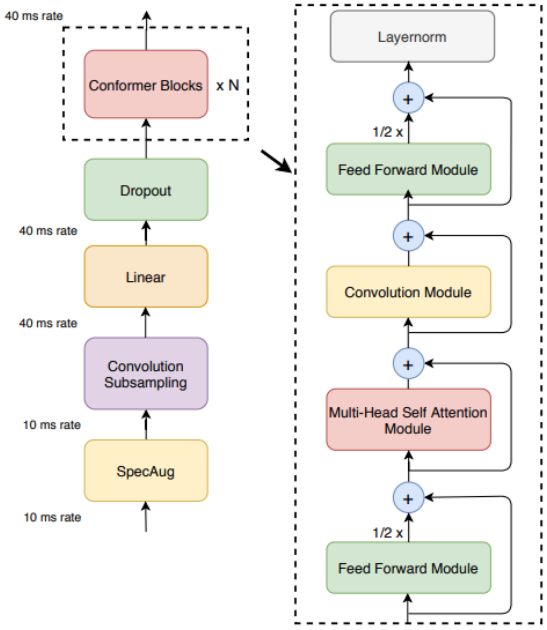


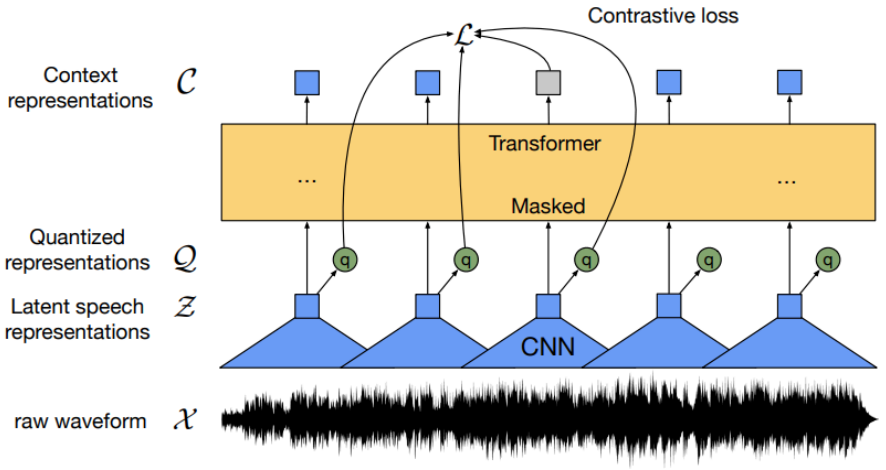
Figure 1: ***Conformer encoder model architecture.*** *Conformer comprises of two macaron-like feed-forward layers with half-step residual connections sandwiching the multi-headed self-attention and convolution modules. This is followed by a post layernorm.*

8. Wav2vec 2.0 (2020)

- Um modelo de aprendizado auto-supervisionado que utiliza apenas áudio de fala para aprender representações poderosas, seguido de um ajuste fino com dados transcritos. Os resultados demonstram que, com uma abordagem conceitualmente mais simples, o wav2vec 2.0 supera os melhores métodos semi-supervisionados. Ótimo desempenho com dados limitados e utiliza de CNNs para extração de

características, masking para aprendizado robusto, e um Transformador para construir representações contextualizadas.

- **Aplicações**: Transcrição de fala para texto, Baixa Dependência de Dados Rotulados, Auto-supervisionado, Baixa latência.
- **Referência**: [wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations]



9. SEW (2021)

- Um modelo de aprendizado auto-supervisionado que utiliza apenas áudio de fala para aprender representações poderosas, seguido de um ajuste fino com dados transcritos. Os resultados demonstram que, com uma abordagem conceitualmente mais simples, o wav2vec 2.0 supera os melhores métodos semi-supervisionados. Ótimo desempenho com dados limitados e utiliza de CNNs para extração de características, masking para aprendizado robusto, e um Transformador para construir representações contextualizadas.
- **Aplicações**: Transcrição de fala para texto, Eficiência de Inferência, Modelos Menores, Baixa latência.
- **Referência**: [Performance-Efficiency Trade-offs in Unsupervised Pre-training for Speech Recognition]

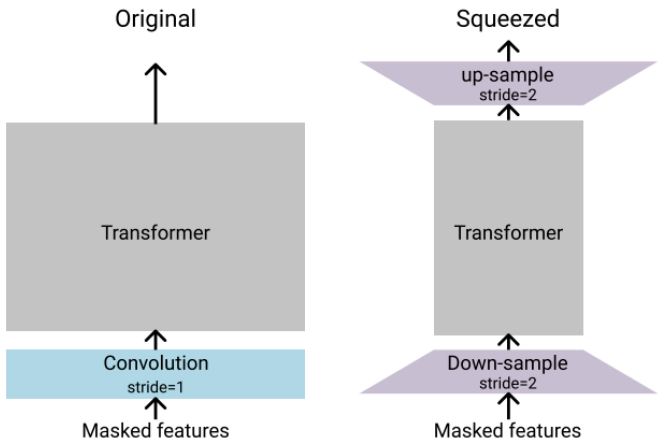
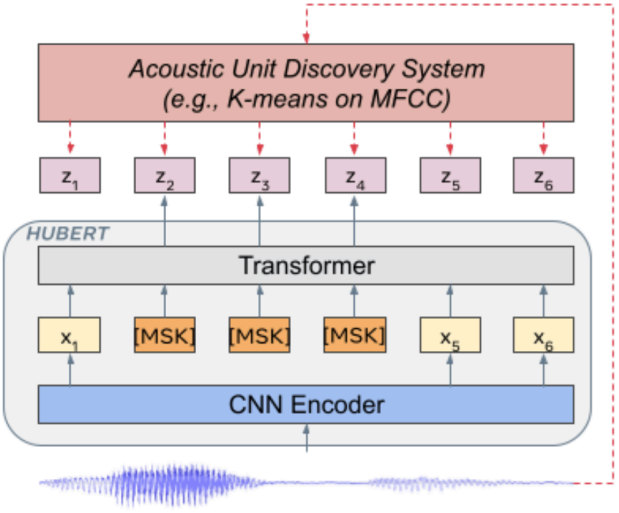


Figure 4: Original vs. squeezed context network. The sequence length is halved by the down-sampling layer.

10. HuBERT (2021)

- O **Hidden-Unit BERT** tem como foco aprendizado de representações de fala por meio de aprendizado auto-supervisionado. Visa lidar com três problemas únicos na representação de fala: múlti unidades sonoras em uma única entrada, a falta de um léxico durante a fase de pré-treinamento, e o comprimento variável das unidades sonoras sem segmentação explícita. A técnica envolve uma etapa de agrupamento offline para gerar rótulos alinhados e aplica uma perda de predição apenas nas regiões mascaradas, forçando o modelo a aprender uma combinação de modelos acústicos e linguísticos. Desempenho que iguala ou supera o estado da arte do **wav2vec 2.0.**
- **Aplicações**: Transcrição de fala, Auto-supervisionado.
- **Referência**: HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units.



11. WavLM (2021)

- Utiliza aprendizado auto-supervisionado para abordar uma variedade de tarefas de processamento de fala. O modelo combina predição de fala mascarada e denoising durante o pré-treinamento, permitindo que aprenda tanto a modelagem do conteúdo da fala quanto a capacidade de lidar com tarefas não relacionadas a ASR. O WavLM utiliza um viés de posição relativa em sua estrutura Transformer, melhorando a captura da ordem da sequência de entrada. Supera limitações encontradas em modelos anteriores, como o HuBERT e o wav2vec 2.0, em diversas tarefas representativas, oferecendo resultados mais robustos e generalizados.
- **Aplicações**: Transcrição de fala para texto, maior robustez e generalização.
- **Referência**: [WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing.]

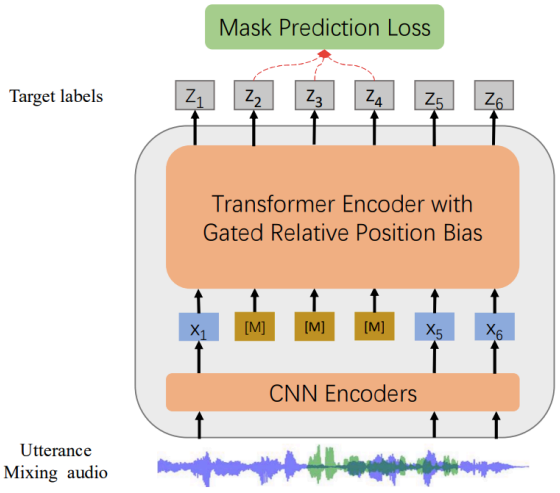


Fig. 1. Model Architecture.

12. Fast Conformer (2023)

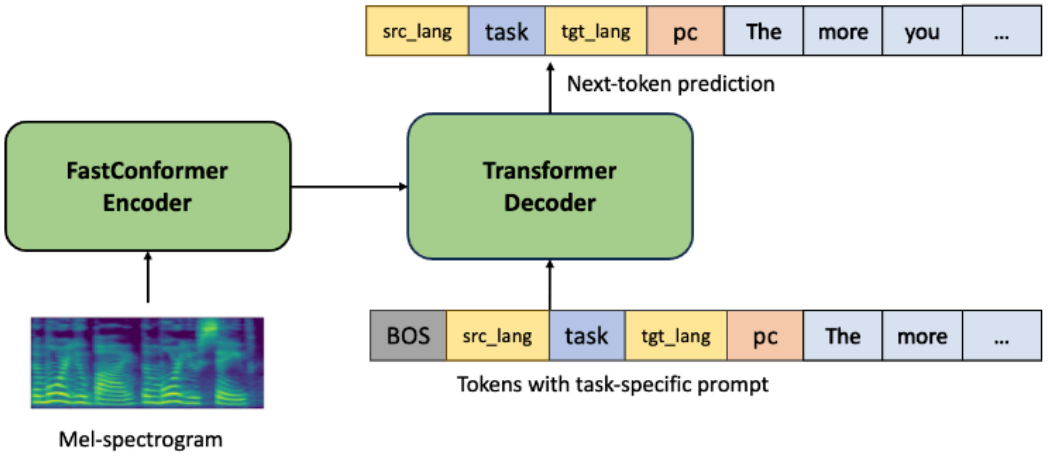
- Versão otimizada da arquitetura Conformer, projetada para melhorar a eficiência no treinamento e na inferência em tarefas de processamento de fala. O modelo é 2,8 vezes mais rápido que o Conformer original, suporta escalabilidade de até 1 bilhão de parâmetros sem alterações na arquitetura central e mantém uma precisão de ponta em benchmarks de Reconhecimento Automático de Fala (ASR). Para transcrever fala de longa duração (até 11 horas), o Fast Conformer substitui a atenção global por uma atenção de contexto limitada, melhorando a precisão por meio de um ajuste fino com a adição de um token global.
- **Aplicações**: Transcrição de fala para texto, aplicações em baixa latência/streaming.
- **Referência**: Fast Conformer with Linearly Scalable Attention for Efficient Speech Recognition

13. Canary (2024)

- Demonstra que é possível alcançar alta precisão sem depender de grandes volumes de dados da Internet. O Canary, que é um modelo multilingue, supera modelos como Whisper, OWSM e Seamless-M4T em inglês, francês, espanhol e alemão, sendo treinado com uma quantidade de dados significativamente menor. Os três principais fatores que possibilitam essa eficiência são: uma arquitetura de encoder-decoder baseada em FastConformer, treinamento em dados sintéticos gerados por meio de tradução automática e técnicas avançadas de treinamento, como balanceamento de

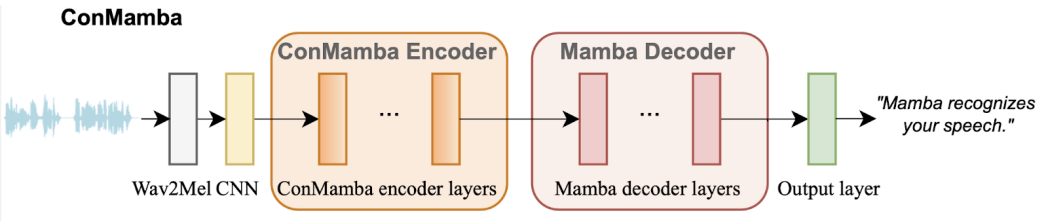
dados, mistura dinâmica de dados, agrupamento dinâmico e ajuste fino robusto ao ruído.

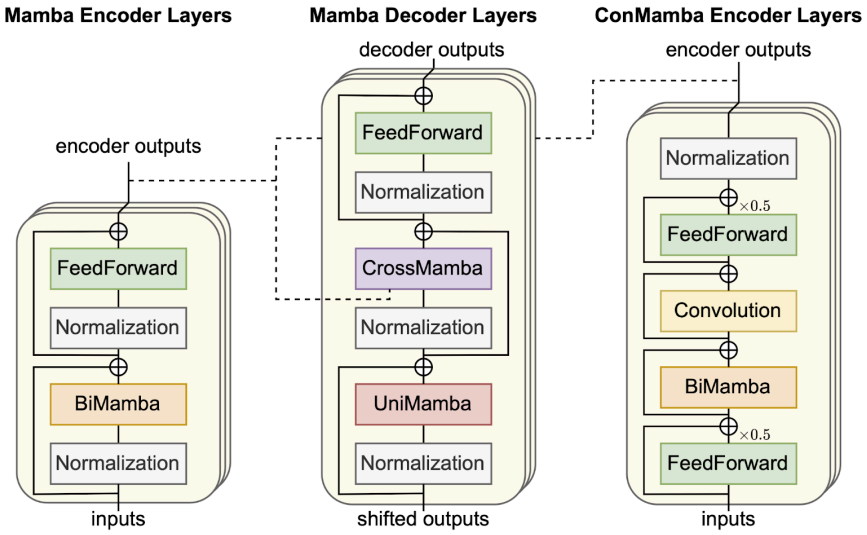
- **Aplicações**: Transcrição de fala para texto com altíssima qualidade, aplicações com menor dependência de dados, uso de dados sintéticos.
- **Referência**: Less is More: Accurate Speech Recognition & Translation without Web-Scale Data.



14. ConMamba (2024)

- Mamba, um modelo baseado em espaço de estado, apresenta uma complexidade linear em relação ao comprimento do token, o que o torna mais eficiente em termos de memória e velocidade, especialmente para sequências de fala mais longas. Mamba não se destaca em todas as situações; para trechos de fala mais curtos, sua eficiência é semelhante à dos transformadores. Os resultados indicam que a superioridade de Mamba depende das características das tarefas e dos modelos utilizados.
- **Aplicações**: Transcrição de fala para texto com longas sequências.
- **Referência**: Speech Slytherin: Examining the Performance and Efficiency of Mamba for Speech Separation, Recognition, and Synthesis.




Mamba Encoder Layers
Mamba Decoder Layers
ConMamba Encoder Layers
decoder outputs
encoder outputs
encoder outputs
FeedForward
Normalization
CrossMamba
Normalization
UniMamba
Normalization
Normalization
×0.5
FeedForward
Convolution
BiMamba
×0.5
FeedForward
FeedForward
Normalization
BiMamba
Normalization
inputs
shifted outputs
inputs

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]:

Alexandre Costa Ferro Filho

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

As atividades realizadas nesta semana abrangeram a realização de novos estudos e análises, com foco na área de **Processamento de Áudio e Voz**:

- **Levantamento bibliográfico:** Pesquisa de modelos recentes voltados para algumas outras áreas de aplicação, com ênfase em trabalhos publicados a partir de 2020. Other Taks - Models - 11 modelos.

  Cada modelo inclui:

  - Introdução/Resumo do que foi proposto.
  - Destaque nas aplicações.
  - Artigo de referência do modelo.

- **Estudo formas de representações recentes do áudio:** Pesquisa de formas de representações do áudio usadas em trabalhos recentes, principalmente na área de síntese de fala e multimodais.

  - Estudo sobre Audio Neural CODECs
    - Material: Audio Codecs & the AI Revolution - An Introduction to Machine Lea…
  - Leitura completa do artigo VALL-E
    - Resumo: VALL-E

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

Para próxima Entrega planejo:

- Continuar estudo mais aprofundado das diferentes formas de representação do áudio e texto nos principais modelos.
- Estudo de frameworks relacionados com Deep Learning e áudio:
  - Librosa - Biblioteca de análise e transformações de áudios.

- Torchaudio - Extensão do PyTorch para processamento de áudio, focada em suporte a modelos de aprendizado profundo e manipulação eficiente de dados de áudio.
- NeMo - Toolkit da NVIDIA para construir modelos de diversas áreas.
- Espnet - Framework para pesquisa e desenvolvimento de modelos de processamento de fala.

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

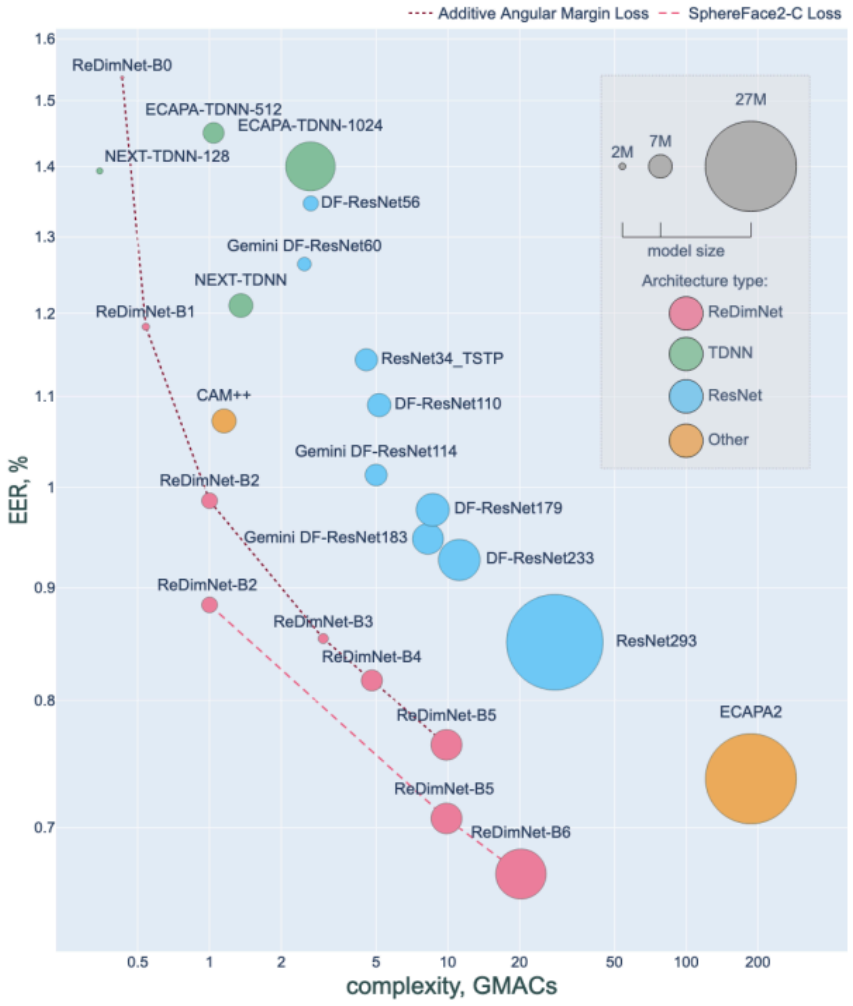
Modelos Recente PAV

Speaker recognition:

Área de estudo da inteligência artificial que se concentra na identificação ou verificação de pessoas com base em suas características vocais. Ele se divide em dois principais subtipos:

1. **Identificação de falante:** Determina a identidade de um falante entre um grupo conhecido de falantes.
2. **Verificação de falante:** Verifica se a pessoa que está falando é quem ela afirma ser.

`Comparação de alguns modelos:`



1. Titanet (2021)

- Sua arquitetura utiliza camadas convolucionais e mecanismos de atenção, permitindo capturar características temporais e espectrais da fala de forma eficiente. Ele se destaca pelo uso de mecanismos de atenção adaptativa, o que o torna robusto em ambientes ruidosos e em condições variáveis de gravação.
- **Aplicações**: Verificação de locutor em larga escala e em ambientes com múltiplas fontes de ruído.
- **Referência**: TitaNet: Neural Model for speaker representation with 1D Depth-wise separable convolutions and global context

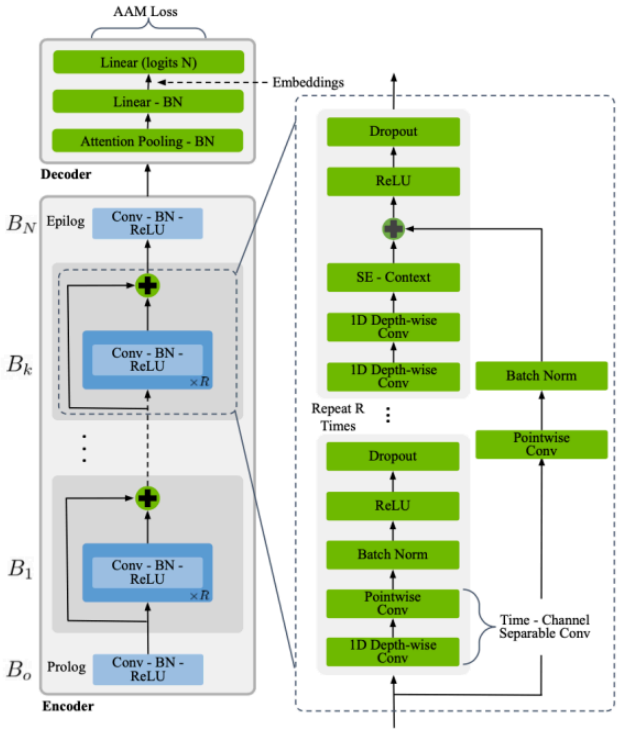


Fig. 1. TitaNet Encoder and Decoder Architecture

2. ECAPA-TDNN (2020)

- O ECAPA-TDNN - Emphasized Channel Attention Propagation and Aggregation Time Delay Neural Network - é uma versão estendida da arquitetura ECAPA, que adiciona o uso de redes neurais de atraso temporal. O diferencial desse modelo é sua

habilidade de modelar informações temporais com alta precisão ao mesmo tempo em que aplica atenção a diferentes canais de áudio. Isso resulta em melhor desempenho em tarefas de reconhecimento de fala contínua e em ambientes com ruído.

- **Aplicações**: Verificação de locutor com alto desempenho.
- **Referência**: ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification

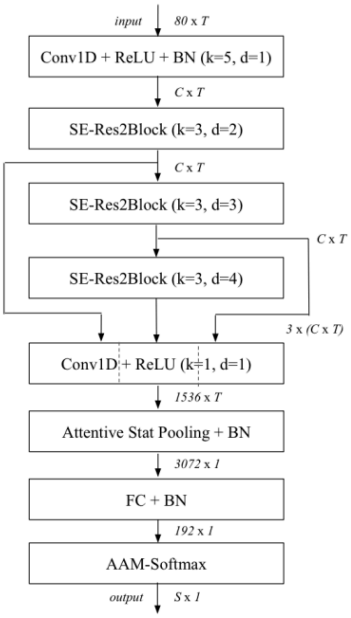


Figure 2: *Network topology of the ECAPA-TDNN. We denote k for kernel size and d for dilation spacing of the Conv1D layers or SE-Res2Blocks. C and T correspond to the channel and temporal dimension of the intermediate feature-maps respectively. S is the number of training speakers.*

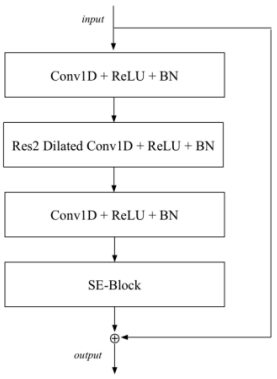


Figure 1: *The SE-Res2Block of the ECAPA-TDNN architecture. The standard Conv1D layers have a kernel size of 1. The central Res2Net [16] Conv1D with scale dimension $s = 8$ expands the temporal context through kernel size k and dilation spacing d.*

3. SKA-TDNN (2022)

- Sua arquitetura inovadora que combina mecanismos de atenção com redes TDNN. A característica diferencial deste modelo é o uso de núcleos seletivos que ajustam dinamicamente as escalas das características temporais.
- **Aplicações**: Verificação de locutor em larga escala e em ambientes com múltiplas fontes de ruído.
- **Referência**: Frequency and Multi-Scale Selective Kernel Attention for Speaker Verification.

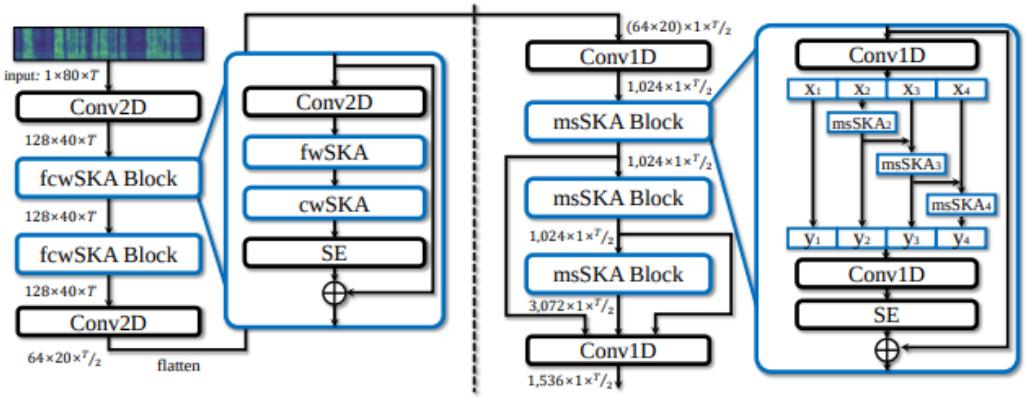
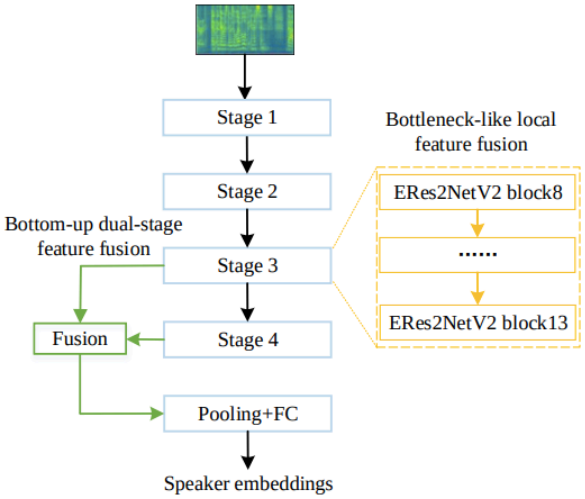


Fig. 3: The overall proposed architecture: The frequency-channel-wise SKA block-based front network (left) and the multi-scale SKA block-based TDNN network (right). This architecture is referred to SKA-TDNN.

4. ERes2NetV2 (2024)

- Uma evolução da arquitetura Res2Net, projetada especificamente para tarefas de reconhecimento de fala e análise de áudio. Ele usa um sistema de blocos residuais escalonáveis que permitem que o modelo processe informações em múltiplas resoluções de tempo simultaneamente.
- **Aplicações**: Verificação de locutor em larga escala e em ambientes com múltiplas fontes de ruído.
- **Referência**: [ERes2NetV2: Boosting Short-Duration Speaker Verification Performance with Computational Efficiency]



5. ECAPA2 (2024)

- Evolução da arquitetura ECAPA-TDNN, com foco em melhorar ainda mais o desempenho em reconhecimento de locutor e classificação de áudio. ECAPA significa "Emphasized Channel Attention, Propagation and Aggregation", uma técnica que enfatiza a atenção entre os canais e melhora a propagação das informações no modelo. O ECAPA2 mantém essa base, mas introduz novos blocos de propagação e mecanismos de agregação refinados, permitindo uma latência mais baixa e uma precisão superior em ambientes ruidosos.
- **Aplicações**: Verificação de locutor em larga escala e em ambientes com múltiplas fontes de ruído.
- **Referência**: ECAPA2: A Hybrid Neural Network Architecture and Training Strategy for Robust Speaker Embeddings

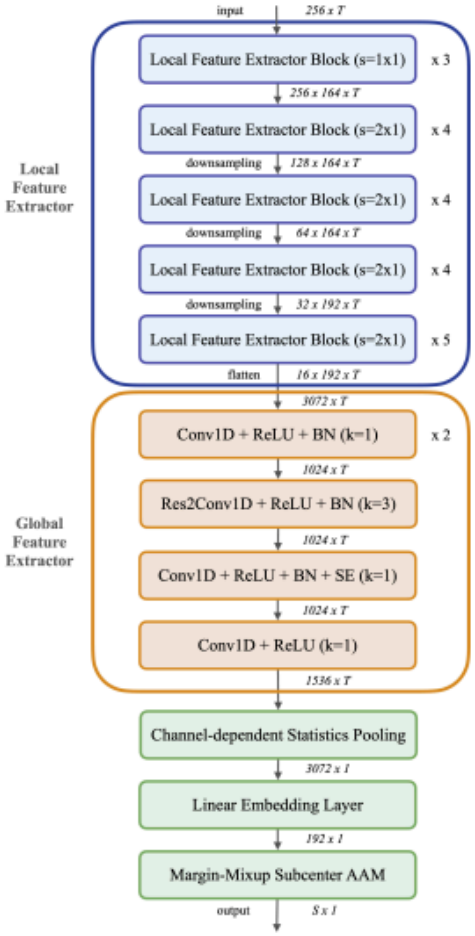


Fig. 5. Topology of the ECAPA2 architecture. T denotes the number of temporal frame-level features and k indicates kernel size.

Classificação de Áudio:

É uma tarefa dentro do campo da inteligência artificial que envolve a identificação e categorização de sons com base em suas características. A classificação de áudio pode ser dividida em várias categorias, incluindo:

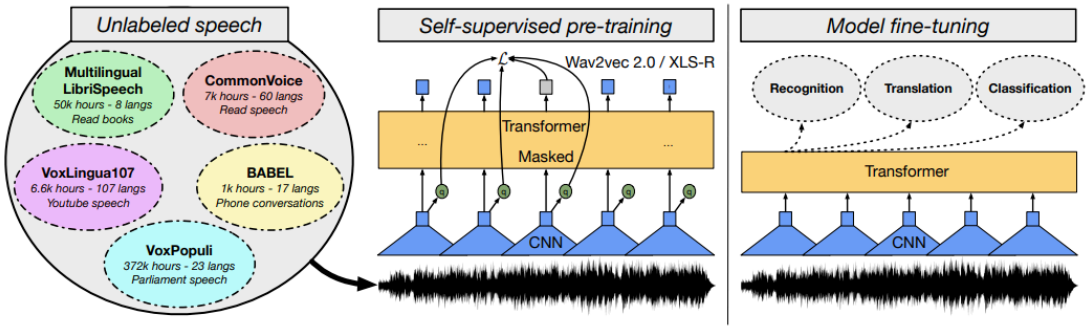
1. **Classificação de Gêneros Musicais:**
    - Essa tarefa consiste em categorizar uma faixa musical em um ou mais gêneros com base em suas características sonoras.
2. **Reconhecimento de Eventos Sonoros:**

- Envolve a identificação de eventos específicos em um ambiente, como o som de um carro passando, o latido de um cachorro ou um alarme tocando.

3. **Classificação de Voz:**
   - Refere-se à categorização de gravações de voz em diferentes classes, que podem incluir emoções ou contextos.
4. **Classificação de Sons Ambientais:**
   - Identifica e classifica sons de ambientes específicos, como sons de natureza, ruídos urbanos ou sons de máquinas.

1. Wav2Vec2-xlsr-2B (2020)

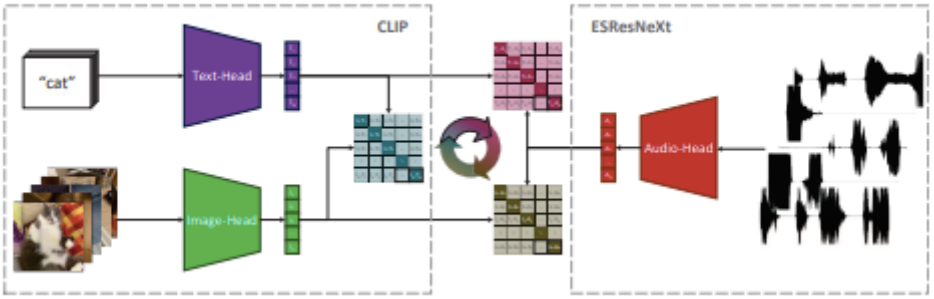
- É uma evolução do modelo Wav2Vec2.0, que utiliza aprendizado auto-supervisionado para representar dados de áudio, principalmente voz. O modelo xlsr (cross-lingual speech recognition) é otimizado para reconhecimento de fala em múltiplas línguas, e a versão 2B se refere ao número de parâmetros do modelo, indicando que é uma versão mais ampla e precisa.
- **Aplicações:** Reconhecimento automático de fala (ASR) multilinguístico, Detecção de emoções no áudio, e Transcrição de diálogos em vários idiomas.
- **Referência:** XLS-R: SELF-SUPERVISED CROSS-LINGUAL SPEECH REPRESENTATION LEARNING AT SCALE



2. AudioCLIP (2021)

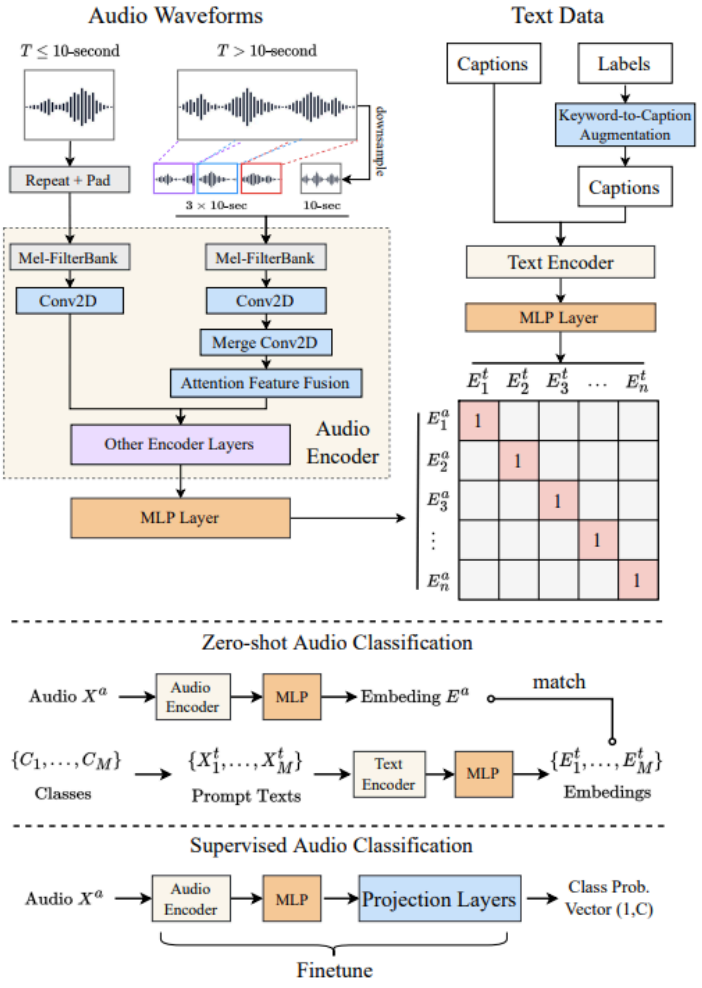
- É uma extensão do modelo CLIP, integrando dados de áudio à estrutura de aprendizado contrastivo. Ele combina informações de áudio, texto e imagem, tornando-o capaz de gerar representações multimodais que capturam a semântica das três modalidades.

- **Aplicações:** Classificação de sons, detecção de eventos multimodais e pesquisa cruzada entre áudio, imagem e texto.
- **Referência:** AudioCLIP: Extending CLIP to Image, Text and Audio



3. LaionCLAP(2022)

- É um modelo baseado em aprendizado contrastivo semelhante ao CLIP, mas focado especificamente em áudio e texto. Ele é treinado em uma grande quantidade de dados auditivos com descrições textuais, o que lhe permite capturar representações profundas tanto do áudio quanto do texto, possibilitando o alinhamento semântico entre esses dois domínios.
- **Aplicações:** Classificação de sons, identificação de música, detecção de emoções e tradução automática de descrições sonoras.
- **Referência:** Large-scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation



4. BEATs (2022)

- Um modelo de áudio que utiliza transformadores bidirecionais para capturar dependências temporais e contextuais de dados de áudio.
- **Aplicações:** Classificação de sons, reconhecimento de fala e tarefas relacionadas a eventos acústicos.
- **Referência:** BEATS : Audio Pre-Training with Acoustic Tokenizers

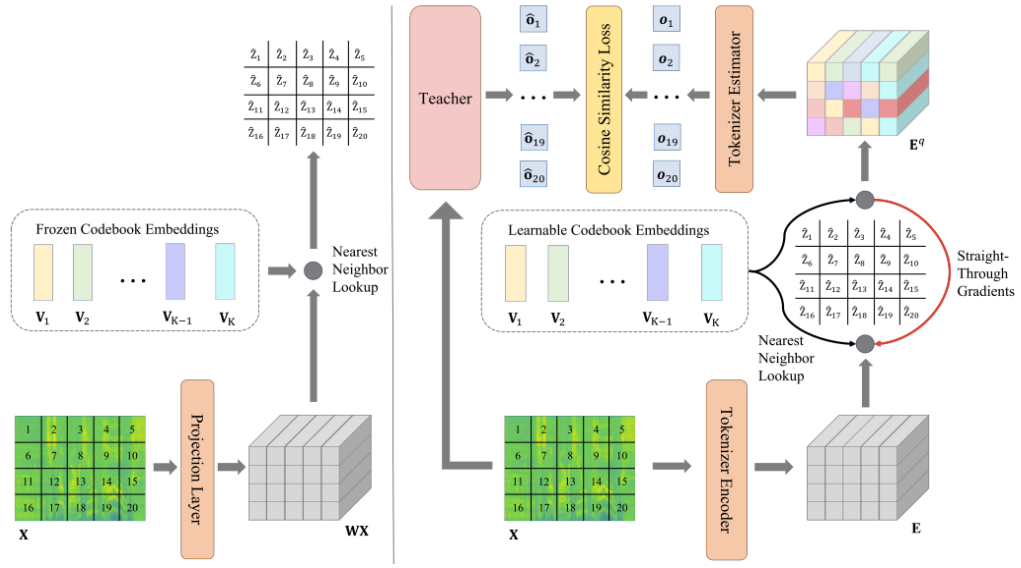


Figure 2: Acoustic tokenizers for discrete label generation.

Detecção de Atividade de Voz (VAD):

VAD é uma tarefa na análise de sinais de áudio, especialmente em aplicações que envolvem processamento de fala. O objetivo do VAD é identificar segmentos de áudio que contêm atividade vocal, distinguindo-os de períodos de silêncio ou ruído de fundo. Essa tarefa é crucial para otimizar o processamento de fala e melhorar a eficiência em várias aplicações, como:

1. **Reconhecimento de Fala:**
   - Em sistemas de reconhecimento de fala, o VAD é utilizado para concentrar a análise em segmentos relevantes de áudio, evitando o processamento de silêncios ou ruídos, o que pode aumentar a precisão do reconhecimento.
2. **Transcrição Automática:**
   - Para transcrições automáticas, o VAD permite identificar quais partes do áudio devem ser convertidas em texto, facilitando a geração de transcrições mais precisas e relevantes.

1. MarbleNet (2021)

- Rede neural end-to-end projetada para **detecção de atividade de voz,** baseada em blocos de convolução separável 1D de tempo-canal, com camadas de batch normalization, ReLU e dropout.
- **Aplicações:** Detecção de atividade de voz em dispositivos móveis e vestíveis.
- **Referência:** MARBLENET: DEEP 1D TIME-CHANNEL SEPARABLE CONVOLUTIONAL NEURAL NETWORK FOR VOICE ACTIVITY DETECTION

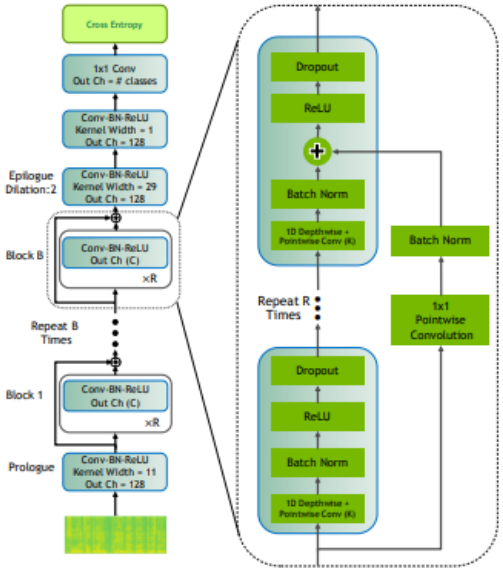


Fig. 1. MarbleNet $BxRxC$ model: B - number of blocks, R - number of sub-blocks, C - the number of channels.

2. Nas-vad (2022)

- Propõem uma macroestrutura modificada e um espaço de busca expandido, incluindo mecanismos de atenção. Os resultados experimentais mostram que os modelos gerados pelo NAS superam as arquiteturas manuais em conjuntos de dados com ruído e em cenários reais, além de generalizarem melhor para novos dados, com menos parâmetros.
- **Aplicações:** Detecção de atividade de voz em ambientes ruidosos.
- **Referência:** Nas-vad: Neural architecture search for voice activity detection.

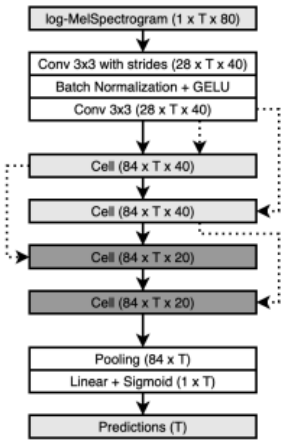


Figure 1: *The proposed macro structure. Each block denotes an operation or a module, and the parenthesis beside it shows the output shape of it. The first dimension is the channel dimension, and the second dimension, denoted as T, represents the time dimension, and the third number indicates the feature dimension. Dashed lines denote the second connection for each cell.*

Interpretações / Resumo artigo VALL-E

Artigo: Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers.

Resumo: Um modelo de fala Zero Shot TTS que utiliza uma abordagem com LMs para generalização do contexto, fato que só é possível devido aos Codecs Neurais de áudio, que geram representações discretas interpretáveis por esses modelos, saindo de abordagens tradicionais como mel-espectrogramas.

Base de treinamento: O modelo foi treinado com 60 mil horas de fala em inglês e mais de 7.000 locutores, permitindo uma generalização para falantes não vistos.

Inovação do VALL-E

Ao contrário dos modelos de TTS tradicionais, o VALL-E trata a síntese de fala como uma tarefa de modelagem de linguagem, usando tokens de áudio gerados por um modelo de codec neural. Ele consegue sintetizar fala personalizada com apenas 3 segundos de gravação de um locutor que nunca foi visto pelo modelo. Superando sistemas TTS anteriores em termos de naturalidade da fala e similaridade com o locutor.

- **Escalabilidade**: Diferente dos modelos tradicionais, que usam dezenas ou centenas de horas de dados de um único locutor ou múltiplos locutores.

- **Personalização/Capacidade in-contexto:** O VALL-E pode usar gravações curtas de áudio como orientação para sintetizar fala de alta qualidade que retém as características emocionais e o ambiente acústico do locutor.

Abordagem Principal

- **Uso de códigos de áudio discretos**: Em vez de trabalhar com mel-espectrogramas, o VALL-E usa um **codec de áudio neural** para codificar a fala em **tokens discretos**. Isso permite tratar a síntese de fala como uma tarefa de **modelagem de linguagem** - LLM. Esses tokens discretos servem como uma representação compacta do áudio, capturando informações ricas sobre o conteúdo e a voz.

- **Modelagem condicional**: O VALL-E trata o TTS como uma tarefa de **modelagem de linguagem condicional**, onde o modelo gera tokens de áudio com base em uma sequência de fonemas e um pequeno trecho de áudio do locutor. Permitindo que o modelo gere fala que mantém a identidade do locutor a partir de **apenas 3 segundos de gravação**.

- **Avaliações**: O VALL-E foi avaliado usando conjuntos de dados como o **LibriSpeech** e o **VCTK**, superando os modelos com +0.12 CMOS - comparação da naturalidade da fala - e +0.93 SMOS - comparação de similaridade do locutor.

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]:

Alexandre Costa Ferro Filho

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

As atividades realizadas nesta semana abrangeram a realização de novos estudos e análises, com foco na área de **Processamento de Áudio e Voz**:

- **Estudo Focado:** Pesquisa e estudo da área de maneira mais direcionada e aprofundada - SpeechLMs e Speech Tokenizers.
    - Leitura do paper recente: Recent Advances in Speech Language Models: A Survey
    - Leitura de papers relacionados: Low Frame-rate Speech Codec: a Codec Designed for Fast High-quality Speech LLM Training and Inference
        - Marcações: Low_Frame-rate_Speech_Codec_marcado_certo.pdf
        - Resumo: LFSC
- **Estudo de frameworks relacionados com Deep Learning e áudio:**
    - **Librosa:**
        - Estudo documentação: Librosa Docs
        - Elaboração material e funções: Librosa
        - Material prático: Librosa.ipynb
    - **Torchaudio:**
        - Estudo documentação: Torchaudio Docs
        - Elaboração de material e funções: Torchaudio

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

Para próxima Entrega planejo:

- Continuar estudo mais aprofundado focados em Áudio Neural CODECs e buscar implementações e testes nos repositórios disponíveis.
- Continuar estudo de frameworks relacionados com Deep Learning e áudio:

- Torchaudio - **Parte prática**
- NeMo
- Espnet

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Low Frame-rate Speech Codec: a Codec Designed for Fast High-quality Speech LLM Training and Inference

Resumo: apresenta um codec neural de áudio, o **Low Frame-rate Speech Codec (LFSC)**, desenvolvido para acelerar o treinamento e a inferência de modelos de linguagem grandes (LLMs) focados em fala, mantendo alta qualidade de áudio. Ele se destaca ao operar com **21,5 quadros por segundo (FPS)** e uma **taxa de bits de 1,89 kbps**, o que melhora a eficiência sem comprometer a qualidade.

Contribuições principais:

1. **Codec de baixa taxa de quadros**: LFSC reduz significativamente o número de quadros por segundo, quatro vezes menor do que codecs anteriores, como Spectral Codec e DAC, resultando em uma inferência mais rápida.
2. **Melhora no treinamento de modelos de fala baseados em LLM**: Usando LLMs de fala como discriminadores no treinamento de NACs, eles alcançam qualidade superior em cenários de baixa taxa de quadros.
3. **Avaliação empírica**: Comparado com outros codecs, o LFSC oferece uma melhora substancial na inteligibilidade e rapidez da inferência de modelos TTS, com um ganho de até três vezes na velocidade da inferência.

Modelo e Arquitetura

O LFSC usa uma arquitetura composta por uma rede neural convolucional totalmente conectada, com codificação vetorial quantizada e decodificadores baseados no HiFi-GAN. Os discriminadores incluem redes neurais adversariais e, inovadoramente, **modelos de linguagem de fala (SLMs)** como discriminadores, especificamente o WavLM pré-treinado. Isso auxilia o modelo a lidar com a complexidade das compressões de alta taxa.

Resultados e Discussão

O LFSC é competitivo com outros codecs SOTA, com uma melhoria significativa na qualidade percebida e na inteligibilidade da fala, medidos por MOS e CER. Além disso, oferece desempenho três vezes mais rápido do que os codecs anteriores em termos de

inferência. Também foi analisada outras métricas objetivas da fala com a codificação porém nada surpreendente, só mostra que não fica tanto prejudicado mesmo com alta performance.

Estudo de ZS-TTS

Na tarefa de zero-shot TTS (ZS-TTS), o codec alcançou taxas de erro de caracteres (CER) mais baixas e uma maior similaridade entre falantes, mantendo alta qualidade de fala.

Conclusão

O LFSC traz melhorias significativas para o treinamento de modelos de fala baseados em LLMs, tornando-se uma solução promissora para otimizar a eficiência sem comprometer a qualidade.

Librosa: Processamento de Áudio em Python

Definição/função: **Librosa** é uma biblioteca para análise e processamento de áudio, amplamente utilizada em tarefas de machine learning e inteligência artificial aplicadas ao som. Ela oferece uma variedade de ferramentas para **manipulação de sinais, extração de características e visualização de dados de áudio.**

Estrutura do Pacote Librosa

O **librosa** é organizado em uma coleção de **submódulos**, cada um oferecendo funcionalidades específicas para o processamento de áudio e análise musical. Cada submódulo e suas aplicações será discutido a seguir:

1. `librosa.beat`

- **Funções:** Estimação de andamento e detecção de eventos de batida.
- **Utilização:** Utilizado principalmente para detectar padrões rítmicos em músicas ou áudios. É amplamente aplicado em análise musical, como para identificar o andamento (BPM - batidas por minuto) ou localizar batidas ao longo de uma faixa.

2. `librosa.core`

- **Funções:** Contém as principais funcionalidades, como **carregamento de áudio, geração de espectrogramas e transformações fundamentais no áudio.**
- **Utilização:** O submódulo central do **librosa**, ele oferece operações essenciais para processamento de áudio, como carregar arquivos de áudio, executar transformadas de Fourier, e converter sinais em representações temporais e espectrais.

3. librosa.decompose

- **Funções:** Separação de fontes harmônicas-percussivas (HPSS) e decomposição de espectrogramas usando métodos de decomposição de matriz (`scikit-learn`).
- **Utilização:** Ideal para separar componentes harmônicos (melodias) e percussivos (batidas) em uma faixa de áudio. Isso é útil em várias aplicações, como remoção de ruído ou separação de instrumentos.

4. librosa.display

- **Funções:** Ferramentas para visualização de dados de áudio, integrando-se com `matplotlib`.
- **Utilização:** Usado para exibir formas de onda, espectrogramas, cromagramas e outras representações visuais de áudio. Facilita a análise visual de características do som.

5. librosa.effects

- **Funções:** Processamento de áudio no domínio do tempo, incluindo alteração de tonalidade (pitch shifting) e tempo (time stretching). Também fornece wrappers para funções do submódulo `decompose`.
- **Utilização:** Comumente usado para modificar o áudio sem perder qualidade, como quando se deseja mudar a velocidade de reprodução ou alterar o tom sem distorção.

6. librosa.feature

- **Funções:** Extração e manipulação de características de áudio, como MFCCs, espectrograma Mel, cromagramas, e outras características espectrais e rítmicas.

- **Utilização:** Essencial para tarefas de machine learning em áudio, como reconhecimento de fala, classificação de gêneros e análise de emoções.

7. librosa.filters

- **Funções:** Geração de bancos de filtros (chroma, CQT, pseudo-CQT, etc.). Esse submódulo é geralmente utilizado de forma interna por outras partes do **librosa**.
- **Utilização:** Útil para gerar filtros que ajudam na conversão de sinais de áudio em diferentes representações espectrais, como os filtros Mel ou CQT (Constant-Q Transform).

8. librosa.onset

- **Funções:** Detecção de início de eventos sonoros (onset) e cálculo de força desses eventos.
- **Utilização:** Utilizado em aplicações como a sincronização de batidas, segmentação de áudio e análise de transientes, para detectar o momento exato em que um som começa.

9. librosa.segment

- **Funções:** Ferramentas para segmentação estrutural, como construção de matrizes de recorrência, representação de defasagem de tempo e agrupamento.
- **Utilização:** Pode ser usado para segmentar diferentes partes de uma música ou gravação de áudio, como estrofes, refrões ou seções instrumentais.

10. librosa.sequence

- **Funções:** Ferramentas para modelagem sequencial, como decodificação de Viterbi e funções auxiliares para construção de matrizes de transição.
- **Utilização:** Essencial em tarefas de modelagem de sequências, como em algoritmos de detecção de fala contínua ou análise de padrão musical.

11. librosa.util

- **Funções:** Ferramentas auxiliares, como normalização, preenchimento de zeros e centralização de dados.
- **Utilização:** Esses utilitários são amplamente usados para manipulação de dados antes de serem processados por outras funções da biblioteca, ajudando a preparar o áudio de forma correta.

Torchaudio

Definção: é um pacote do PyTorch voltado para o processamento de áudio e voz, oferecendo uma variedade de funcionalidades para manipulação, transformação e análise de dados de áudio. Ele é amplamente utilizado em projetos de Machine Learning, como reconhecimento de fala, classificação de eventos sonoros e geração de áudio. Segue, um breve explicação e aplicação sobre cada submódulo existente na biblioteca.

1. `torchaudio`

- Este é o módulo oferecendo a interface de alto nível para **carregar, salvar e extrair informações de arquivos de áudio**. A principal função aqui é o carregamento de áudio em tensores do PyTorch, facilitando a integração direta com redes neurais.
- **Aplicações**: Usado principalmente para manipulação básica de arquivos de áudio, como carregamento e salvamento em diferentes formatos, e para acessar outras sub-bibliotecas do pacote.

2. `torchaudio.io`

- Este submódulo contém funções de baixo nível para leitura e escrita de áudio de forma eficiente. Facilita a manipulação de áudio em projetos de PyTorch, oferecendo uma interface intuitiva para interagir com arquivos de áudio e aplicar diversas transformações.
- **Aplicações**: Ideal para cenários que exigem controle sobre as operações de I/O de áudio, como em pipelines que lidam com grandes quantidades de dados ou formatos não convencionais. É uma ferramenta versátil que pode ser utilizada em diversas etapas de um projeto de processamento de áudio com PyTorch.

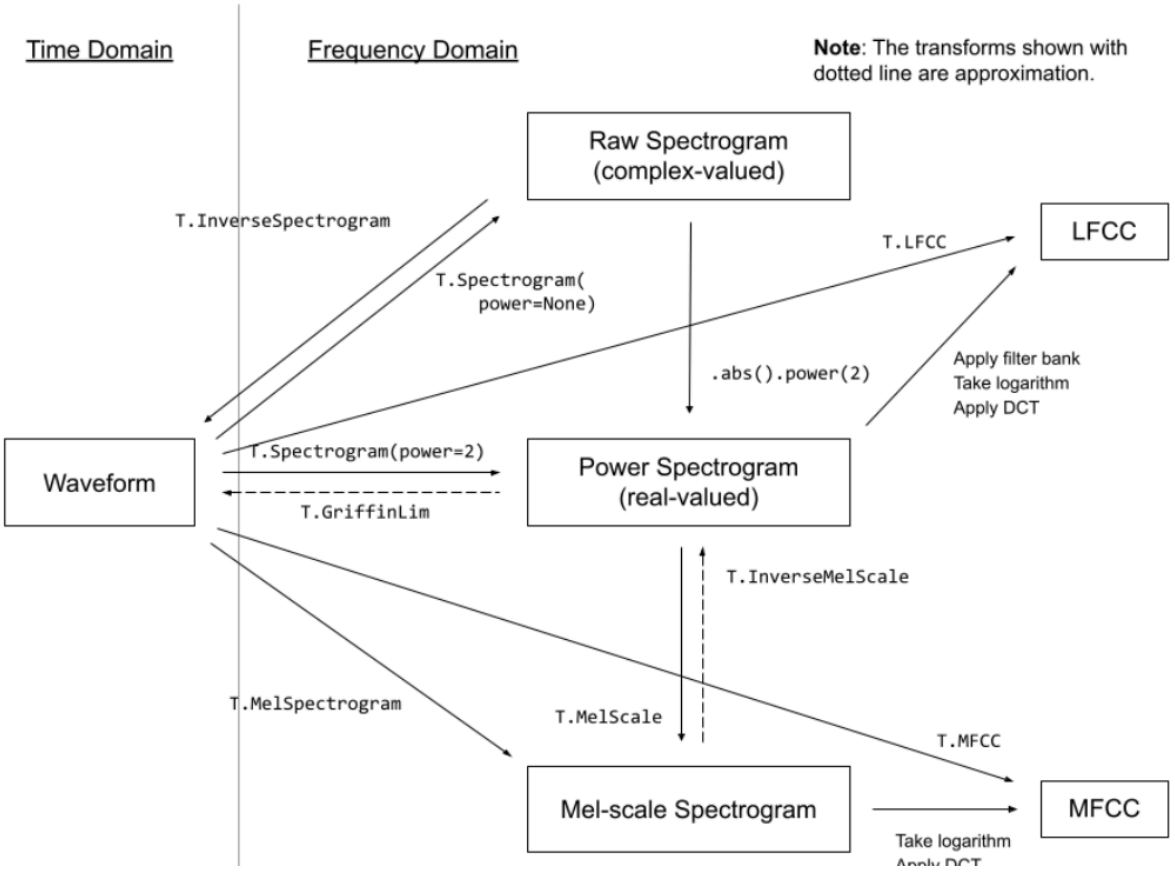
3. `torchaudio.functional`

- Este submódulo fornece funções de processamento de áudio em nível funcional, como equalização, resampling, modificação de volume, e adição de ruído. As funções são de baixo nível e permitem maior controle sobre as transformações e filtros.

- **Aplicações**: Usado para realizar transformações personalizadas em sinais de áudio e preparar dados para modelos de deep learning, como augmentação de dados de áudio ou pré-processamento de sinais.

4. torchaudio.transforms

- Oferece uma coleção de transformações de áudio de alto nível, que podem ser aplicadas diretamente a tensores de áudio. Exemplos incluem transformações como espectrogramas, Mel-frequency cepstral coefficients (MFCC), e resampling.
- **Aplicações**: Muito utilizado no pré-processamento de áudio para machine learning. As transformações são fundamentais para extrair características relevantes de sinais de áudio, como espectrogramas e MFCCs, que são comumente usadas em tarefas de reconhecimento de fala e classificação de som.
-



5. torchaudio.datasets

- Este módulo fornece interfaces para vários datasets populares de áudio, como o **LibriSpeech**, **VCTK**, e **CommonVoice**. Ele facilita o carregamento desses datasets de forma simplificada para uso em projetos de deep learning.

- **Aplicações**: Extremamente útil acessar rapidamente datasets de áudio padrão para treinamento de modelos relacionados com tasks de áudio e voz.

6. torchaudio.models

- Este submódulo contém modelos pré-treinados de redes neurais voltados para tarefas de processamento de áudio. Ele inclui arquiteturas de deep learning como Wav2Vec2.0, DeepSpeech, e outras redes que podem ser usadas para tarefas de áudio.
- **Aplicações**: Facilita a aplicação de modelos de última geração em tarefas de áudio sem a necessidade de treinar um modelo do zero. Útil para tarefas como transcrição automática de fala (ASR) ou classificação de fala.

7. torchaudio.models.decoder

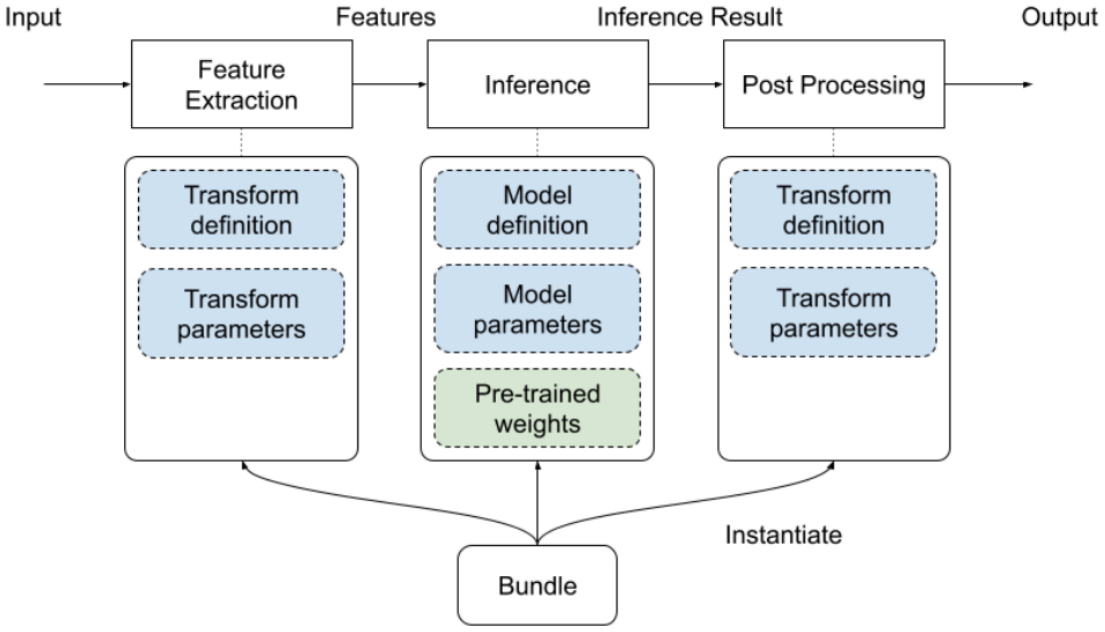
- Este submódulo é focado na decodificação de saída de modelos de reconhecimento de fala. Oferece funções para converter as predições dos modelos, como Wav2Vec, em texto utilizando algoritmos como beam search e decodificadores como CTC.

- **Aplicações**: Usado em sistemas de reconhecimento automático de fala (ASR) para melhorar a precisão da transcrição, fornecendo decodificação mais eficiente e precisa dos resultados.

8. torchaudio.pipelines

- Este módulo oferece pipelines pré-configuradas para tarefas de processamento de áudio. Ele combina modelos pré-treinados e transformações para criar um fluxo de trabalho que pode ser utilizado diretamente.

- **Aplicações**: Muito útil para quem deseja experimentar rapidamente uma solução de processamento de áudio, como reconhecimento de fala, utilizando pipelines otimizados e fáceis de usar.



9. `torchaudio.sox_effects`

- Esta parte da biblioteca permite aplicar efeitos de áudio do Sox diretamente aos tensores de áudio. O Sox (Sound eXchange) é uma ferramenta poderosa de manipulação de áudio, e este submódulo integra seus efeitos diretamente no PyTorch.
- **Aplicações**: Usado para manipulação avançada de áudio, como equalização, normalização, ajuste de pitch, entre outros efeitos sonoros, permitindo a personalização de dados de áudio para diversas aplicações.

10. `torchaudio.compliance.kaldi`

- Este submódulo oferece funções compatíveis com a biblioteca Kaldi, que é amplamente utilizada em tarefas de reconhecimento de fala. Ele permite que usuários usem funções de Kaldi diretamente no PyTorch, como a extração de features acústicas.
- **Aplicações**: Ideal para quem está migrando de Kaldi para PyTorch ou deseja utilizar algoritmos de Kaldi em novos projetos de reconhecimento de fala.

11. `torchaudio.kaldi_io`

- Este submódulo é uma extensão para permitir a leitura e escrita de dados no formato Kaldi. O Kaldi usa formatos de arquivo específicos, e este módulo permite que esses dados sejam facilmente manipulados no ambiente do PyTorch.
- **Aplicações**: Facilita a integração de pipelines Kaldi com PyTorch, especialmente para quem já possui um conjunto de dados ou pipelines configurados no Kaldi.

12. `torchaudio.utils`

- Contém várias funções utilitárias que suportam as operações dentro do Torchaudio. Isso inclui funções para manipulação de tensores de áudio, conversão de formatos de dados, entre outras tarefas auxiliares.
- **Aplicações**: Usado para suporte geral em manipulações de dados de áudio, como conversão de formato ou manipulação de tensores de áudio, facilitando a integração e o fluxo de trabalho entre diferentes componentes do **Torchaudio**.

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]:

Alexandre Costa Ferro Filho

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

As atividades realizadas nesta semana abrangeram a realização de novos estudos e análises, com foco na área de **Processamento de Áudio e Voz**:

- **Estudo aprofundado:** Realização de um estudo aprofundado sobre algoritmos de áudio neural codecs, incluindo a avaliação de diferentes abordagens e dos principais modelos propostos.
    - Leitura detalhada dos artigos sobre **SoundStream** e **EnCodec**, com foco nas metodologias, resultados e impactos dos codecs em aplicações de áudio.
    - **Apresentação sobre Speech Tokenizers**: Análise e explicação dos tipos de tokenizadores de fala, com ênfase no padrão das arquiteturas empregadas em áudio neural codecs: Speech Tokenizers.
    - **Testes em repositórios do Framework NeMo**: Avaliação e experimentação com implementações de modelos estudados, utilizando o framework NeMo.
      Links github importantes: Encodec Modules, Audio Codec Model, Audio Codec Inference.
        - **Exploração das etapas dos modelos**: Análise detalhada das etapas de processamento, incluindo entrada (input shapes) e arquitetura dos modelos.
          codec_modules.ipynb
        - **Testes e modificações**: Realização de testes modificando parâmetros para entender o impacto no desempenho e na qualidade dos codecs de áudio.
          Audio_Codec_Training.ipynb

- **Estudo frameworks relacionados com Deep Learning e áudio**
    - **NeMo**
        - Estudo documentação: NeMo Docs
            - TTS.

- Elaboração material e funções: NeMo
- **Espnet**
  - Estudo documentação: Espnet2 Docs
  - Elaboração material e funções: Espnet

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

Para próxima Entrega planejo:

- **Levantamento de Datasets de Áudio**: Identificar e compilar datasets relevantes para a área de síntese de fala, documentando características como taxa de amostragem, qualidade do áudio, domínio específico (ex: conversação, música, ambientes ruidosos).
- **Exploração de Novas Aplicações em Português**: Estudar aplicações ainda pouco exploradas para codecs de áudio, com foco em domínios específicos da língua portuguesa.
- **Terminar a exploração nos frameworks (NeMo e EspNet) e aderir um para seguir.**

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

NeMo: Framework para Modelos de IA em Áudio e Linguagem Natural

Definição/função: **NeMo** (NVIDIA NeMo) é um framework desenvolvido pela NVIDIA, projetado para a **construção e treinamento de modelos de inteligência artificial** focados em processamento de linguagem natural (**NLP**), reconhecimento de fala (**ASR**) e síntese de voz (**TTS**). Ele facilita a implementação de modelos complexos, integrando ferramentas para pré-processamento de dados, treinamento e ajuste fino, além de permitir a exportação dos modelos para implantação em dispositivos variados.

Estrutura do Pacote NeMo

O **NeMo** é estruturado em diversos módulos **especializados**, cada um oferecendo funcionalidades específicas para diferentes áreas de IA em áudio e NLP. Abaixo, apresento os principais funcionalidades e suas aplicações na área de speech:

- **Treinamento e Personalização**: Ferramentas para criar modelos de ASR, classificação de fala, reconhecimento e diarização de falantes e TTS, com uma abordagem reprodutível.
- **Modelos Pré-treinados**: Oferece checkpoints de última geração (SOTA) para ASR e TTS, incluindo orientações para uso.
- **Ferramentas para Dados de Fala**:
  - **NeMo Forced Aligner (NFA)**: Gera timestamps detalhados para áudio com modelos baseados em CTC.
  - **Speech Data Processor (SDP)**: Simplifica o processamento de dados com arquivos de configuração reutilizáveis.
  - **Speech Data Explorer (SDE)**: Aplicação interativa para explorar e analisar datasets.
  - **Ferramentas de Criação e Avaliação de Dataset**: Alinha áudios longos com transcrições e permite avaliação de modelos ASR com precisão de palavras e detecção de atividade vocal.

    - **Ferramenta de Normalização de Texto**: Converte entre formas escrita e falada.

- **Inferência**: Modelos treinados no NeMo podem ser otimizados para produção com o NVIDIA Riva, que oferece ferramentas de implantação automatizada em ambientes empresariais.

- **Suporte a Modelos**: O NeMo suporta fluxos completos de desenvolvimento para modelos de linguagem e multimodais, incluindo Gemma, Llama, Baichuan, Falcon, e modelos NeMo próprios.

NeMo Collections

- **ASR** - collection of modules and models for building speech recognition networks
- **TTS** - collection of modules and models for building speech synthesis networks
- **NLP** - collection of modules and models for building NLP networks
- **Vision** - collection of modules and models for building computer vision networks
- **Multimodal** - collection of modules and models for building multimodal networks
- **Audio** - collection of modules and models for building audio processing networks

EspNet: End-to-end Speech Processing toolkit

Definição/função: O **ESPnet** é um kit de ferramentas de processamento de fala de ponta a ponta que abrange **reconhecimento de fala de ponta a ponta, conversão de texto em fala, tradução de fala, aprimoramento de fala, diarização de locutor, compreensão da linguagem falada e assim por diante**. O ESPnet usa o pytorch como um mecanismo de aprendizado profundo e também segue o processamento de dados no estilo Kaldi, extração/formato de recursos e receitas para fornecer uma configuração completa para vários experimentos de processamento de fala.

Estrutura do Pacote ESPnet:

O **ESPnet** é organizado em uma coleção de módulos, cada um focado em funcionalidades específicas para o reconhecimento de fala e outras tarefas relacionadas a processamento de áudio e fala. Abaixo estão os principais módulos e suas aplicações:

1. `ESPnet.asr`

Funções: Inclui as principais ferramentas para Reconhecimento Automático de Fala (ASR), suportando desde o pré-processamento dos dados até o treinamento e a inferência dos modelos.
Utilização: Utilizado para converter fala em texto com precisão, o módulo ASR permite treinar modelos customizados ou utilizar modelos pré-treinados para diversas aplicações, como sistemas de transcrição automática e assistentes de voz.

2. `ESPnet.tts`

Funções: Contém funcionalidades para Síntese de Texto para Fala (TTS), permitindo transformar texto em fala natural usando redes neurais avançadas, como Tacotron e

Transformer-TTS.
Utilização: Esse módulo é usado para gerar áudio a partir de texto, sendo aplicado em leitores automáticos e outras aplicações que necessitam de respostas faladas naturais.

3. `ESPnet.st`

Funções: Responsável pela Tradução de Fala (Speech Translation), que traduz diretamente de um idioma falado para outro.
Utilização: Útil para tradução em tempo real, esse módulo facilita a criação de sistemas de tradução automática de voz, possibilitando a comunicação entre diferentes idiomas sem a necessidade de um passo intermediário de conversão de fala para texto.

4. `ESPnet.enh`

Funções: Módulo de Melhoria de Áudio, dedicado a reduzir o ruído e melhorar a qualidade de gravações de áudio.
Utilização: Aplicado em contextos onde a qualidade do áudio é essencial, como em chamadas de conferência e gravações de baixa qualidade, esse módulo ajuda a melhorar a clareza e inteligibilidade da fala.

5. `ESPnet.bin`

Funções: Inclui scripts e ferramentas para executar tarefas e configurar experimentos de treinamento e inferência, facilitando o uso de modelos do ESPnet.
Utilização: Serve como a interface de comando do ESPnet, permitindo o gerenciamento de experimentos e ajustando modelos para tarefas específicas em uma linha de comando.

O ESPnet oferece uma plataforma robusta e modular para desenvolvimento e experimentação em ASR, TTS e outras aplicações de fala, o que o torna uma opção poderosa para pesquisas avançadas e implementações práticas em sistemas de áudio e fala.

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: 6 de nov. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Alexandre Costa Ferro Filho

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

As atividades realizadas nesta semana abrangeram a realização de novos estudos e análises, com foco na área de **Processamento de Áudio e Voz**:

- **Levantamento de Datasets de Áudio**: Identificar e compilar Datasets relevantes para a área de síntese de fala, com uma pequena documentação acerca deles, como taxa de amostragem, formato e peculiaridades.
  - Levantamento Datasets
- **Exploração de Novas Aplicações em Português**:
  - Problemas de síntese de fala demandam alto custo computacional.
    - Improving Robustness of LLM-based Speech Synthesis by Learning Monotonic Alignment
  - Realizar aplicação explorando a compreensão dos Áudio Neural CODECs.
    - Transporte grande quantidade de áudios via redes em sistemas embarcados.
  - Migrar aplicação de síntese de fala para tarefas mais simples:
    - Classificação de áudio - reconhecimento de emoções ou detecção de deep fakes.
- **Exploração nos frameworks (NeMo e EspNet) e aderir um para seguir.**
  - EspNet modelos suportados:
    - SoundStream
    - Encodec
    - DAC
    - FunCodec
    - HiFiCodec
  - NeMo modelos suportados
    - AudioCodec
    - Encoder

- MelCodec
- **Low Frame-rate Speech Codec**

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

Para próxima Entrega planejo:

- **Levantamento de recursos e viabilidade de treinar síntese de fala com CODECs**:
  - Delimitar arquitetura de modelo de linguagem para construção do speechLM.
- **Realizar treinos iniciais**
  - Adaptar código do NeMo para integrar LFSC no speechLM.
  - Testar integração simples com mlp.

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

Levantamento Datasets

- **CML-TTS**
  - Sample rate: **24kHz**
  - Formato: **.flac**
  - 70 horas de áudio em português
  - Cerca de 10gb de áudio.

- **TTS-Portuguese-Corpus**
  - 10,5 horas de áudio em português.
  - Apenas um locutor
  - Sample rate: **48kHz**
  - Formato: **.wav**

- **Fleurs:**
  - No Hugging Face, disponíveis os arquivos de transcrição em .tsv e pastas compactadas .tar.gz contendo os arquivos de áudio, divididos em “dev”, “test” e “train”. O download através da biblioteca “datasets”, em Python, é uma opção.
  - Sample rate: **16kHz**
  - Formato: **.wav**
  - 2.67GB de áudios em português

- **CETUC**
  - Formato: **.wav**
  - 145 horas de áudio
  - Apresenta os prompts de todos os áudios
  - Sample rate: **16kHz**

- **Constituição Federal**
  - Formato: **.wav**
  - 9 horas de áudio
  - 30 segundos de áudio em média
  - Sample rate: **16kHz**

- **Código de Defesa do Consumidor**
  - Sample rate: **16kHz**
  - Formato: **.wav**
  - Outras informações:
    - Voz masculina, alto SNR: áudio de boa qualidade com pouco ruído
    - 85min de áudio

- **Common Voice**
  - Formato: **.mp3**
  - Domínio Público
  - 176 horas validadas
  - Sample rate: **32kHz**
  - Baixo SNR - ambiente não controlado (ruído presente)

- **Librispeech**
  - Formato: **.flac (9GB) e .opus (2.5G)**
  - Pasta separada unicamente para teste
  - Sample rate: **16kHz**
  - Alto SNR - ambiente controlado

- **TEDX**
  - Sample rate: **44.1kHz** ou **48kHz**

- Formato: **.wav / .flac**
- Outras informações:
  - linguagem de origem, título, palestrante, duração do áudio, palavras-chave e uma descrição curta da palestra.
  - contém 164 horas de áudio e aproximadamente 93 mil enunciados em português
  - Tamanho total: 27.1 GB

- **Sidney**
  - Formato: **.wav**
  - Domínio Público
  - Apresenta os prompts de todos os áudios
  - Sample rate: **22.05kHz**
  - Alto SNR - ambiente controlado
  - Aproximadamente 7 horas

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: 14 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Alexandre Costa Ferro Filho

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

As atividades realizadas nesta semana abrangeram a realização de novos estudos e análises, com foco na área de **Processamento de Áudio e Voz**:

- **Levantamento de recursos e viabilidade de treinar síntese de fala**:
  - Não foi encontrada uma alternativa low resource para treinamento de síntese de fala, dentro das limitações de tempo da residência.
- **Realizar teste iniciais:**
  - Análise sobre compressão dos CODECs para transmissão de dados.
    Latência Codecs
- **Realizar treinos iniciais:**
  - Problemas na implementação do LFSC em speechLMs - Falam que tem checkpoint público porém não foi disponibilizado ainda.
  - Teste integrando com AudioCodec - arquitetura semelhante.
  - Treinamento de MLP para classificação de deep fake voice. teste_1_speechLM.ipynb

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

Para próxima Entrega planejo:

- **Continuação dos treinamentos para classificação:**
  - Utilização de diferentes modelos e técnicas.
- **Levantar/computar tempo de inferência entre diferentes Codecs.**

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

Estudo Latência Codecs para Transferência

Vantagens da utilização dos **CODECs** na transferência de dados

- Facilitando a transmissão de áudio em tempo real.
- Reduzindo significativamente a latência, especialmente em aplicações de baixa largura de banda e em sistemas que exigem respostas rápidas.

Minimizar a quantidade de dados necessária para transmitir áudio com qualidade aceitável, economizando largura de banda e permitindo uma comunicação mais fluida.

Impacto da utilização dos CODECs

Considere um áudio de 1 segundo de duração com qualidade de CD, que tem uma taxa de amostragem de 44.1 kHz e uma profundidade de 16 bits por amostra em estéreo. O tamanho bruto desse áudio pode ser calculado como:

44.100 amostras/s × 16 bits/amostra × 2 canais = 1.411.200 bits/s = **1.411kbps**

Para transmitir 1 segundo de áudio em qualidade CD sem compressão, seriam necessários 1.411 kbps. Para redes de baixa largura de banda, essa taxa é inviável. Codecs como Opus, utilizado em comunicação de baixa latência, reduzem esse valor drasticamente. Com a compressão do Opus, por exemplo, é possível reduzir essa taxa para cerca de 32 a 64 kbps sem perda perceptível de qualidade em voz. Isso representa uma redução de mais de 95% no volume de dados.

Melhorando ainda mais essa relação, utilizando Neural Audio Codecs esses valores chegam a ser menores ainda, como no caso do Audio Codec com 6.4 kbps.

Isso representa uma redução de cerca de 99,4% em relação ao áudio sem compressão e de 75% em relação ao codec tradicional Opus.

Latência e Eficiência

A redução no volume de dados se traduz em menor latência em redes de baixa capacidade. Em uma rede com velocidade de upload de 500 kbps, por exemplo, um áudio comprimido

em 32 kbps poderia ser transmitido com apenas uma fração da capacidade total, permitindo uma comunicação mais rápida e a alocação de recursos de rede para outras tarefas.

Em resumo, os codecs de áudio eficientes reduzem a latência ao reduzir o tamanho dos dados transmitidos. Essa compressão inteligente não só facilita a transmissão de áudio em redes limitadas, mas também melhora a experiência do usuário em tempo real, oferecendo uma comunicação mais fluida e responsiva. Porém deve ser analisado o custo que é levado ao realizar a codificação para o funcionamento real time.

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: 28 de out. de 2024

Participantes da Entrega [matriculados em Residência em IA]:

Alexandre Costa Ferro Filho

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

As atividades realizadas nesta semana abrangeram a realização de novos estudos e análises, com foco na área de **Processamento de Áudio e Voz**:

- **Teste de diferentes abordagens com treinamentos de classificadores**:
  - Dificuldade: modelos não estão aprendendo bem as representações de áudio.
- **Levantamento dos melhores CODECs para tarefa de transmissão de áudios em streaming para redes com baixa latência.**
  - Codecs Streaming
- **Construção do fluxo de transmissão de áudios contínuos - Retirar peso da rede.**
  - Áudio do microfone para um servidor através de websockets.
  - Teste do fluxo com e sem integração dos Neural Audio Codecs.
    Testes Latência e Controle de Tráfico
  - Utilização da ferramenta tc no linux para controle do tráfego da rede.
  - Github da aplicação construída até o momento:
    - Utilização de docker e documentação para replicar experimentos: https://github.com/alexandreacff/Audio_Codecs_Streaming.git

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

Para próxima Entrega planejo:

- Continuar em diferentes abordagens para treinamento de modelos com a representação em tokens acústicos com a esperança de obter resultados satisfatórios em um baixo custo de inferência.
- Realizar o estudo final sobre a degradação do áudio ao utilizar os CODECs na transmissão de

áudios.

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

ACEITE DA ENTREGA:

CEDRIC LUIZ DE CARVALHO: Go!

CODECs Interessantes para realização de streaming

- **WavTokenizer - 24khz**
  - Estado da arte atual.
  - 0.9kbs
  - A reconstrução dos áudio é boa sonoramente
  - Atrapalha menos downstream tasks.
- **APCodec - 48khz**
  - Qualidade tão boa quanto wavtokenizer
  - 6kbs
  - Descrito como um bom codec para realizar streaming.
- **AudioDec - 48khz**
  - Modelo da meta voltado justamente para streaming de audios
  - 12.8kbs
  - Não testado
- **AudioCodec - 16khz**
  - Tem uma taxa de amostragem menor
  - 6.4kbs
  - Integração fácil com NeMo

Testes de Latência e Utilização de Tokenizers em Fluxos de Áudio via WebSockets

O crescente uso de sistemas de comunicação em tempo real, especialmente em aplicações de **streaming de áudio**, destaca a importância de **pipelines eficientes e de baixa latência.** WebSockets, amplamente utilizados para comunicação bidirecional em tempo real, oferecem uma base robusta para o transporte de fluxos de áudio, mas sua performance pode ser significativamente influenciada por técnicas de compactação e codificação de dados.

Ambientes que a utilização dessas técnicas se torna indispensáveis:

- **Redes de Baixa Velocidade**: Em áreas com infraestrutura de rede limitada, codecs neurais podem reduzir significativamente o tamanho dos dados transmitidos sem perda perceptível de qualidade.
- **Aplicações em Tempo Real com Recursos Limitados**: Assistentes virtuais, streaming de áudio ou chamadas VoIP, onde latência e qualidade precisam ser otimizadas simultaneamente.
- **Transmissão Massiva e Escalável**: Plataformas que distribuem bilhões de transmissões simultâneas podem se beneficiar de codecs neurais para economizar largura de banda globalmente.
- **Ambientes Críticos**: Para telemedicina ou robótica, onde a perda de informações pode ser prejudicial, um codec eficiente pode ser imprescindível - **Foco da aplicação.**

Neste contexto, os tokenizers desempenham um papel essencial ao transformar fluxos de áudio em representações mais compactas e processáveis. O teste de diferentes implementações de tokenizers pode revelar suas implicações em termos de latência, qualidade do áudio e eficiência de transmissão. Além disso, o **controle do tráfego** durante esses testes é crucial para avaliar a capacidade do sistema em cenários variados de carga e banda disponível, simulando condições reais de rede.

Esta pesquisa se propõe a investigar como CODECs Neurais específicos afetam a latência em pipelines de comunicação de áudio via WebSockets, analisando métricas-chave como tempo de transmissão, processamento e reconstrução do áudio. A implementação do **controle de tráfego** permite avaliar o desempenho sob diferentes níveis de congestionamento, fornecendo insights valiosos para o desenvolvimento de soluções mais robustas e eficientes.

O controle de tráfego foi realizado utilizando o programa tc, é uma ferramenta do pacote amplamente utilizada em sistemas Linux para gerenciar e controlar o tráfego de rede. Ele permite **configurar, monitorar e manipular** diferentes aspectos do tráfego, como filas, prioridades, **limites de banda** e **introdução de latência artificial**. Isso o torna uma escolha essencial para **simular diferentes condições de rede**, o que é indispensável em testes de sistemas que dependem de comunicação em tempo real, como pipelines de áudio em WebSockets.

Testes de cpu - Média de 100 transmissões:

- Sem controle de tráfego:

Tipos de Fluxo	Latência - 100 ms	Latência - 1000 ms
Normal	11 ms	44,6 ms
WavTokenizer	60,15 ms	230,37 ms

- Teste reduzindo a taxa de banda larga para 3 Mbit:

Tipos de Fluxo	Latência - 100 ms	Latência - 1000 ms
Normal	33,55 ms	304,49 ms

WavTokenizer	76,42 ms	342,88 ms

- Teste reduzindo a taxa de banda larga para 1 Mbit:

Tipos de Fluxo	Latência - 100 ms	Latência - 1000 ms
Normal	80,8 ms	3266,80 ms
WavTokenizer	86,42 ms	356,63 ms

- Teste reduzindo a taxa de banda larga menos de 1 Mbit:

  O modelo de fluxo normal até para 100 ms aumenta a latência drasticamente, enquanto o Wavtokenizer mantém seu valor.

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]:

Alexandre Costa Ferro Filho

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

As atividades realizadas nesta semana abrangeram a realização de novos estudos e análises, com foco na área de **Codecs Neurais de Áudio**:

- **Testes e Análises com Treinamentos de Classificadores Neurais Simples**:
  - **Problema abordado:** Detecção de deepfake em áudio.
    - **Objetivo:** Identificar deepfakes em áudios utilizando diferentes arquiteturas de classificadores.
  - **Arquiteturas testadas:**
    - MLP (Perceptron Multicamadas)
    - CNN1d (Redes Neurais Convolucionais 1D)
    - CNN1d + LSTM (Rede Convolucional com Memória de Longo Prazo)
    - Modelos estatísticos
  - **Resultados:**
    - Acurácia dentro do domínio: 96% (MLP).
    - Acurácia fora do domínio: 90% (MLP).
  - **Documentação e logs:**
    - Logs treinamento

- **Treinamento do BERT com Representação Customizada:**
  - **Atividades realizadas:**
    - Modificação do tamanho do vocabulário (vocab size) e da dimensão dos embeddings (emb dim).
    - Treinamento do modelo BERT para adaptar à representação específica de tokens de áudio.
  - **Resultados obtidos:**
    - Modelo não conseguiu generalizar bem o conjunto do próprio dataset de treino.
    - Possivelmente devido ao pouco tempo de treinamento.

        - Métricas de loss tiveram um bom comportamento, o que indicava que o modelo estava aprendendo algo.
        - Logs: Testes Realizados com o BERT

- **Testes de Degradação de Áudio**
    - **Objetivo:** Avaliar a degradação do áudio para tarefas específicas após passar por codecs neurais.
    - **Metodologia:**
        - Realizar inferência em áudios reconstruídos por diferentes codecs e medir o impacto em tarefas como classificação e síntese.
    - **Codecs testados:**
        - LFRSC
        - WavTokenizer
    - **Tarefas Testadas** - Biometria de voz:
        - Teste degradação
    - **Resultados esperados:**
        - Identificar codecs que preservam melhor as características dos áudios para diferentes aplicações.
        - Comparar a eficiência dos codecs para cenários de uso.

- **Fine-tune do XTTs, arquitetura semelhante a CODECs.**
    - **Arquitetura:**
        - Zero Shot TTS
    - **Atividades:**
        - Especialização da síntese zero-shot para um falante específico.
        - Explorar a capacidade do XTTs em reproduzir a voz-alvo com alta fidelidade.
    - **Resultado:**
        - Aumento da qualidade de síntese personalizada.
        - Dialog-9-Turn-7-Speaker-Doctor-Voice-12287_12742_000003-0001.wav

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!

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

Registro de Logs e Resultados de Testes de Treinamento

Introdução

Este documento tem como objetivo centralizar as informações relacionadas aos testes realizados durante o treinamento de modelos para classificação de áudios deep fake. Ele detalha os logs, configurações, métricas, observações e resultados obtidos, auxiliando na análise e aprimoramento contínuo dos modelos.

Estrutura do Registro

1. Detalhes do Experimento

- **Métricas:** Acurácia e Equal Error Rate
- **Modelo Utilizado:** MLP, CNN1d, CNN1d + LSTM,
- **Objetivo do Experimento:** Alcançar um modelo satisfatório que
- **Datasets:** LA, MLAAD e In-The-Wild

---

2. Configurações do Treinamento

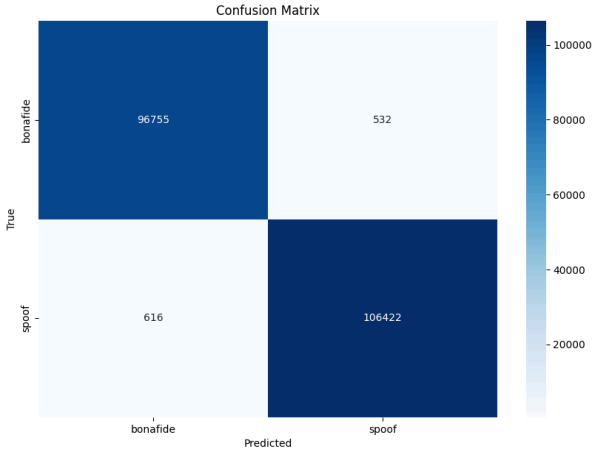
- **Framework/Plataforma:** PytorchLichning e Transformers
- **Hyperparâmetros:**
  - Taxa de Aprendizado: 1e-4
  - Tamanho do Batch: 32
  - Número de Épocas: ~15
  - Otimizador: ADAM
  - Outros (ex.: regularização, inicialização de pesos): Estratégias como batch normalization, dropout e diferentes funções de ativação foram testadas.

3. Logs do Treinamento

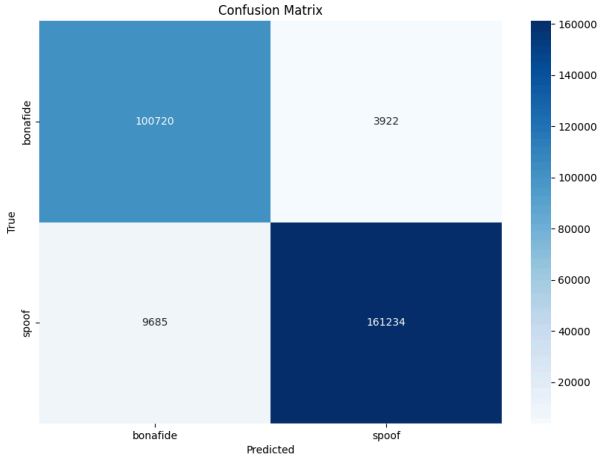
- **MLP**
  - `train_loss=0.120, val_loss=0.176, val_acc=0.93`
- **CNN1d**
  - `train_loss=0.267, val_loss=0.238, val_acc=0.905`
- **CNN1d + LSTM**
  - `train_loss=0.174, val_loss=0.215, val_acc=0.914`
- **Modelos estatísticos:** sem logs, porém os resultados não foram interessantes.

4. Resultados Finais Melhor Experimento:

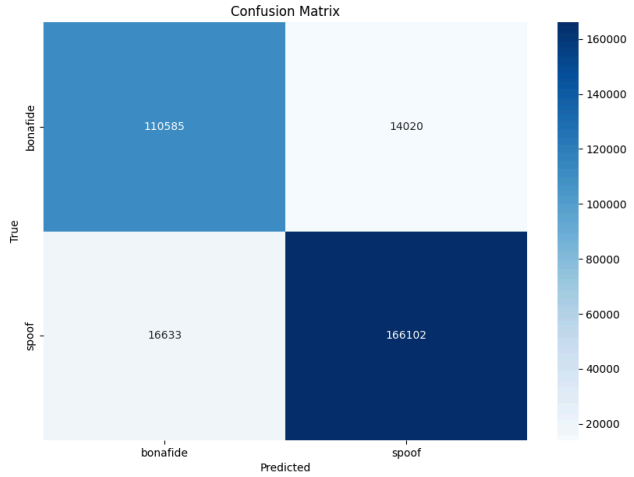
- **Modelo Finais no Teste com melhores resultados:**
  - MLP
- **Métricas Finais no Teste:**
  - Matriz de confusão:
    - MLAAD:



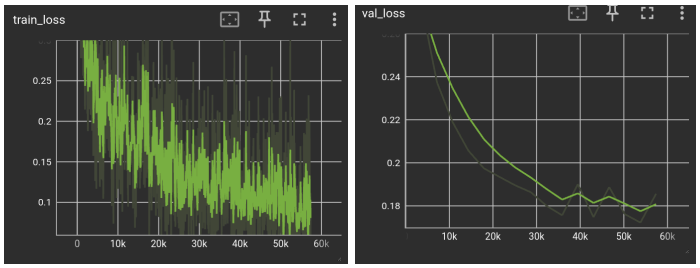
- LA:



- In-The-Wild:



- **Gráficos:**



---

5. Observações e Insights

- **Comportamento do Modelo:**
  - Para os dados dentro do mesmo domínio o modelo performou bem.
  - Quando testado em um dataset não visto no treinamento a performance teve uma piora.
- **Desafios Encontrados:**
  - Erros comuns ou problemas no treinamento.
  - Problemas para garantir um bom Dataset balanceado
- **Sugestões para Melhorias Futuras:**
  - Alterações no dataset ou no modelo.
  - Modificações nos hyperparâmetros.
  - Treinar modelos mais complexos com essas representações.

---

6. Conclusão

O MLP Classifier demonstrou ser uma abordagem eficaz para a tarefa de classificação proposta.

Futuras melhorias podem incluir a experimentação com diferentes arquiteturas de rede, ajustes de hiperparâmetros e técnicas de regularização para melhorar ainda mais a performance do modelo.

Testes Realizados com o BERT

Introdução

O experimento está focado em ajustar e treinar um modelo BERT para uma tarefa de classificação de sequência, utilizando tokens de áudio e uma configuração personalizada. Aqui estão os principais pontos que podem ser incluídos na conclusão:

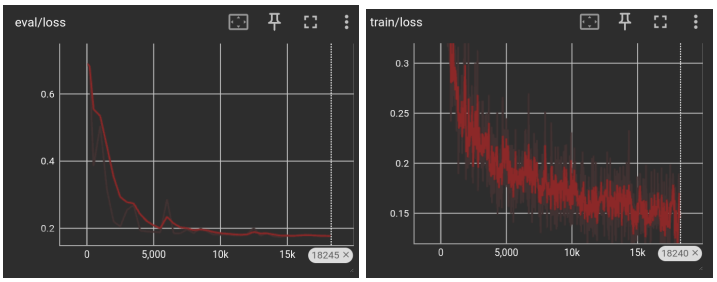
Configuração Personalizada do Modelo: O modelo BERT foi configurado com um vocabulário de 4096 tokens, 256 dimensões de embeddings, 6 camadas Transformer, 8 cabeças de atenção e um comprimento máximo de sequência de 512.

Preparação dos Dados: Os dados de treinamento e validação foram preparados a partir de um conjunto de tokens específico, indicando um pré-processamento cuidadoso dos dados de entrada.

Objetivo do Experimento: O objetivo principal parece ser avaliar a eficácia do modelo BERT configurado para uma tarefa de classificação binária, relacionada à detecção de spoofing em dados de fala.

Resultados Obtidos: Os resultados obtidos indicam que o modelo apresentou dificuldades em generalizar para o próprio conjunto de treino. Essa limitação pode ser atribuída, em grande parte, ao curto tempo de treinamento, que pode ter sido insuficiente para que o modelo consolidasse um aprendizado robusto.

Apesar dessa dificuldade, as métricas de loss demonstraram um comportamento satisfatório, sugerindo que o modelo estava, de fato, captando padrões nos dados. Isso indica potencial de melhoria com ajustes no tempo de treinamento ou outras configurações, como hiperparâmetros ou estratégias de regularização, para melhorar a capacidade de generalização.


eval/loss
0.6
0.4
0.2
0
5,000
10k
15k
18245
train/loss
0.3
0.25
0.2
0.15
0
5,000
10k
15k
18240

Testes Degradação Áudio Codecs

Introdução

Este documento centraliza as informações relacionadas aos testes realizados para avaliar a performance de modelos em tarefas simples de processamento de áudio. Os testes consistem em analisar a execução de tasks específicas em áudios originais e reconstruídos por diferentes codecs neurais.

Biometria de voz

Modelo utilizado para os testes foi o **Titanet** e o dataset **Voxceleb pt**

TIPO DE ÁUDIO	EER
Normal	**5%**
LFRSC	**9%**
WavTokenizer	**16%**