Aplicação de um Modelo Neural para Reconhecimento de Fala em Áudios com Características de Comunicação via Rádio ⋆

Lucas Grigoleto Scart * **Raquel Frizera Vassallo** *
Jorge Leonid Aching Samatelo *

* *Universidade Federal do Espírito Santo, ES, (e-mails: scart.lucas@gmail.com, raquel.vassalo@ufes.br, jorge.samatelo@ufes.br).*

Abstract: Automatic speech recognition is essential for machines to understand the content of words and sentences in a spoken language. Machine learning models known as deep neural networks are the focus of actual research in the artificial intelligence area, obtaining superior results compared with classical models and enabling the learning of features through unlabeled data. Despite the significant advance in applying these models to languages with a low volume of labeled data, there is still a barrier to the practical use of speech recognition models caused by the domain mismatch between training and inference data. This article proposes a methodology for simulating radio communication characteristics, enabling the development of datasets oriented to the robust training of neural models. The simulation was carried out through the implementation via *software* of a narrowband FM transmitter and receiver, together with the noisy communication channel. A state-of-the-art speech recognition architecture is also implemented and trained using advanced regularization techniques. When performing the training with the simulated data, it is was observed a relative reduction of 51.7% in the character error rate considering the most challenging noise level (SNR of 0 dB), with a similar decrease at all noise levels. We expected that the methodology developed in this work would open space to develop more robust speech recognition models with future applications in radio communication.

Resumo: O reconhecimento automático de fala é essencial para que máquinas possam compreender o conteúdo de palavras e frases em uma linguagem pronunciada. Modelos de aprendizado de máquina conhecidos como redes neurais profundas são o foco de pesquisas atuais na área de inteligência artificial, obtendo resultados superiores a outros modelos clássicos e possibilitando também o aprendizado de características por meio de dados não rotulados. Apesar do grande avanço na aplicação destes modelos, há linguagens que possuem baixo volume de dados rotulados, existindo, ainda, uma barreira para a utilização prática do reconhecimento de fala, causado pela diferença de domínio entre dados de treinamento e inferência. Neste artigo é proposta uma metodologia para a simulação de características da comunicação via rádio, possibilitando o desenvolvimento de conjuntos de dados para o treinamento robusto de modelos neurais. A simulação foi realizada através da implementação via *software* de um transmissor e receptor FM de banda estreita, em conjunto com o canal de comunicação ruidoso. Também é implementada uma arquitetura de reconhecimento de fala estado da arte, a qual é treinada usando técnicas avançadas de regularização. Ao realizar o treinamento com os dados simulados, foi observada uma redução relativa de $51,7\%$ na taxa de erro por caracteres considerando o nível de ruído mais desafiante (SNR de 0 dB), com redução semelhante em todos os níveis de ruído. Espera-se que a metodologia proposta neste trabalho abra espaço para o desenvolvimento de modelos de reconhecimento de fala mais robustos com futuras aplicações na comunicação via rádio.

Keywords: neural networks; automatic speech recognition; regularization; dataset construction; radio communication.

Palavras-chaves: redes neurais; reconhecimento automático de fala; regularização; construção de um conjunto de dados; comunicação via rádio.

1. INTRODUÇÃO

Reconhecimento automático de fala, o foco deste trabalho, é a capacidade de uma máquina em reconhecer o conteúdo das palavras e frases em uma linguagem pronunciada e transformá-las em um formato compreensível para a máquina.

Nas últimas décadas, métodos baseados em aprendizado de máquina tem se mostrado eficazes em aplicações para processamento de voz, especialmente para a tarefa de reconhecimento de fala. Porém pesquisas recentes tem se focado em utilizar um subconjunto de modelos de aprendizado de máquina conhecidas como redes neurais profundas, apresentando resultados superiores aos obtidos com a utilização de outros modelos clássicos de aprendizado.

A partir do trabalho que apresentou a metodologia *wav2vec* 2.0 por Baevski et al. (2020), ocorreu um avanço na área de reconhecimento de fala aplicado a linguagens que possuem baixo volume de dados rotulados, quando comparadas à língua inglesa. A metodologia tem como base um processo de treinamento dividido em duas etapas. Na primeira etapa é utilizado um grande volume de dados não rotulados, com um paradigma de aprendizado *self-supervised*. O modelo é restringido a produzir representações discretas de trechos do áudio, ao mesmo tempo em que deve contextualizar as representações com base no áudio. Assim, dado um conjunto de representações extraídas do áudio, incompletas no tempo, a tarefa alvo do modelo é predizer quais outras representações discretas preenchem corretamente os instantes faltantes para alcançar a representação completa do áudio. Na segunda etapa, é adicionada uma camada linear ao final do modelo, realizando o mapeamento entre a dimensionalidade do vetor discreto aprendido na etapa anterior, e o tamanho do vocabulário alvo. São utilizados então dados rotulados, e o modelo final é treinado utilizando o objetivo *Connectionist Temporal Classification* de Graves et al. (2006).

Apesar da melhoria alcançada com a utilização de dados não rotulados, em conjunto com o treinamento de grandes modelos que aprendem representações compartilhadas entre línguas diferentes por Conneau et al. (2021), ainda existe uma barreira para a utilização prática do reconhecimento de fala em português. Isso ocorre pois os dados utilizados para o treinamento contêm, em sua maioria, gravações de leituras e discursos preparados. São áudios gravados em ambientes controlados com microfones de alta qualidade, não contendo ruídos ou interferências externas, além de um tom de voz calmo e dicção clara. Em contraste, áudios extraídos de comunicação via rádio possuem um alto nível de ruído, compressão dos dados com perdas utilizada para alcançar uma grande distância de comunicação à custo da qualidade, além de falas rápidas com dicção duvidosa.

Para a obtenção de sistemas de reconhecimento automático de fala que possam ser utilizados com dados provenientes da comunicação via rádio, é proposta a adaptação de modelos existentes para que eles se tornem robustos às condições encontradas. A utilização de modelos treinados com um grande volume de dados como base permite o reaproveitamento das representações aprendidas reduzindo o volume de dados necessário. Em conjunto, é realizada a simulação do processo de transmissão de voz por sistema de rádio de forma automatizada, gerando um grande volume de dados com diversos perfis de ruído.

As principais contribuições deste trabalho são:

- Simulação via *software* da comunicação via rádio por sistema de transmissão FM de banda estreita com canal ruidoso;
- Desenvolvimento de um grande conjunto de dados com diferentes condições de ruído;
- Implementação de um modelo de reconhecimento de fala robusto utilizando uma arquitetura estado da arte e treinamento do mesmo usando avançadas técnicas de regularização.

O restante do artigo está estruturado da seguinte maneira. A Seção 2 discute trabalhos relacionados, apresentando os principais conjuntos de dados, modelos de reconhecimento de fala em português e sistemas de reconhecimento de fala robustos. A Seção 3 detalha a simulação da comunicação via rádio proposta e o modelo estado da arte utilizado. A Seção 4 descreve os experimentos realizados. A Seção 5 apresenta os resultados obtidos, e um estudo comparativo com resultados da literatura. Por fim, a Seção 6 conclui o trabalho.

2. TRABALHOS RELACIONADOS

2.1 Conjunto de dados

Parte essencial para o treinamento de modelos de aprendizado de máquina, os conjuntos de dados voltados para o reconhecimento automático de fala em português vêm crescendo em volume de forma acelerada nos últimos anos. Em 2019, existiam apenas quatro conjuntos de dados disponíveis abertamente totalizando cerca de 60 horas. Atualmente, se tem mais de 500 horas disponíveis abertamente, com dados extraídos de diversas fontes com gêneros que variam desde à audiolivros até palestras. Dentre os maiores conjuntos de dados, se pode destacar:

O conjunto de dados CETUC, Alencar and Alcaim (2008) contém 145 horas de áudio, com um total de 100 locutores sendo metade de cada sexo. É utilizado um conjunto de 1.000 frases distintas, onde cada locutor fala todas as frases uma única vez. O áudio foi gravado em ambiente controlado, com uma taxa de amostragem de 16 kHz.

O *Multilingual LibriSpeech* (MLS), Pratap et al. (2020) é composto por áudios extraídos de audiolivros do projeto *Librivox*, contendo gravações pertencentes ao domínio público. Estão presentes áudios em oito idiomas diferentes, sendo que o português corresponde a 160 horas de duração, com 36 locutores do sexo masculino e 26 locutores do sexo feminino.

⋆ Os autores agradecem à Empresa Vale S.A. pelo apoio financeiro dado através do Projeto 889/4600061953 "Desenvolvimento de uma Ferramenta de Software Orientada à Auditoria de Comunicação Via Rádio".

O conjunto *Multilingual TEDx*, Salesky et al. (2021) é composto por gravações de palestras *TEDx* em 8 idiomas. O português brasileiro corresponde a 164 horas de duração.

O projeto *Common Voice*, Ardila et al. (2020) busca gerar dados para sistemas de reconhecimento de fala disponíveis abertamente para o maior número de idiomas possível. Para isso, voluntários utilizam o site ou aplicativo para doar a sua voz, por meio da leitura de frases preparadas. Em outra parte do ambiente, é possível realizar a validação dos dados submetidos por usuários. Com atualizações do conjunto de dados público ocorrendo algumas vezes ao ano, a sua versão 7.0 disponibilizada em julho de 2021 contém 84 horas validadas no idioma português, com um total de 2038 oradores. Como não possui controle de ambiente, os áudios possuem variados níveis de ruído e qualidade.

Por fim, o conjunto de dados mais recente a ser disponibilizada é a *Corpus of Annotated Audios* (CORAA), Junior et al. (2021). Contendo 290 horas de áudio, ela tem como foco áudios contendo trechos de conversas, ao contrário das frases preparadas e narrações utilizadas em outras bases de dados.

2.2 Reconhecimento de fala em português

Junto com a criação de novos conjuntos de dados, existe o desenvolvimento de sistemas de reconhecimento automático de fala. Tendo como base o conjunto de ferramentas *Kaldi*, que utiliza abordagens tradicionais quando comparado às redes neurais profundas, os autores de Batista et al. (2018) desenvolveram modelos base para a língua portuguesa com a utilização de vários conjuntos de dados abertos e pagos, sendo o maior dentre eles o CETUC. Utilizando modelos baseados em cadeias de markov e misturas de gaussianas, foi alcançada uma taxa de erro por palavras de $6,5\%$. Como prova de conceito, é treinada uma rede neural profunda para realizar o papel de modelo acústico, reduzindo a taxa de erro para $4,5\%$, porém ainda ocorre a utilização do dicionário fonético e modelo de linguagem dos sistemas tradicionais.

Um marco importante no desenvolvimento de redes neurais profundas que realizam o reconhecimento de fala de ponta a ponta é o trabalho de Quintanilha (2017). Utilizando uma combinação de conjuntos de dados menores, alcançando cerca de 13 horas no total, é treinando um modelo *end-to-end* alcançando uma taxa de erro por caracteres de $25,13\%$. Posteriormente, Quintanilha et al. (2020) expandiu o seu trabalho anterior, construindo um conjunto de dados com 158 horas no total. Utilizando uma arquitetura baseada no *DeepSpeech*-2, conjuntamente com modelos de linguagem para o pós-processamento das predições, foi alcançada uma taxa de erro por caracteres de $10,49\%$.

Com o surgimento do *Wav2vec* 2.0, o primeiro trabalho focado no português foi o de Gris et al. (2021). Utilizando apenas uma hora de dados rotulados, foi obtida uma taxa de erro por palavras de 34% quando avaliado com o conjunto de testes do *commonvoice*. Este trabalho foi expandido em Stefanel Gris et al. (2022), com a utilização de 427 horas de áudio para o treinamento do modelo alcançada pela união de diferentes conjuntos de dados. Como resultado, foi apresentada uma taxa de erro por palavras média de $12,4\%$, quando avaliada sobre sete conjuntos de dados diferentes. Com a utilização de modelos de linguagem realizando o pós-processamento das predições, a taxa de erro média sofreu uma redução para $10,5\%$.

Simultaneamente à publicação do conjunto de dados CORAA, o trabalho de Junior et al. (2021) realizou o treinamento de um modelo linha de base utilizando os dados novos juntamente com o *wav2vec* 2.0. Quando comparado ao resultado de Stefanel Gris et al. (2022), o modelo treinado possui maior taxa de erro quando avaliado sobre o conjunto de testes original do *commonvoice*, porém apresenta melhores resultados com os novos dados.

2.3 Reconhecimento de fala robusto

Uma alternativa para diminuir o efeito que a mudança de domínio possui na qualidade do modelo é o treinamento de modelos resistentes a variações, utilizando diretamente dados do domínio alvo. Essa é a abordagem adotada por *robust-wav2vec*, Hsu et al. (2021) e *wav2vec-switch*, Wang et al. (2021b). Enquanto o primeiro utiliza apenas dados ruidosos para o treinamento, o segundo requer a utilização de pares limpo - ruidoso. Ambas metodologias permitem a utilização de dados não rotulados para a melhoria dos modelos.

Outra linha de pesquisa busca realizar o pré-processamento dos áudios com o objetivo de reduzir o ruído, como um passo anterior ao modelo de reconhecimento de fala Van Segbroeck and Narayanan (2013); Yoshioka and Gales (2015); Kinoshita et al. (2020); O'Malley et al. (2021). Aqui, o processo de treinamento busca extrair apenas a informação relevante dos áudios ruidosos, tentando torná-los mais próximos aos dados utilizados para treinamento do modelo.

Existe ainda a possibilidade de realizar a transferência de aprendizado, a partir do reaproveitamento de modelos treinados com um grande volume de dados e a utilização de *data augmentation*. Narayanan et al. (2018) mostrou que ao realizar o treinamento com um grande volume de dados de diferentes domínios auxilia na capacidade de generalização do modelo, reduzindo a quantidade de dados necessários para a utilização em um novo domínio. É utilizada ainda a simulação de condições adversas, com a adição de ruídos, diferentes larguras de banda e aplicação de *codecs* para a compressão com perdas do áudio. Já em Balam et al. (2020), ocorreu a transferência de conhecimento utilizando um modelo treinado previamente apenas com dados limpos. Novamente foi utilizada de forma extensiva a simulação de condições adversas, adicionando agora a resposta ao impulso para reproduzir as características acústicas de diferentes ambientes. Luo et al. (2021) e Pol'ak and Bojar (2021) mostraram a efetividade da utilização de modelos pré treinados como base para modelos mais robustos, com variações de sotaque, idioma e domínio de aplicação.

Quando aplicado à comunicação via rádio no idioma português, temos como referência o trabalho de Duarte and Colcher (2021). Tendo como base o conjunto de dados *commonvoice*, foram construídos novos conjuntos de dados por meio de quatro cenários diferentes: (i) adição de ruído gaussiano branco, (ii) adição de ruído coletado de antenas

ao vivo, (*iii*) simulação via *software* do canal de transmissão e (*iv*) simulação via *hardware* do canal de transmissão.

As principais diferenças para este trabalho são a utilização de arquitetura de reconhecimento de fala estado da arte, a transferência de conhecimento a partir de modelos treinados previamente com grande volume de dados e a simulação do canal de transmissão por meio de uma biblioteca de código que fornece uma interface de programação permitindo a automação do processamento.

3. PROPOSTA

3.1 Construção do conjunto de dados de áudios com características de comunicação via rádio

O GNU Rádio, Blossom (2004), é uma biblioteca de código que possui um conjunto de blocos implementando diferentes modulações e modelos de canal. Embora o seu objetivo principal seja a criação de dispositivos de rádio definido por software, ela pode ser utilizada para a modelagem de sistemas de telecomunicação complexos permitindo a simulação da transmissão do sinal de voz em condições adversas.

Foram implementados o transmissor e receptor FM de banda estreita, devido à sua aplicação em rádios comunicadores padrão. A entrada do transmissor consiste em áudios salvos em disco, em formato wav com taxa de amostragem de 16 kHz. O primeiro bloco utilizado é um filtro passa-faixa, que limita as frequências presentes no sinal de entrada para o intervalo de 300 Hz a 3400 Hz, o qual é normalmente utilizado para sinais puros de voz. O filtro ainda possui ganho unitário, e uma banda de transição de 200 Hz. A seguir é utilizado o bloco *NBFM Transmit*, o qual é responsável por transformar as amostras de áudio para a representação complexa do sinal com modulação em frequência. Por se uma transmissão do tipo banda estreita, este bloco possui um filtro passa-baixa interno com uma frequência de corte de 4,5 kHz. Para a escolha da frequência da portadora utilizada na modulação, foi escolhido o critério de 4 vezes o valor da taxa de amostragem do sinal, resultando numa portadora em 64 kHz. A seguir é utilizada uma interpolação de três vezes, aumentando a frequência do sinal transmitido para 192 kHz.

O receptor tem como primeiro bloco um filtro passa-faixa, centrado em 192 kHz e com uma janela de 6 kHz em torno desta frequência. Este filtro ainda é responsável por realizar a dizimação do sinal em três vezes, gerando como saída um sinal de 64 kHz. A seguir é utilizado o *NBFM Receive*, que transforma o sinal complexo de volta à sua representação real. Ao recuperar o sinal modulado em frequência, obtemos a forma de onda final com taxa de amostragem de 16 kHz.

Entre transmissor e receptor é utilizado o modelo de canal simples. Por meio deste canal, são simulados dois efeitos, a presença de ruído aditivo gaussiano branco e um desvio de frequência que pode ocorrer durante o processo de transmissão. Ainda como parte do GNU Rádio, existem outros modelos de canal de maior complexidade que podem realizar a simulação de propagação multivias, efeito dopler e variações das distorções em função do tempo.

3.2 Modelo de reconhecimento automático de fala

A abordagem do *wav2vec* realiza a codificação do áudio por meio de uma rede neural profunda convolucional que recebe como entrada um áudio puro e gera uma sequência de vetores que formam a representação latente, com uma janela de T amostras do áudio para cada vetor. É aplicado um módulo de quantização aos vetores latentes, gerando então uma representação quantizada do áudio que era originalmente contínuo. Em seguida é aplicada uma máscara às representações latentes obtidas. As representações latentes são então processadas pela utilização de um modelo *transformer* por Vaswani et al. (2017) que constrói representações contextualizadas capturando informação de toda a sequência. O modelo é treinado por completo utilizando uma tarefa contrastiva, onde a representação latente verdadeira deve ser identificada para os instantes aos quais foram aplicadas as máscaras.

O modelo utilizado possui três componentes principais. O primeiro módulo consiste no extrator de características, que recebe como entrada o áudio original e utiliza uma rede neural profunda composta por diversos blocos contendo camadas convolucionais, seguidas por uma camada de normalização e função de ativação GELU. O áudio de entrada possui apenas um processamento, no qual é realizada a normalização para que tenha média zero e variância unitária.

Já o segundo módulo consiste em um quantizador. Ele recebe como entrada as representações latentes, e realiza o mapeamento para um conjunto finito de representações discretas por meio de quantização de produto Jegou et al. (2010). São utilizados G grupos de representações, cada um deles com V entradas. Para encontrar a representação discreta de um vetor latente, é escolhida uma entrada de cada grupo, sendo realizado o concatenamento das G representações e aplicada uma transformação linear $\mathbb{R}^d \mapsto \mathbb{R}^f$ para obter o vetor discreto $q \in \mathbb{R}^f$.

O processo de escolha das entradas discretas é realizado pela operação de *Gumbel Softmax*. A probabilidade de se escolher a entrada v para o grupo g é dada pela Equação 1.

$$p_{g,v} = \frac{exp(l_{g,v} + n_v)/\tau}{\sum_V^{k=1} exp(l_{g,v} + n_k)/\tau}. \quad (1)$$

Onde: $l_{g,v}$ é o resultado de se multiplicar os vetores latentes por uma matriz de quantização; τ é um fator de temperatura positivo; $n = -log(-log(u))$, sendo que $u \sim U(0, 1)$.

Por fim, o último módulo consiste em um modelo *transformer*. Ele segue a arquitetura *transformer* Vaswani et al. (2017), onde cada camada é formada pela sequência de operações: (*i*) codificação de posição, (*ii*) auto-atenção, (*iii*) normalização por camada e (*iv*) projeção linear. Diferentemente da arquitetura original, que realiza uma codificação de posição de forma global tendo problemas para longas sequências como é o caso de sinais de áudio, é utilizada como alternativa uma codificação relativa que opera de forma local à entrada. A codificação de posição é essencial para que seja mantida a ordenação ao se utilizar o modelo *transformer*, pois a camada de auto-atenção, que

permite a cada elemento da sequência de entrada interagir com outros elementos da sequência, não preserva a ordem temporal da mesma.

O primeiro treinamento do modelo define uma tarefa de aprendizagem contrastiva. Inicialmente uma máscara aleatória é aplicada a cerca de 50% dos vetores latentes, onde eles são substituídos por um vetor especial z_m. Após isso os vetores latentes são processados pelo modelo *transformer*, gerando a representação contextual para todos os estados, incluindo àqueles que foram mascarados. Para cada posição mascarada então, são amostradas 100 representações latentes incorretas. A tarefa de aprendizagem compara a similaridade cosseno entre o vetor contextual c, o vetor latente verdadeiro q_p e os vetores latentes incorretos q_n. A função de custo então maximiza a similaridade entre c e q_p, enquanto penaliza a similaridade entre c e q_n.

Existe uma segunda componente da função de custo utilizada durante esta etapa de treinamento, que promove a diversidade entre as representações. Ela funciona ao maximizar a entropia média da distribuição gerada pelas representações discretas, forçando o modelo a utilizar toda a capacidade de representação de forma igual.

Depois de ter sido realizado o primeiro treinamento, se tem um modelo base que encontra representações compactas do áudio que guardam importante informação contextual. Para extrair informação linguística dessa representação, é realizado o processo de *fine-tuning*. Primeiramente é adicionada uma nova camada linear ao final do modelo, realizando o mapeamento da dimensionalidade do vetor de contexto, para a dimensionalidade do alfabeto alvo. A seguir, é realizado o treinamento de forma supervisionada utilizando conjuntos de dados rotuladas.

Porém, existem alguns problemas ao utilizar os dados conjuntamente com técnicas de aprendizado de máquina. Primeiramente, não é conhecido o alinhamento entre os caracteres de saída e a parte correspondente no áudio de entrada, visto que obter tal informação adiciona um grau de complexidade durante o processo de rotulação que impossibilita a construção de grandes conjuntos de dados. Além disso, ambas a entrada e saída podem variar em tamanho, assim como a proporção entre o tamanho deles.

O algoritmo *Connectionist Temporal Classification*, Graves et al. (2006) é uma forma de lidar com a falta do alinhamento entre a entrada e a saída para encontrar a probabilidade de que ocorra uma sequência específica $\mathbf{Y}$ dada uma entrada $\mathbf{X}$. Para isso, o algoritmo realiza a soma das probabilidades de todos os possíveis alinhamentos entre $\mathbf{X}$ e $\mathbf{Y}$. Primeiramente, é adicionado ao vocabulário um símbolo ϵ, correspondente a uma predição em branco. Dado um vetor de predições do modelo, que possui tanto caracteres do vocabulário alvo quanto o símbolo especial, são removidos os caracteres repetidos e, logo após, são removidos os símbolos especiais ϵ para se obter a saída correspondente.

Realizar esse mapeamento entre diferentes alinhamentos para a mesma sequência de rótulos é o que viabiliza ao CTC utilizar dados sem alinhamento prévio, pois ele possibilita que o modelo realize predições sem saber onde elas ocorrem.

4. EXPERIMENTOS

4.1 Bibliotecas de código

Para o treinamento do modelo de reconhecimento automático de fala, foi utilizada a biblioteca *thunder-speech* [1] . Ela foi desenvolvida para facilitar o treinamento de redes neurais profundas aplicadas especificamente ao problema de reconhecimento de fala. Tendo como base o *framework Pytorch* por Paszke et al. (2017), é definida uma interface unificada para a representação de todo o sistema que engloba a rede neural. Dado um modelo base, devem ser definidas cinco partes para formar o módulo de treinamento: (i) classe que realiza o pré-processamento do áudio, incluindo a extração de características, (ii) *encoder*, parte do modelo responsável por extrair características de alto nível e que normalmente é reaproveitado, (iii) *decoder*, o qual é responsável por transformar as características de alto nível em uma sequência de probabilidades representando a predição do modelo, (iv) classe responsável pelas transformações do texto, guardando o vocabulário específico e decodificando as probabilidades para a representação textual adequada, e (v) otimizador utilizado para o treinamento.

A partir da interface, são definidas camadas de compatibilidade que permitem a importação de modelos de outras bibliotecas. A implementação do *wav2vec* utilizada neste artigo tem como fonte o *huggingface* de Wolf et al. (2020), que é referência na implementação de modelos que utilizam a arquitetura *transformer*.

4.2 Detalhes de implementação

Devido ao alto custo de treinar o modelo *wav2vec* por completo, foi realizado apenas o procedimento de *fine-tuning* a partir de um modelo treinado previamente e disponibilizado abertamente. Foi escolhido o modelo identificado por "facebook/wav2vec2-base-100k-voxpopuli"na biblioteca huggingface. Ele corresponde ao *wav2vec* em sua configuração de tamanho base, a qual contém 95 milhões de parâmetros, e que passou pela etapa de aprendizagem contrastiva utilizando o conjunto de dados *VoxPopuli* de Wang et al. (2021a), com 100 mil horas de áudio, incluindo 4.400 horas de português de Portugal coletadas de gravações do parlamento europeu.

Como otimizador foi escolhido o algoritmo *AdamW*, Loshchilov and Hutter (2019). Foram criados dois grupos de parâmetros, o primeiro contendo o *encoder* treinado previamente, enquanto o segundo grupo contém o *decoder* que foi inicializado aleatoriamente com base no vocabulário alvo. No início do treinamento, apenas o *decoder* é atualizado enquanto o *encoder* se mantém constante. A taxa de aprendizado inicial é de 0.03, e possui um decaimento linear de modo a atingir 10% do seu valor inicial ao fim do treino. Após a primeira época, os parâmetros do *encoder* são adicionados ao otimizador, de modo que todo o modelo seja treinado. Para que ocorra estabilidade no treinamento, a taxa de aprendizagem para os parâmetros do *encoder* é sempre 1.000 vezes menor do que o valor utilizado no *decoder*, seguindo o mesmo decaimento linear. Os outros parâmetros do otimizador seguem o padrão do *Pytorch*, sendo os betas iguais a $(0.9, 0.999)$ e o *weight decay* de 0.01.

[1] Disponível em https://github.com/scart97/thunder-speech

Foi utilizado um *batch* de 10 elementos, com o processo de acumulação de gradientes por 4 passos.

De forma a reduzir a quantidade de memória de vídeo necessária para o treinamento, foi utilizado o recurso de precisão mista automática durante o treinamento, Micikevicius et al. (2018). Ele funciona ao realizar certas operações de multiplicação de matrizes utilizando o formato numérico de ponto flutuante com 16 bits, enquanto os valores que precisam ser armazenados com grande precisão numérica seguem o padrão de armazenamento utilizando 32 bits.

Para a regularização do modelo, foram utilizados em conjunto *dropout*, *layerdrop* e *specaugment*. O *dropout* é uma técnica de regularização que modifica a rede neural durante o treinamento, por meio da eliminação aleatória de uma porcentagem dos neurônios ocultos da rede. O processo de eliminação significa a remoção temporária do neurônio, em conjunto com a sua conexão de entrada e saída. A cada iteração do treinamento é gerada uma máscara aleatória que define quais neurônios serão eliminados, seguindo uma probabilidade p definida como parâmetro do treinamento. Já o *layerdrop* pode ser visto como uma extensão do *dropout*, onde todos os neurônios de uma camada são eliminados aleatoriamente e funcionam como uma operação de identidade, o que é equivalente a treinar um modelo que possui profundidade variável. Por fim o *specaugment* é aplicado diretamente às características de entrada da rede, gerando máscaras aleatórias que removem blocos contínuos tanto em frequência quanto no tempo. Os hiper-parâmetros utilizados se encontram na Tabela 1.

Tabela 1. Hiper-parâmetros utilizados para a regularização.

hiper-parâmetro	valor
dropout	0.1
layerdrop	0.1
mask feature probability	0.1
mask time probability	0.2

Para a criação do conjunto de dados utilizando a simulação descrita na Seção 3.1, precisam ser definidos dois hiper-parâmetros que controlam o canal de comunicação. O primeiro deles é a relação sinal ruído, indicando a potência do sinal gaussiano branco que modelo o canal *AWGN*. Foram escolhidos cinco valores que representam o aumento incremental no nível de ruído: $[20, 10, 5, 3, 0]$ dB. Já o segundo parâmetro controla o desvio de frequência causado pela propagação no canal, e foram utilizados apenas dois valores: nenhuma distorção, ou $0,5\%$ de distorção. Valores maiores de distorção causam a perda de sintonia, de forma que o sinal de voz não é recuperado pelo receptor.

São criadas então dez novas versões de cada áudio e salvas em disco, contendo todas as combinações possíveis entre os dos dois hiper-parâmetros. Durante o processo de treinamento do modelo específico, ao ser carregado um arquivo de áudio é escolhida de forma aleatória entre o original e as novas versões, com a mesma probabilidade para todas as onze versões disponíveis. Isso garante que cada áudio seja visto somente uma vez durante uma época de treinamento, porém ocorra variabilidade entre diferentes épocas [2] .

4.3 Métricas

Para a avaliação do sistema proposto neste trabalho, a métrica utilizada será a taxa de erro por caractere (CER). Ela pode ser computada de acordo com a Equação 2.

$$CER = \frac{S + D + I}{N} \tag{2}$$

Onde: S é o número de substituições realizadas, D é o número de deleções, I é o número de inserções e N é o número de caracteres presentes na referência.

5. RESULTADOS E DISCUSSÃO

Para a avaliação da metodologia proposta foram realizados dois treinamentos, que utilizam os mesmos parâmetros descritos na Seção 4.2 porém conjuntos de dados diferentes. O primeiro modelo obtido, identificado como base, utiliza os dados do *commonvoice* originais. Já o modelo específico é treinado utilizando os dados resultantes da simulação de comunicação via rádio. Ambos modelos foram treinados utilizando uma única *GPU Titan V* com 12 GB de memória de vídeo.

Foram realizadas então uma série de avaliações de ambos os modelos obtidos. O primeiro caso apresenta o conjunto de testes na qualidade original, com o objetivo de propiciar uma comparação com trabalhos relacionados. Na Tabela 2 é possível visualizar os resultados obtidos. Ambos os modelos treinados neste trabalho apresentam uma taxa de erro superior quando comparados a outros trabalhos que também utilizam a arquitetura *wav2vec*, porém são inferiores ao erro obtido por Duarte and Colcher (2021).

Tabela 2. Taxa de erro por caractere utilizando o conjunto de teste em qualidade original.

Modelo	Taxa de erro
Stefanel Gris et al. (2022)	4,15
Junior et al. (2021)	6,34
Duarte and Colcher (2021) [3]	30,0
Modelo base	8,63
Modelo específico	9,80

A taxa de erro obtida se encontra dentro do esperado, considerando o volume de dados utilizados. Enquanto este trabalho e Duarte and Colcher (2021) utilizam apenas o conjunto de dados *commonvoice* para o treinamento, os dois melhores resultados da tabela foram treinados com um volume maior de dados, sendo que Stefanel Gris et al. (2022) utiliza uma união de sete conjuntos de dados distintos para o treinamento. Também é possível notar que o treinamento utilizando dados simulados acarretou em um aumento relativo de $13,5\%$ na taxa de erro quando avaliado utilizando os dados originais.

As próximas avaliações utilizam o conjunto de testes do *commonvoice* com modificações controladas utilizando a

[2] Código utilizado para o treinamento disponível em https://github.com/scart97/radio_speech

[3] Valor estimado a partir da Figura 13 de Duarte and Colcher (2021)

simulação de rádio. Na Tabela 3 temos os resultados para diferentes níveis de relação sinal ruído, no caso em que não ocorre o desvio de frequência no canal de transmissão. O modelo base possui uma alta taxa de erro em todas as situações. Mesmo com um baixo nível de ruído, a quantidade limitada de informação espectral possui um grande efeito na qualidade das predições.

Tabela 3. Taxa de erro por caractere aplicando a simulação de rádio, sem desvio de frequência.

SNR (dB)	20	10	5	3	0
Modelo base	23,60	33,32	40,43	43,59	48,59
Modelo específico	13,54	16,51	19,43	20,9	23,57

Na Tabela 4 temos os resultados quando é aplicado o desvio de frequência, em conjunto com o ruído gaussiano no canal de transmissão. Para todos os níveis de ruído ocorre uma taxa de erro mais elevada, quando comparado com os resultados da tabela anterior. Porém, o efeito maior ocorre sobre o modelo base, mostrando a robustez desenvolvida pelo modelo específico.

Tabela 4. Taxa de erro por caractere aplicando a simulação de rádio, com desvio de frequência.

SNR (dB)	20	10	5	3	0
Modelo base	26,31	35,41	41,99	44,87	49,73
Modelo específico	13,88	16,91	19,85	21,33	24,29

6. CONCLUSÃO

Através do reconhecimento automático de fala, é possível transformar o conteúdo das palavras e frases de uma linguagem pronunciada na sua respectiva forma escrita. Com a utilização de redes neurais profundas, em especial a metodologia *wav2vec* 2.0, ocorreu um grande avanço na aplicação do reconhecimento de fala a linguagens que possuem baixo volume de dados. Porém, a diferença de domínio entre os dados de treinamento e inferência cria uma barreira para a utilização prática desses sistemas. Em busca de modelos robustos que possam ser aplicados a comunicação via rádio, foi desenvolvida uma simulação via *software* das características específicas desse tipo de comunicação, para a criação de conjuntos de dados. Por meio do treinamento de modelos que realizam o reconhecimento de fala de ponta a ponta, com a utilização de técnicas avançadas de regularização, foi observada uma redução expressiva na taxa de erro por caracteres em todos os níveis de ruído, com a redução relativa de $51,7\%$ no nível de ruído mais desafiante (SNR de 0dB). Os resultados obtidos abrem espaço para a implementação de sistemas de reconhecimento de fala aplicados na comunicação via rádio. Como trabalho futuro, pretende-se utilizar um volume maior de dados para o treinamento do modelo, a partir da união de múltiplos conjuntos de dados e aplicação das técnicas de simulação desenvolvidas.

REFERÊNCIAS

Alencar, V.F.S. and Alcaim, A. (2008). Lsf and lpc - derived features for large vocabulary distributed continuous speech recognition in brazilian portuguese. In *2008 42nd Asilomar Conference on Signals, Systems and Computers*, 1237–1241. doi:10.1109/ACSSC.2008.5074614.

Ardila, R., Branson, M., Davis, K., Kohler, M., Meyer, J., Henretty, M., Morais, R., Saunders, L., Tyers, F., and Weber, G. (2020). Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, 4218–4222. European Language Resources Association, Marseille, France. URL `https://www.aclweb.org/anthology/2020.lrec-1.520`.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, 12449–12460. Curran Associates, Inc. URL `https://proceedings.neurips.cc/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf`.

Balam, J., Huang, J., Lavrukhin, V., Deng, S., Majumdar, S., and Ginsburg, B. (2020). Improving noise robustness of an end-to-end neural model for automatic speech recognition. doi:10.48550/ARXIV.2010.12715. URL `https://arxiv.org/abs/2010.12715`.

Batista, C., Dias, A.L., and Sampaio Neto, N. (2018). Baseline Acoustic Models for Brazilian Portuguese Using Kaldi Tools. In *Proc. IberSPEECH 2018*, 77–81. doi:10.21437/IberSPEECH.2018-17.

Blossom, E. (2004). Gnu radio: Tools for exploring the radio frequency spectrum. *Linux J.*, 2004(122), 4.

Conneau, A., Baevski, A., Collobert, R., Mohamed, A., and Auli, M. (2021). Unsupervised cross-lingual representation learning for speech recognition. In *Interspeech 2021*, 2426–2430. doi:10.21437/Interspeech.2021-329.

Duarte, J.C. and Colcher, S. (2021). Building a noisy audio dataset to evaluate machine learning approaches for automatic speech recognition systems. *Monografias em Ciência da Computação*, nº 05/2021.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural 'networks. In *ICML 2006 - Proceedings of the 23rd International Conference on Machine Learning*, volume 2006, 369–376. doi:10.1145/1143844.1143891.

Gris, L.R.S., Casanova, E., de Oliveira, F.S., da Silva Soares, A., and Candido-Junior, A. (2021). Desenvolvimento de um modelo de reconhecimento de voz para o português brasileiro com poucos dados utilizando o wav2vec 2.0. In *Anais do XV Brazilian e-Science Workshop*, 129–136. SBC.

Hsu, W.N., Sriram, A., Baevski, A., Likhomanenko, T., Xu, Q., Pratap, V., Kahn, J., Lee, A., Collobert, R., Synnaeve, G., and Auli, M. (2021). Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training.

Jegou, H., Douze, M., and Schmid, C. (2010). Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1), 117–128.

Junior, A.C., Casanova, E., Soares, A., de Oliveira, F.S., Oliveira, L., Junior, R.C.F., da Silva, D.P.P., Fayet, F.G., Carlotto, B.B., Gris, L.R.S., and Aluísio, S.M. (2021). Coraa: a large corpus of spontaneous and prepared speech manually validated for speech recognition in brazilian portuguese. doi:10.48550/ARXIV.2110.15731. URL `https://arxiv.org/abs/2110.15731`.

Kinoshita, K., Ochiai, T., Delcroix, M., and Nakatani, T. (2020). Improving noise robust automatic speech recognition with single-channel time-domain enhancement network. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7009–7013. doi:10.1109/ICASSP40776.2020.9053266.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Luo, J., Wang, J., Cheng, N., Xiao, E., Xiao, J., Kucsko, G., O'Neill, P., Balam, J., Deng, S., Flores, A., Ginsburg, B., Huang, J., Kuchaiev, O., Lavrukhin, V., and Li, J. (2021). Cross-language transfer learning and domain adaptation for end-to-end automatic speech recognition. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6. doi:10.1109/ICME51207.2021.9428334.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., and Wu, H. (2018). Mixed precision training. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=r1gs9JgRZ.

Narayanan, A., Misra, A., Sim, K.C., Pundak, G., Tripathi, A., Elfeky, M., Haghani, P., Strohman, T., and Bacchiani, M. (2018). Toward domain-invariant speech recognition via large scale training. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, 441–447. IEEE.

O'Malley, T., Narayanan, A., Wang, Q., Park, A., Walker, J., and Howard, N. (2021). A conformer-based asr frontend for joint acoustic echo cancellation, speech enhancement and speech separation. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 304–311. doi:10.1109/ASRU51503.2021.9687942.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch. In *NeurIPS Autodiff Workshop*.

Pol'ak, P. and Bojar, O. (2021). Coarse-to-fine and cross-lingual asr transfer. In *ITAT*.

Pratap, V., Xu, Q., Sriram, A., Synnaeve, G., and Collobert, R. (2020). Mls: A large-scale multilingual dataset for speech research. *Interspeech 2020*. doi:10.21437/interspeech.2020-2826. URL http://dx.doi.org/10.21437/Interspeech.2020-2826.

Quintanilha, I.M. (2017). End-to-end speech recognition applied to brazilian portuguese using deep learning. *MSc dissertation*.

Quintanilha, I.M., Netto, S.L., and Biscainho, L.W.P. (2020). An open-source end-to-end asr system for brazilian portuguese using dnns built from newly assembled corpora. *Journal of Communication and Information Systems*, 35(1), 230–242.

Salesky, E., Wiesner, M., Bremerman, J., Cattoni, R., Negri, M., Turchi, M., Oard, D.W., and Post, M. (2021). The multilingual tedx corpus for speech recognition and translation. *CoRR*, abs/2102.01757. URL https://arxiv.org/abs/2102.01757.

Stefanel Gris, L.R., Casanova, E., de Oliveira, F.S., da Silva Soares, A., and Candido Junior, A. (2022). Brazilian portuguese speech recognition using wav2vec 2.0. In *Computational Processing of the Portuguese Language: 15th International Conference, PROPOR 2022, Fortaleza, Brazil, March 21–23, 2022, Proceedings*, 333–343. Springer-Verlag, Berlin, Heidelberg. doi:10.1007/978-3-030-98305-5_31. URL https://doi.org/10.1007/978-3-030-98305-5_31.

Van Segbroeck, M. and Narayanan, S.S. (2013). A robust frontend for asr: Combining denoising, noise masking and feature normalization. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 7097–7101. doi:10.1109/ICASSP.2013.6639039.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., and Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wang, C., Riviere, M., Lee, A., Wu, A., Talnikar, C., Haziza, D., Williamson, M., Pino, J., and Dupoux, E. (2021a). VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 993–1003. Association for Computational Linguistics, Online. doi:10.18653/v1/2021.acl-long.80. URL https://aclanthology.org/2021.acl-long.80.

Wang, Y., Li, J., Wang, H., Qian, Y., Wang, C., and Wu, Y. (2021b). Wav2vec-switch: Contrastive learning from original-noisy speech pairs for robust speech recognition.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Association for Computational Linguistics, Online. doi:10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Yoshioka, T. and Gales, M. (2015). Environmentally robust asr front-end for deep neural network acoustic models. *Computer Speech & Language*, 31(1), 65–86. doi:https://doi.org/10.1016/j.csl.2014.11.008. URL https://www.sciencedirect.com/science/article/pii/S0885230814001259.