A Data-Centric Approach for Portuguese Speech Recognition: Language Model And Its Implications

João Paulo Reis Alvarenga, Luiz Henrique de Campos Merschmann
and Eduardo José da Silva Luz

***Abstract*—Recent advances in Automatic Speech Recognition have made it possible to achieve a quality never seen before in the literature, both for languages with abundant data, such as English, which has a large number of studies, and for the Portuguese language, which has a more limited amount of resources and studies. The most recent advances address speech recognition problems with *Transformers* based models, which have the capability to perform the speech recognition task directly from the raw signal, without the need for manual feature extraction. Some studies have already shown that it is possible to further improve the quality of the transcription of these models using language models within the decoding stage, however, the real impact of such language models is still not clear, especially for the Brazilian Portuguese scenario. Also, it is known that the quality of the data used for training the models is of paramount importance, however, there are few works in the literature addressing this issue. This work explores the impact of language models applied to Portuguese speech recognition both in terms of data quality and computational performance, with a data-centric approach. We propose an approach to measure similarity between datasets and, thus, assist in decision-making during training. The approach indicates paths for the advancement of the state-of-the-art aiming at Portuguese speech recognition, showing that it is possible to reduce the size of the language model by 80% and still achieve error rates around 7.17% for the Common Voice dataset. The source code is available at https://github.com/joaoalvarenga/language-model-evaluation.**

***Index Terms*—automatic speech recognition, language model, brazilian portuguese, wav2vec2, KenLM.**

I. Introdução

A tarefa transformação de sinais sonoros em representações textuais, chamada de reconhecimento automático de fala (RAF), é um tópico de pesquisa e estudo para diferentes idiomas. Dentre eles, os estudos voltados para língua inglesa, como por exemplo [1], [2], se destacam por alcançarem baixas taxas de erro (2,9% a 1,8% de *Word Error Rate* (WER)).

O cenário da língua inglesa conta com vasto recurso disponível publicamente para o treinamento, como o conjunto *LibriSpeech* [3] que contém 960h de áudio transcrito. Entretanto para a língua portuguesa não existe um conjunto de dados como esse, tornando-se um problema para a qualidade dos modelos em língua portuguesa.

As prospostas de sistemas de RAF presentes na literatura se encaixam em diferentes categorias, tais como sistemas de reconhecimento de fala supervisionados, semi-supervisionados, não-supervisionados, híbridos e sistemas de ponta a ponta.

João Paulo Reis Alvarenga is with the PPGCC, Universidade Federal de Ouro Preto, Ouro Preto, MG, 35400-000, Brazil.

Eduardo J. da S. Luz is with Computing Department, Universidade Federal de Ouro Preto, Ouro Preto, MG, 35400-000, Brazil.

Luiz H. de C. Merschmann is with Department of Applied Computing, Universidade Federal de Lavras, Lavras, MG, 37200-000, Brazil.

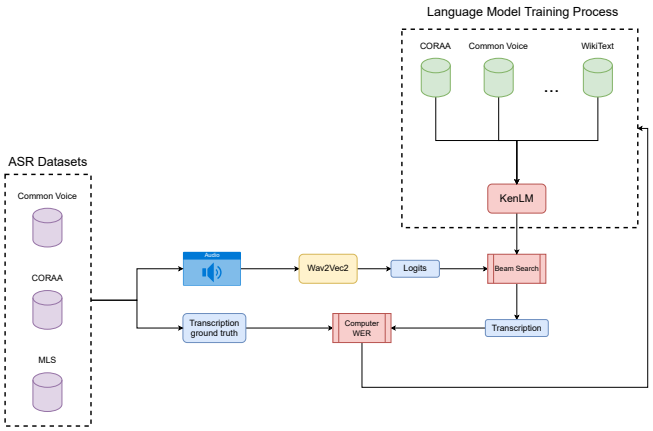


Fig. 1. Ilustração da metodologia proposta para avaliação das técnicas em abordagem centrada em dados. Fonte: Autor.

Os sistemas supervisionados são aqueles que necessitam de um conjunto de pares de áudio e transcrição para o treinamento dos modelos [1], [4]–[15]. Já sistemas de reconhecimento semi-supervisionados e baseados em autoaprendizado necessitam de um grande volume de dados de áudio para uma etapa de pré-treinamento. A maior parte dos dados pode ser não anotada [16]–[19]. Sistemas de reconhecimento não-supervisionados são aqueles em que nenhum dado é anotado. Ou seja, a transcrição dos aúdios não é requerida. Os sistemas de RAF também podem ser categorizados como híbridos e sistemas de ponta a ponta. Em sistemas híbridos de reconhecimento de fala, o processamento nas etapas de modelagem acústica, modelagem de pronunciação e modelagem de língua, podem ser feitos por técnicas distintas e de forma mais desacoplada [20]–[22]. Em contraposição aos sistemas híbridos, têm-se os sistemas de ponta-a-ponta, que propõem a criação de sistemas de RAF utilizando apenas um modelo para realizar todas as etapas de execução [1], [4]–[7], [9]–[13], [15].

O recente avanço das técnicas de aprendizado semi-supervisionadas tem contribuído para a melhoria da qualidade dos sistemas de reconhecimento de fala em português brasileiro. Os experimentos realizados em [2] mostraram que é possível obter resultados expressivos mesmo com apenas 10h de áudio, cenário em que os autores obtiveram desempenho competitivo com aquele alcançado por modelos treinados com um conjunto muito maior de dados (960h de áudio).

Trabalhos recentes exploraram a heurística *Beam Search* [23] em conjunto com modelos de língua [2], [8], [14], [24], [25] e mostraram que a aplicação dessas técnicas reduz a taxa de erro dos modelos de RAF para o português. Entretanto,

trabalhos envolvendo RAF e o português ainda são incipientes e diversas questões de pesquisa se encontram em aberto. Por exemplo, qual é o impacto do conjunto de dados usado para o treinamento dos modelos de língua de um RAF? Como o modelo de língua afeta o tempo de processamento final do sistema de reconhecimento de fala? Como medir a qualidade dos dados de treinamento? Modelos de RAF estado-da-arte para o inglês, como o *Wav2Vec2.0*, alcançam a mesma performance para o português?

Portanto, este trabalho investiga os impactos da qualidade (e tamanho) dos conjuntos de dados em português para o treinamento dos modelos de língua a partir de técnicas estado-da-arte. Também investiga-se aqui o ajuste dos parâmetros da heurística *Beam Search* no processo de decodificação da transcrição. Ainda, propomos uma abordagem para permitir uma análise comparativa entre os conjuntos de dados a partir de uma vetorização *Term Frequency–Inverse Document Frequency* (*TF-IDF*) e métricas de distância. Conforme ilustrado na Figura 1, a proposta detalha um protocolo de avaliação das bases de dados e das técnicas de transcrição e modelagem de língua, visando reprodutibilidade.

Os resultados experimentais mostram que conjuntos de dados de treino com exemplos sintaticamente diferentes podem não ter tanto impacto na qualidade final do sistema de reconhecimento de fala. Além disso, conjuntos de dados com vocabulários maiores tendem a apresentar menor métrica *WER*, impactando mais o resultado final do que modelos de língua grandes (em termos de memória). A abordagem apresentada aqui possibilitou reduzir o tamanho do modelo de língua em 80% e ainda alcançar taxas de erro em torno de 7,17% para a base *Common Voice*, avançando o estado-da-arte com *Wav2Vec2.0*.

II. O Problema de Reconhecimento de Fala

A tarefa de reconhecimento automático de fala pode ser resumida como a transformação de uma representação sonora digitalizada para texto. Ou seja, sistemas e algoritmos que possibilitem extrair o que é falado em um sinal sonoro em sua forma textual. Abordagens supervisionadas são uma opção bem explorada para esta tarefa. As abordagens supervisionadas utilizam conjunto de dados (D) contendo N pares de fala (X_i) e transcrição (Y_i), ou seja,

$$\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^{N}. \tag{1}$$

Usualmente, cada fala X_i é representada por uma sequência de características (ver Equação 2), em que cada trecho de áudio x_t é mapeado em um vetor contínuo m-dimensional. Um exemplo popular na literatura é o uso dos coeficientes *Mel-frequency cepstral* (*MFCCs*) [26] para representação do sinal do sonoro.

$$X_i = [x_1, \cdots, x_T] \in (\mathbb{R}^m)^* \tag{2}$$

As transcrições também são definidas como na Equação 3, em que B é o vocabulário extraído durante o processo de treinamento do modelo de transcrição e cada elemento da sequência (X_i) corresponde a uma palavra, caso sejam modelos baseados em palavras, ou caracteres. S corresponde a quantidade de elementos, sejam eles palavras ou caracteres.

$$Y_i = [y_1, \cdots, y_S] \in B^* \tag{3}$$

Dessa forma, modelos de aprendizagem supervisionada podem utilizar conjuntos de dados para ajustar seus parâmetros e executar a tarefa de transcrição, tendo como objetivo construir um modelo probabilístico ($p(Y \mid X)$), em que dado um conjunto de falas, objetiva-se predizer suas transcrições como em [1], [4]–[15].

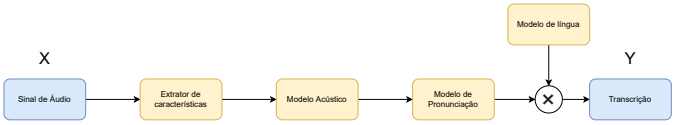


Fig. 2. Um sistema de reconhecimento de fala dividido em sub-problemas (abordagem híbrida). Fonte: Autor.

O problema de RAF, conforme ilustrado na Figura 2, pode ser dividido em três sub-problemas: (i) Modelagem Acústica (MA), em que o objetivo é mapear o sinal sonoro X em fonemas P, representado pela equação 4, (ii) Modelagem de Pronunciação (MP), que por sua vez mapeia os fonemas em representações textuais, representado na equação 5, e por fim (iii) Modelagem de Língua (ML), em que o objetivo é modelar a distribuição entre as palavras de um determinado idioma ou uma mistura delas, associando a probabilidade de uma palavra y_i dado o seu contexto anterior $y_1, \cdots, y_{i-1}$, representado pela equação 6.

$$p_{MA}(X \mid P) \tag{4}$$

$$f_{MP} : \mathcal{P} \mapsto \mathcal{Y} \tag{5}$$

$$p_{ML}(Y) = p(y_i \mid y_1, \cdots, y_{i-1}) \tag{6}$$

Conectando estes três sub-problemas, é possível representar o problema de RAF pela Equação 7 e visualmente pelo diagrama da Figura 2.

$$p_{Hb}(X, Y) = p_{MA}(X \mid f_{MP}(Y)) p_{ML}(Y) \tag{7}$$

Alguns trabalhos desenham soluções para o problema de RAF subdividindo entre sub-problemas conforme indicado na Figura 2, essa abordagem é chamada abordagem híbrida [20]–[22].

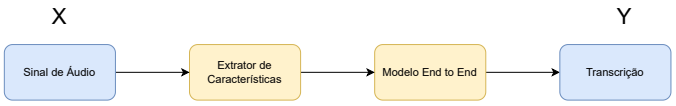


Fig. 3. Esquema de abordagens de ponta a ponta para um sistema RAF. Fonte: Autor.

Além disso, também existem abordagens que solucionam o problema de RAF como um todo. Estas abordagens são chamadas abordagens de ponta-a-ponta (do inglês, *end-to-end*), em que a objetiva-se ter um módulo realizando toda a tarefa de RAF. Geralmente, abordagens ponta-a-ponta são baseadas em aprendizado profundo [1], [4]–[7], [9]–[13], [15]. A Figura 3 ilustra uma abordagem de ponta-a-ponta.

III. Trabalhos Relacionados

Um dos elementos mais importantes na construção de ferramentas de RAF são os conjuntos de dados utilizados para o treinamento dos modelos de aprendizado de máquina.

Utilizando o conjunto de dados de domínio público *LibriSpeech* [3], que contém áudios obtidos de audiolivros em inglês (cerca de 960 horas de áudio para treinamento e cerca de 11h para validação e teste), trabalhos recentes da literatura utilizaram modelos baseados em aprendizado profundo (ponta a ponta) [1], [2], [12], [13], [15] e alcançaram um *WER* de 2.9% a 1.8% para a partição *clean* e de 8.79% a 3.3% para a partição *other*.

No cenário do conjunto de dados em português, em [27], comparou-se trabalhos publicados nos últimos 6 anos e reportou-se resultados em 24 conjuntos de dados com gravações de aúdio em português. Desses conjuntos de dados, 9 possuiam dados suficientes para reconhecimento de fala contínuo. Apenas os trabalhos *Spoltech* [28], *LapsMail* [29], *CoralBR* [30] e *GoogleVoice* [31] continham dados em português brasileiro, sendo apenas o *LapsMail* e o *CoralBR* conjuntos de dados públicos.

O trabalho apresentado em [11] utilizou os conjuntos de dados *LapsStory* [32], contendo aproximadamente 5 horas de transcrições de leituras de livros; *LapsBenchmark* contendo 54 minutos de transcrições de áudios gravados em equipamentos de baixo custo; um conjunto de dados contendo a Constituição [33] e outro contendo o Código de Defesa do Consumidor [33], uma iniciativa contendo as leituras desses documentos para atender as necessidades de pessoas com deficiência, contendo cerca de 9 horas e 1,5 hora respectivamente; *Spoltech* [28], um conjunto de dados privado com 477 falantes e cerca de 5,5 horas de áudios transcritos; *West Point Brazilian Portuguese Speech* constituído de cerca de 5,5 horas de áudios transcritos de 128 falantes, coletados em 1999 em Brasília; CETUC [34], um corpus criado pelo Centro de Estudos e Telecomunicações, totalizando, aproximadamente, 145 horas de áudios transcritos. Nesse trabalho, o conjunto de dados *LapsBenchmark* foi reservado para a avaliação dos modelos e, no melhor cenário, os resultados alcançados foram 4.75% de WER, utilizando um modelo baseado em *Hidden Markov Models* e *Gaussian Mixture Models*.

Em [14], os conjuntos de dados *Sid* [14] foram utilizados, contendo 72 falantes e cerca de 7 horas de áudios transcrito; *VoxForge*, um projeto focado em distribuir dados para transcrição de maneira livre, contendo uma seção em português brasileiro com 111 falantes e aproximadamente 5 horas de áudio transcrito; além dos conjuntos *Spoltech*, *LapsBenchmark* e CETUC. A avaliação dos modelos foi feita com seleção aleatória de 200 falas de 20 falantes do conjunto de dados CETUC. Importante frisar que o conjunto de dados selecionado para teste ficou de fora do conjunto usado para treinamento. A partir de uma solução baseada no *DeepSpeech 2*, um modelo baseado em redes neurais profundas, o trabalho em [14] mostrou um resultado de 25,45% de taxa de erro por palavra. Ressaltamos que esse trabalho utilizou o *LapsBenchmark* apenas como conjunto de validação e seleção dos híper-parâmetros do modelo.

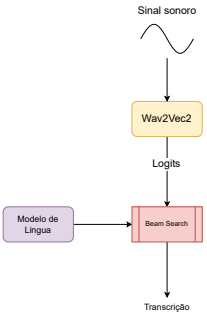


Fig. 4. Fluxograma do processo de reconhecimento de fala baseado no Wav2Vec2 e na heurística Beam Serarch. Fonte: Autor.

Recentemente, [24], [25] avaliaram a performance do *Wav2Vec2* por meio de um *fine-tuning* a partir do modelo pré-treinado disponibilizado em [35]. O modelo disponibilizado em [35] foi treinado em 53 línguas incluindo o português. Os trabalhos em [24], [25] utilizaram combinações de conjuntos de dados em português, alcançando taxas de erros por palavra na média de 10,5% a 22,13%, variando os conjuntos de dados de teste.

Diferente dos trabalhos citados, o objetivo deste trabalho é investigar o impacto dos dados para treinamento, quantitativamente e qualitativamente, na etapa de decodificação do processo reconhecimento de fala. Para tal, foi explorado o modelo estado-da-arte *Wav2Vec2* e conjuntos de dados mais populares para a língua portuguesa.

IV. Materias e Métodos

Este trabalho é um estudo centrado em dados, em que impacto de cinco conjuntos de dados distintos são avaliados para o ajuste fino de um modelos de reconhecimento de fala (*Wav2Vec2* [2]) e também um modelos de língua no processo de decodificação (*Beam Search* [23]).

Conforme ilustrado na Figura 4, um modelo de reconhecimento de fala estado-da-arte, aqui o *Wav2Vec2* [2], é aplicado a um sinal sonoro s, representado como uma sequência de valores s_t, em que s_t representa o valor da amplitude do sinal em um determinado momento. O *Wav2Vec2* tem como saída uma matriz $M_{T\times V}$ que apresenta a probabilidade de um caractere de um vocabulário V aparecer no instante T do sinal original. Utilizando o algoritmo de busca *beam search* é possível decodificar o texto predito pelo modelo. O papel do modelo de língua é ranquear palavras candidatas que são mais coerentes de acordo com o contexto determinado pelas palavras já decodificadas. É explorado aqui o ajuste de parâmetros do *KenLM* [36].

A. Conjunto de Dados

Nesse trabalho foi utilizada uma variedade de conjuntos de dados a fim de investigar o impacto dos dados de treinamento, tanto em quantidade (volume de texto) quanto no domínio de origem do conjunto de dados. Todos os conjuntos utilizados estão disponíveis de forma pública para pesquisa. Foram utilizados os seguintes conjuntos:

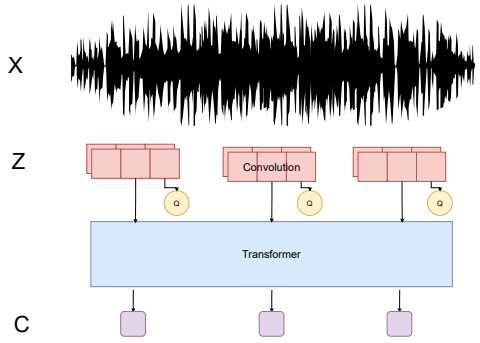


Fig. 5. Ilustração do *framework* do *Wav2Vec 2.0* que aprende representações de fala contextualizadas. Figura adaptada de [2].

- *CommonVoice* [37] [38], um conjunto de dados multilingual para reconhecimento de fala, criado a partir de uma ferramenta de coleta voluntária online, contendo aproximadamente 2508 horas, sendo 30 horas em português (neste trabalho foi utilizada a versão 6.1 e a versão 8.0 do conjunto de dados).
- *Multilingual LibriSpeech* (MLS) [39] [40], a versão multilingual do conjunto *LibriSpeech*, criado a partir de audiolivros, contendo, em sua seção em língua portuguesa, aproximadamente 168 horas de áudio divididos entre 62 falantes.
- CETENFolha [41] [42] um corpus textual extraído de textos da Folha de São Paulo, contendo cerca de 24 milhões de palavras.
- CORAA [43] [25], um conjunto de dados para reconhecimento de fala criado a partir da união de 5 conjuntos de dados: ALIP, que reúne características da fala do interior do estado de São Paulo; C-ORAL Brasil I, um projeto que explora a varidade linguística de Minas Gerais; NURC-Recife que compila exemplos de falas das capitais: Recife, Salvador, Rio de Janeiro, São Paulo e Porto Alegre; SP2010 reúne falas da cidade de São Paulo; *TEDx Portuguese* que contém apresentações realizadas em eventos TEDx. Ao todo o CORAA reune fala espontânea e fala preparada, totalizando 290 horas de áudio divididos entre os subconjuntos de treino, validação e teste.
- WikiText PT-BR [44] [14], um *dump* da Wikipedia de 2018 em português contendo cerca de 8,5 milhões de sentenças.

B. Modelos

1) Wav2Vec2.0: A arquitetura proposta em [2] apresenta um modelo composto por um codificador baseado em múltiplas camadas convolucionais, responsável por mapear o sinal unidimensional de forma de onda $\mathcal{X}$ em representações de fala multidimensionais $\mathcal{Z}$. Essas representações $\mathcal{Z}$ alimentam o módulo de *Transformer* com o objetivo de gerar representações finais ($\mathcal{C}$) capturando informações da sequência de entrada como um todo. Além disso, a arquitetura também possui um módulo de quantização, discretizando as representações de fala $\mathcal{Z}$ em representações quantizadas $\mathcal{Q}$. A organização da arquitetura está esquematizada na Figura 5.

O *Wav2Vec 2.0* apresenta uma vantagem por ser uma proposta de aprendizado auto-supervisionado [45], fazendo com que seja possível treinar boas representações de fala com dados não-supervisionados. A tarefa de pré-treinamento, também cohecida como tarefa de pretexto, é similar ao pré-treino do *BERT* [46], em que para cada sinal de áudio de entrada, a partir de uma proporção p, são mascarados aleatoriamente algumas representações do codificador, de tal forma que o objetivo do modelo é estimar as representações quantizadas dessa proporção mascarada. Uma vez treinado, o modelo é capaz de gerar boas representações contextuais de falas. Ainda, é possível adicionar uma camada linear ao final da arquitetura e realizar o *fine-tuning* para a tarefa de reconhecimento de fala com um conjunto de dados anotado. O trabalho apresentado em [2] mostrou que é possível atingir *WER* de 2.0% com apenas 10% da base de dados para o treinamento, um valor bem próximo ao estado-da-arte (1.8% de *WER*), que considera treinamento com 100% da partição de treino. O trabalho apresentado em [2] mostra que o pré-treino com dados não-supervisionados, inclusive, pode diminuir a quantidade de dados supervisionados necessários para o ajuste fino do modelo.

C. Wav2Vec2.0 em Português

Alguns trabalhos também avaliaram o modelo *Wav2Vec2* treinado com conjuntos de dados em português, como por exemplo o trabalho apresentado em [35], que gerou um modelo de representação de fala treinado em 53 línguas diferentes e apresentou um resultado de 14,7% de taxa de erro por palavra com uma adaptação do modelo para a tarefa de reconhecimento de fala com o conjunto de dados *Multilingual LibriSpeech*. Um modelo de representação de fala foi utilizado no trabalho proposto em [24], que realizou um *fine-tuning* para tarefa de reconhecimento de fala com um conjunto de dados em português brasileiro. Uma taxa de erro de palavra de 9,2% foi reportada quando avaliado na partição de teste do *Common Voice* 7.0, empregando *Beam Search* e um modelo de língua.

O trabalho em [25], que propõe o conjunto de dados CORAA, também avaliou a performance do *Wav2Vec2*. Com um *fine-tuning* na partição de treino CORAA, reportou-se uma taxa de erro por palavra de 20,08% na partição de teste do *Common Voice* 7.0 e uma taxa de erro por palavra de 24,18% na partição de teste do CORAA, ambos utilizando o algoritmo de decodificação gulosa que seleciona apenas o item com maior confiança da matriz de saída.

D. Modelo de Língua

Modelos de língua são aplicados em diversos problemas de linguagem natural e têm mostrado avanços significativos no contexto de geração de texto, como mostrado nas versões do *GPT 1,2 e 3* [47]–[49], que são modelos de língua baseados em *Transformers* treinados com grandes volumes de dados.

Dada a capacidade dos modelos de língua de gerar textos com coerência, esses modelos podem ser utilizados para auxiliar a etapa de decodificação de sistemas de reconhecimento de fala, frequentemente utilizados como mecanismos de checagem durante o processo de decodificação.

Os principais trabalhos que apresentam modelos de língua da atualidade trazem arquiteturas que exigem uma grande capacidade computacional tanto durante a inferência quanto durante o processo de treinamento, mesmo no caso das versões menores, como por exemplo a versão reduzida do *GPT-3*, chamada de *GPT-3 Small* que apresenta 125 milhões de parâmetros; a versão reduzida do *GPT-2* que apresenta 117 milhões de parâmetros e também a versão reduzida do *BERT* [46], que apresenta 110 milhões de parâmetros. Ambas arquiteturas necessitariam de uma infraestrutura com GPUs tanto no treinamento quanto na inferência para aplicações em tempo real.

Uma alternativa ao cenário dos modelos baseados em *Transformers* é utilizar uma abordagem menos custosa e bem aceita tanto na indústria quanto na academia [8], [11], [18], o *KenLM* [36]. O *KenLM* é um modelo estatístico que no processo de construção do modelo segue as etapas:

1) A partir de um corpus textual é realizada a contagem das janelas de palavras (n-grama).
2) É realizado um ajuste na contagem para os casos em que a quantidade de palavras em uma janela é menor que a ordem do modelo.
3) São realizadas a contabilização das probabilidades de cada palavra ocorrer em cada n-grama.

Ao final do processo é obtido um modelo em que é possível inferir a probabilidade de ocorrência de uma palavra w_n dado uma janela de contexto w_1^{n-1} que pode ser expressa por:

$$p(w_n|w_1^{n-1}) \tag{8}$$

em que n é a ordem do n-grama.

O fato de ser um modelo baseado em algoritmo de busca, seja ele baseado em *hashing* ou árvores Trie, facilita a implantação em uma aplicação real, pois não exige a utilização de *GPUs*. Assim, diminui-se o custo financeiro da operação do sistema. Entretanto, o *KenLM* se limita a ser uma base de busca textual baseada na quantidade n-gramas extraídos dos conjuntos de treino. Ainda, no *KenLM* a complexidade de espaço cresce na medida em que o volume dos conjuntos de treino aumenta, diferentemente dos modelos baseados em redes neurais artificiais.

Além da avaliação dos modelos pela decodificação, por meio de uma heurística gulosa, o presente trabalho também avalia a performance dos modelos utilizando a heurística *Beam Search* em conjunto com o *KenLM* como ferramenta para pontuação dos possíveis caminhos de decodificação.

Dessa forma, aqui utiliza-se uma metodologia híbrida, por meio do uso do modelo de língua após o *Wav2Vec2*.

E. Protocolo de Avaliação e Métrica

Nesse trabalho foram avaliados os modelos baseados na *Wav2Vec2*, treinados em português, descritos na subseção IV-C em diferentes cenários.

Para analisar o impacto dos modelos de língua no processo de decodificação, foram treinados modelos de língua com diferentes combinações: (i) entre as partições de treinos dos conjuntos de dados citados na subseção IV-A e também (ii) variando o tamanho do *beam* na heurística *beam search*. Sendo assim, as análises são realizadas executando os seguintes passos:

1) Combinação das partições de treino dos conjuntos de dados para o modelo de língua.
2) Predição dos modelos de reconhecimento utilizando o Wav2Vec2 nas partições de teste dos conjuntos de dados.
3) Para cada combinação são gerados um modelo de língua.
4) Para cada modelo de língua e cada predição dos modelos são realizadas a decodificação para o texto final.
5) Avaliação quantitativa das decodificações computando a taxa de erro por palavra e a taxa de erro por caractere.
6) Analise qualitativa dos erros das decodificações em comparação com a heurística gulosa.

A métrica utilizada para avaliação é o *Word Error Rate* (*WER*), em português taxa de erro por palavra, por ser a mais popular entre os trabalhos de reconhecimento automático de fala. Neste trabalho foi utilizada a biblioteca *Jiwer* [50] para o cálculo da *WER*. Essa métrica foi desenhada para mitigar a dificuldade de medir a performance dado que o modelo de reconhecimento pode predizer palavras com tamanhos diferentes da palavra correta. Essa métrica deriva da distância de Levenshtein, trabalhando no nível de palavra ao invés do nível de fonema ou caracter. A *WER* é computada alinhando-se a palavra reconhecida com a palavra de referência e, a partir desse alinhamento, a taxa de erro é computada de acordo com a seguinte equação:

$$WER = \frac{S + D + I}{S + D + C} \tag{9}$$

em que S é a quantidade de substituições, D a quantidade de deleções, I a quantidade de inserções, C a quantidade de palavras corretamente transcritas. Apresentando no seu valor de saída um número de 0 a infinito, que pode ser interpretado como a porcentagem de erro, em que 0 representa corretude total das palavras transcritas. A porcentagem de erro apresenta um valor superior a 1 quando o $I > C$.

F. Normalização

Assim como no trabalho apresentado em [25], para limpeza e normalização dos conjuntos de dados textuais, foi realizado:

- Remoção da capitalização dos textos [25].
- Remoção de espaços duplicados [25].
- Padronização das pausas [25].
- Expansão de números [25].
- Normalização do símbolo % para "porcentagem" [25].
- Remoção de pontuação [25].

Além disso, foram acrescentadas as seguintes limpezas:

- Remoção de URL.
- Remoção de HTML.
- Normalização de apóstrofe, ex: "d'ele" -> "dele".
- Normalização de moeda, ex: "R$ 15,50" -> "quinze reais e cinquenta centavos".
- Normalização de horas e minuto no padrão HH[h:]MM, ex: "15:30" -> "quinze horas e trinta minutos".
- Normalização de horas padrão 24h, ex: "14h" -> "catorze horas".

- Normalização de datas, ex: "04/08/1996" -> "quatro do oito de mil novecentos e noventa e seis".
- Normalização de métricas, ex: "$10m^2$" -> "dez metros quadrados".

Todos os outros símbolos não endereçados acima foram descartados dos textos.

G. Comparação dos Conjuntos de Dados

Para analisar o impacto dos dados de treinamento, foram propostas duas abordagens para realizar testes de similaridade entre os conjuntos de treino do modelo de língua e os conjuntos de teste do modelo de transcrição.

Primeiramente, para calcular a similaridade, usamos a distancia de *Levenshtein* [51], um método utilizado para comparação entre duas sequências de caractere. Este método computa a quantidade de substituições e inserções de caracteres que precisariam ser realizados para que as duas sequências de caracteres se tornem semelhantes. Esse método é descrito pelo Algoritmo 1.

Algoritmo 1: Calcula a similaridade entre dois conjuntos de dados utilizando Levenshtein

Input: S_{teste} = Sentenças do teste
Input: S_{treino} = Sentenças do treino
$menoresDistancias \leftarrow []$
for s_a ***in*** S_{teste} **do**
  $distancias \leftarrow []$
  for s_b ***in*** S_{treino} **do**
    $d \leftarrow computaLevenshtein(s_a, s_b)$
    insere d **em** $distancias$
  insere $min(distancias)$ **em** $menoresDistancias$
$distancia \leftarrow media(menoresDistancias)$
return $distancia$

Uma alternativa para calcular a similaridade entre os conjuntos de dados é utilizar a similaridade de cosseno a partir de uma representação vetorial das sentenças. Para este trabalho, propõem-se uma segunda abordagem utilizando o *TF-IDF* [52], visto que é popular no contexto de recuperação de informação. Esse método apresenta uma perspectiva de comparação de sentenças com palavras idênticas, uma vez que a representação do *TF-IDF* desconsidera a ordem e o tamanho das sentenças. Assim, o *TF-IDF* considera somente quais palavras pertencem à sentença, qual a frequência das palavras e o quão relevante são diante dos corpora. Aqui, o cálculo das similaridades é realizado utilizando o Algoritmo 2.

Outros aspectos que podem ser utilizados para comparar os conjuntos de dados estão relacionados aos vocabulários que possibilitam entender melhor se os conjuntos de dados possuem conjuntos de palavras parecidos e também o volume desse conjunto. O cálculo da similaridade dos vocabulários está apresentado na equação abaixo:

$$sim = \frac{|Vocab_{teste} \cap Vocab_{treino}|}{|Vocab_{teste}|} \tag{10}$$

Para a análise do conteúdo dos conjuntos de dados, os Algoritmos 1 e 2 foram utilizados para comparar cada conjunto

Algoritmo 2: Calcula a similaridade entre dois conjuntos de dados utilizando TF-IDF e similaridade de cosseno

Input: S_{teste} = Sentenças do teste
Input: S_{treino} = Sentenças do treino
$V \leftarrow extraiVocabulario(S_{treino} \cup S_{teste})$
$W \leftarrow computaPesosTfIdf(V)$
$R_{treino} \leftarrow vetorizaTfIdf(S_{treino}, W)$
$R_{teste} \leftarrow vetorizaTfIdf(S_{teste}, W)$
$S \leftarrow similaridadeCosseno(R_{teste}, R_{treino})$
$maioresSimilaridades \leftarrow []$
for s_a ***in*** S_{teste} **do**
  $s \leftarrow selecionaMaiorSimilaridade(s_a, S)$
  insere s **em** $maioresSimilaridades$
$similaridade \leftarrow media(maioresSimilaridades)$
return $similaridade$

de teste com os conjuntos de treino do modelo de língua, ou seja cada exemplo do conjunto de dados de teste é comparado com todos os exemplos dos conjuntos de treino com o objetivo de encontrar exemplos similares entre a etapa de treino do modelo de língua e a avaliação do sistema de reconhecimento de fala.

V. Experimentos

A. Configurações dos Experimentos

Os experimentos foram realizados combinando as partições de treino entre os 6 conjuntos de dados, *WikiText* PT-BR, *Common Voice* 6.1, *Common Voice* 8.0, *MLS*, CETENFolha e CORAA. A combinação dos conjuntos (1 a 6) é feita conforme a Equação 11, totalizando 63 combinações e são geradas a partir da partição de treino de cada conjunto de dados conforme especificada por cada autor de cada conjunto de dados. Além disso, os modelos de língua foram gerados variando-se o parâmetro de ordem do modelo em *3-gram*, *4-gram* e *5-gram*. Também foi avaliado a variação do parâmetro *beam width* que altera a quantidade de candidatos a serem analisados durante a execução da busca no algoritmo *Beam Search* na etapa de inferência afim de avaliar o impacto na qualidade final do sistema. Para esse parâmetro (*beam width*) foram avaliados os valores: 10, 50 e 100.

Ambos o parâmetros *n-grams* e *beam width* foram escolhidos baseados em testes realizados em trabalhos anteriores [8], [11], [14], acrescentando a variação 4-gram e as variações 50 e 100 para o *beam width*. Para a execução dos experimentos, foi utilizado uma GPU GeForce RTX 3090, com Processador 24-núcleos AMD Ryzen Threadripper 3960X 2.2 GHz e 128 GB of DDR4 RAM.

$$\sum_{k=1}^{6} \frac{6!}{k!(6-k)!} = 63 \tag{11}$$

B. Ajuste dos Parâmetros

A Tabela I detalha o resultado da combinação de cada conjunto de dados de treino, as variações do parâmetro de

TABELA I
TAXA DE ERRO DETALHADA POR ORDEM DO MODELO DE LÍNGUA E *beam width* NAS BASES DE TESTE DO *Common Voice* 6.1 E CORAA

Avaliação	CETENFolha	CORAA	CV 6.1	CV 8.0	MLS	WikiText PT-BR	n-gram	Beam Width	WER
	X	X	X	X		X		100	0,07166955
	X	X	X	X		X	3	50	0,07294531
	X	X	X	X				10	0,08116690
Common Voice 6.1	X		X	X	X	X		100	0,07175460
	X		X	X	X	X	4	50	0,07286026
	X	X	X	X				10	0,08068494
	X		X	X	X	X		100	0,07144275
	X		X	X	X	X	5	50	0,07257676
	X	X	X	X				10	0,08076999
		X	X					100	0,37142562
		X	X				3	50	0,37408322
		X						10	0,39331562
CORAA		X	X	X				100	0,37154770
		X	X				4	50	0,37439312
		X	X					10	0,39409505
		X	X	X				100	0,37188577
		X					5	50	0,37484388
		X	X					10	0,39482754

ordem do modelo de língua e o *beam width* para as partições de teste dos conjuntos de dados *Common Voice* 6.1 e CORAA. É possível observar que para todas as avaliações o maior valor de *beam width* resulta nas menores taxas de erro, entretanto esse parâmetro está diretamente relacionado a complexidade de tempo do algoritmo que no pior caso é denotada por $O(B * m)$ [23], em que B representa o *beam width* e m a profundidade máxima de qualquer caminho na árvore de busca. Sendo assim é necessário analisar a magnitude de melhoria dos resultados no momento em que for necessário implementar esse sistema em aplicações reais.

C. Impacto da Qualidade dos Dados

As Tabelas II e III apresentam a melhoria quantitativa quando comparamos o modelo utilizando-se a heurística gulosa e por meio da equação abaixo:

$$\frac{WER_{greedy} - WER_{beamsearch}}{WER_{greedy}} \tag{12}$$

Analisando a similaridade entre os conjuntos de dados de treino e a partição de teste, a Tabela IV mostra que o *MLS* foi o conjunto de dados que apresentou o maior valor de distância e também foi o único que apresentou uma distância mínima de 5, ou seja, não existe nenhuma sentença da partição de treino do *MLS* que seja exatamente igual a qualquer sentença na partição de teste do *Common Voice* 6.1.

Observando os resultados das similaridades utilizando o método de similaridade de cosseno é possível inferir que o conjunto mais similar a partição de teste foi o CORAA, seguido do CETENFolha e *WikiText* PT-BR. O *MLS* foi o conjunto menos similar e o único que não possui uma sentença exatamente idêntica a partição de teste.

A Tabela VI apresenta os tamanhos dos vocabulários e a similaridade entre a partição de treino dos conjuntos de dados com a partição de teste do Common Voice 6.1.

VI. DISCUSSÃO DOS RESULTADOS

Pela análise das distâncias de Levenshtein entre os conjuntos de treino e teste, é possível notar que o MLS apresenta o maior valor de distância. Com isso, levanta-se a hipótese de que conjuntos de dados de treino com exemplos sintaticamente diferentes do conjunto de teste podem não ter tanto impacto na qualidade final do sistema de RAF. O mesmo padrão encontrado com a distância de Levenshtein é identificado com o uso da similaridade de cosseno, em que o *MLS* também apresenta a menor similaridade (não possui nenhuma sentença exatamente idêntica entre partição de treino e teste). Entretanto, a similaridade de cosseno indica que o conjunto mais similar é o CORAA, seguido do CETENFolha e *WikiText* PT-BR. Diferentemente da distância de cosseno, a distância de *Levenshtein* indica como mais similar o *Common Voice* 8.0, seguido do CORAA e *Common Voice* 6.1.

Comparando os vocabulários dos conjuntos de dados, é possível notar que o *WikiText* PT-BR possui quase todas as palavras contidas no conjunto de teste (cerca de 99% do vocabulário do conjunto de teste). Isto pode se dar a quantidade de *tokens* únicos, cerca de 1,1 milhão palavras - o maior volume entre as bases avaliadas. Analisando a qualidade final do sistema, *WikiText* PT-BR é responsável pela melhoria mais significativa, 31,15%.

Diferentemente da comparação sintática, quando analisamos o aspecto do vocabulário, o conjunto que apresenta a menor similaridade de vocabulário é a partição de teste do próprio *Common Voice* 6.1. Com apenas 55% do vocabulário similar da partição de teste, ainda apresenta uma melhoria na qualidade

TABELA II
MELHORIA DOS MODELOS DE LÍNGUA EM RELAÇÃO A HEURÍSTICA GULOSA NA PARTIÇÃO DE TESTE DO *Common Voice* 6.1

	Melhor melhoria	Pior melhoria	Melhoria sozinho
CETENFolha	33,25%	27,42%	27,95%
CORAA	33,06%	20,61%	20,98%
Common Voice 6.1	33,06%	13,38%	13,48%
Common Voice 8.0	33,06%	19,76%	19,87%
MLS	33,25%	7,45%	7,55%
WikiText PT-BR	33,25%	31,15%	31,39%

TABELA III
MELHORIA DOS MODELOS DE LÍNGUA EM RELAÇÃO A HEURÍSTICA GULOSA NA PARTIÇÃO DE TESTE DO CORAA

	Melhor melhoria	Pior melhoria	Melhoria sozinho
CETENFolha	17,11%	13,50%	14,88%
CORAA	17,86%	15,04%	17,83%
Common Voice 6.1	17,83%	6,58%	6,67%
Common Voice 8.0	17,86%	8,74%	8,84%
MLS	17,10%	9,92%	9,98%
WikiText PT-BR	15,33%	12,92%	13,02%

TABELA IV
DISTÂNCIA DE *Levenshtein* ENTRE PARTIÇÕES DE TREINO E PARTIÇÃO DE TESTE DO *Common Voice* 6.1

	Distância média	Desvio padrão	Distância mínima
CETENFolha	58,867	45,432	0
CORAA	20,815	18,989	0
Common Voice 6.1	58,867	11,537	0
Common Voice 8.0	19,479	12,508	0
MLS	135,807	40,203	5
WikiText PT-BR	94,490	66,084	0

TABELA V
SIMILARIDADE USANDO DISTÂNCIA DE COSSENO E *TF-IDF* ENTRE PARTIÇÕES DE TREINO E TESTE DO *Common Voice* 6.1

	Similaridade média	Desvio padrão	Similaridade máxima
CETENFolha	0,54594	0,14227	1,00000
CORAA	0,55598	0,14159	1,00000
Common Voice 6.1	0,40617	0,14597	1,00000
Common Voice 8.0	0,45120	0,15841	1,00000
MLS	0,24107	0,07221	0,79544
WikiText PT-BR	0,51717	0,11914	1,00000

TABELA VI
SIMILARIDADE ENTRE VOCABULÁRIOS DAS PARTIÇÕES DE TREINO COM AS PARTIÇÃO DE TESTE

	Tamanho do Vocabulário	Similaridade do Vocabulário
CETENFolha	208065	96,08%
CORAA	56553	85,83%
Common Voice 6.1	8210	55,14%
Common Voice 8.0	16300	69,14%
MLS	73346	77,55%
WikiText PT-BR	1163363	98,55%

TABELA VII
SIMILARIDADE USANDO DISTÂNCIA DE COSSENO E *TF-IDF* ENTRE PARTIÇÕES DE TREINO E PARTIÇÃO DE TESTE DO CORAA

	Similaridade média	Desvio padrão	Similaridade máxima
CETENFolha	0,60362	0,17421	1,00000
CORAA	0,67538	0,18607	1,00000
Common Voice 6.1	0,37847	0,14267	1,00000
Common Voice 8.0	0,43383	0,15860	1,00000
MLS	0,26129	0,08296	0,77867
WikiText PT-BR	0,55512	0,14073	1,00000

final do sistema (13,48%) superior à melhoria do conjunto *MLS* (7,55%), que apresenta uma similaridade de vocabulário de cerca de 77%. O resultado sugere que vocabulários ligeiramente similares não sejam suficientes para melhorar o resultado final. Entretanto, em cenário em que o vocabulário é extremamente similar, como por exemplo com o *WikiText* PT-BR ($\sim 99\%$) e CETENFolha ($\sim 96\%$), é possível notar que sozinhos, são os conjuntos de dados que apresentam a menor taxa de erro por palavra no conjunto de teste CV6.1.

Analisando a literatura, o trabalho apresentado em [25] reporta uma *WER* de 24,18% para a partição de teste do CORAA; O trabalho apresentado em [24] reporta um resultado de 8,6% para a partição de teste em português do *Common Voice*. Entretanto, como o presente trabalho utiliza combinações de bases para treino dos modelos, não é possível fazer uma comparação direta dos resultados com outros da literatura. Além disso, este trabalho se difere dos trabalhos da literatura quando leva em conta não somente a qualidade final do sistema (em termos de WER), mas o impacto dos dados usados para o treinamento. Os achados aqui discutidos podem ajudar a otimizar a implantação de sistemas de RAF na indústria.

A análise da qualidade dos modelos no conjunto de teste do CORAA mostra que o uso exclusivo da partição de treino para treinamento do modelo de língua provoca uma melhoria de 17,83% em relação ao emprego do modelo junto com a heurística gulosa. Um outro candidato que apresenta um resultado parecido é o CETENFolha, que sozinho, permite uma melhoria de 14,88%. Diferentemente da avaliação no *Common Voice* 6.1, o *WikiText* PT-BR não apresenta a maior melhoria quando utilizado sozinho.

É possível observar que mesmo utilizando um modelo de língua que se baseia apenas em janelas de n-gramas, como o caso do *KenLM*, não sendo capaz de generalizar sentenças mais complexas e longas, consegue-se diminuir a taxa final de erro de um sistema de reconhecimento de fala. Além disso, a qualidade final do sistema de reconhecimento de fala não pode ser dissociada do custo financeiro operacional. Por exemplo, modelos que necessitam de GPUs modernas durante a fase de inferência podem ser financeiramente inviáveis.

VII. Conclusão

Este trabalho conduz uma investigação centrada em dados visando entender o impacto da qualidade (e quantidade) dos dados em um sistema de reconhecimento de fala para o português brasileiro. A investigação é feita com o uso de um modelo estado da arte para a língua inglesa (*Wav2Vec2.0*) e a heurística *Beam Search* no processo de decodificação da transcrição. Esse trabalho propõe uma abordagem para permitir uma análise comparativa e um melhor entendimento sobre os dados.

Um dos principais aspectos durante a implementação de um sistema de RAF utilizando um modelo de língua é o custo computacional. De forma geral, alinhando esse aspecto à qualidade final do sistema, é possível fazer uma escolha de quais dados priorizar durante o treinamento, visando aumentar performance e reduzir custo computacional. A abordagem proposta aqui auxilia neste processo. Para a partição de teste do CV, por exemplo, a melhor combinação de dados apresenta uma taxa de erro de 18,91%. Entretanto o tamanho final do modelo de língua chega a 17,31 GB. Ainda para a partição de teste do CV, o terceiro melhor caso apresenta uma alternativa melhor em custo computacional, totalizando 9,19 GB e apresentando uma taxa de erro de 18,93%, ou seja uma diferença de 0,02% na taxa de erro e uma redução de 47% no tamanho do modelo final. Cenários similares também estão presentes na avaliação utilizando a partição de teste do CORAA, em que existem modelos que chegam a 18 GB.

Não foram investigados modelos de línguas de maior capacidade, em especial os baseados em *Transformers*, devido ao custo computacional. Contudo, é um desdobramento natural de trabalho futuro investigar modelos como *GPT-2* e *BERT* com a metodologia proposta.

É possível notar que conjuntos de dados com maiores vocabulários tendem a apresentar uma melhor qualidade na avaliação final, reduzindo a métrica *WER*. Todavia, os maiores modelos de língua, em consumo de memória, não necessariamente apresentam os melhores resultados, para sistemas utilizando modelos semelhantes ao *KenLM*. Entender melhor os dados é de suma importância para otimizar o uso de modelos de língua em sistemas de RAF.

Agradecimentos

The authors would also like to thank the CAPES, FAPEMIG - grants Universal APQ-01518-21 and Universal APQ-02176-21, CNPq - grants Universal 406411/2021-2 and 308400/2022-4 and UFOP/PROPPI for supporting the development of the present study.

Referências

[1] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[2] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[3] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[4] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 369–376. [Online]. Available: https://doi.org/10.1145/1143844.1143891

[5] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1764–1772. [Online]. Available: http://proceedings.mlr.press/v32/graves14.html

[6] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. H. Engel, L. Fan, C. Fougner, T. Han, A. Y. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Y. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu, "Deep speech 2: End-to-end speech recognition in english and mandarin," *CoRR*, vol. abs/1512.02595, 2015. [Online]. Available: http://arxiv.org/abs/1512.02595

[7] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[8] I. Macedo Quintanilha, S. Netto, and L. Biscainho, "Towards an end-to-end speech recognizer for portuguese using deep neural networks," 09 2017.

[9] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. C. Sim, T. Bagby, S. Chang, K. Rao, and A. Gruenstein, "Streaming end-to-end speech recognition for mobile devices," *CoRR*, vol. abs/1811.06621, 2018. [Online]. Available: http://arxiv.org/abs/1811.06621

[10] X. Yang, J. Li, and X. Zhou, "A novel pyramidal-fsmn architecture with lattice-free MMI for speech recognition," *CoRR*, vol. abs/1810.11352, 2018. [Online]. Available: http://arxiv.org/abs/1810.11352

[11] C. Batista, A. L. Dias, and N. Sampaio Neto, "Baseline Acoustic Models for Brazilian Portuguese Using Kaldi Tools," in *Proc. IberSPEECH 2018*, 2018, pp. 77–81. [Online]. Available: http://dx.doi.org/10.21437/IberSPEECH.2018-17

[12] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *Interspeech 2019*, Sep 2019. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2680

[13] J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. Cohen, H. Nguyen, and R. Gadde, "Jasper: An end-to-end convolutional neural acoustic model," 09 2019, pp. 71–75.

[14] I. Macedo Quintanilha, S. Lima Netto, and L. Pereira Biscainho, "An open-source end-to-end asr system for brazilian portuguese using dnns built from newly assembled corpora," *Journal of Communication and Information Systems*, vol. 35, no. 1, pp. 230–242, Sep. 2020. [Online]. Available: https://jcis.sbrt.org.br/jcis/article/view/721

[15] W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu, "Contextnet: Improving convolutional neural networks for automatic speech recognition with global context," 2020. [Online]. Available: https://arxiv.org/abs/2005.03191

[16] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," 2020.

[17] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, "Improved noisy student training for automatic speech recognition," *Interspeech 2020*, Oct 2020. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1470

[18] Q. Xu, A. Baevski, T. Likhomanenko, P. Tomasello, A. Conneau, R. Collobert, G. Synnaeve, and M. Auli, "Self-training and pre-training are complementary for speech recognition," *CoRR*, vol. abs/2010.11430, 2020. [Online]. Available: https://arxiv.org/abs/2010.11430

[19] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *CoRR*, vol. abs/2006.13979, 2020. [Online]. Available: https://arxiv.org/abs/2006.13979

[20] H. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, 01 1994.

[21] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4277–4280.

[22] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[23] B. T. Lowerre, "The harpy speech recognition system," Ph.D. dissertation, Carnegie-Mellon University, 1976.

[24] L. R. S. Gris, E. Casanova, F. S. de Oliveira, A. da Silva Soares, and A. C. Júnior, "Brazilian portuguese speech recognition using wav2vec 2.0," *CoRR*, vol. abs/2107.11414, 2021. [Online]. Available: https://arxiv.org/abs/2107.11414

[25] A. C. Junior, E. Casanova, A. Soares, F. S. de Oliveira, L. Oliveira, R. C. F. Junior, D. P. P. da Silva, F. G. Fayet, B. B. Carlotto, L. R. S. Gris, and S. M. Aluísio, "CORAA ASR: a large corpus of spontaneous and prepared speech manually validated for speech recognition in brazilian portuguese," *Language Resources and Evaluation*, Nov. 2022. [Online]. Available: https://doi.org/10.1007/s10579-022-09621-4

[26] T. Ganchev, N. Fakotakis, and G. Kokkinakis, "Comparative evaluation of various mfcc implementations on the speaker verification task," in *in Proc. of the SPECOM-2005*, 2005, pp. 191–194.

[27] T. Lima and M. Da Costa-Abreu, "A survey on automatic speech recognition systems for portuguese language and its variations," *Computer Speech & Language*, vol. 62, p. 101055, 12 2019.

[28] M. Schramm, L. Freitas, A. Zanuz, and D. Barone, "Cslu: Spoltech brazilian portuguese version 1.0 ldc2006s16," 2006.

[29] R. Oliveira, P. Batista, N. Neto, and A. Klautau, "Baseline acoustic models for brazilian portuguese using cmu sphinx tools," in *Computational Processing of the Portuguese Language*, H. Caseli, A. Villavicencio, A. Teixeira, and F. Perdigão, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 375–380.

[30] T. Raso and H. Mello, "The c-oral-brasil i: Reference corpus for informal spoken brazilian portuguese," in *Computational Processing of the Portuguese Language*, H. Caseli, A. Villavicencio, A. Teixeira, and F. Perdigão, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 362–367.

[31] J. Gonzalez-Dominguez, I. Lopez-Moreno, P. J. Moreno, and J. Gonzalez-Rodriguez, "Frame-by-frame language identification in short utterances using deep neural networks," *Neural Networks*, vol. 64, pp. 49–58, 2015, special Issue on "Deep Learning of Representations". [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608014002019

[32] N. Neto, C. Patrick, A. Klautau, and I. Trancoso, "Free tools and resources for brazilian portuguese speech recognition," *Journal of the Brazilian Computer Society*, vol. 17, no. 1, pp. 53–68, 2011.

[33] P. Legal, "Pcd legal: Acessível para todos," 2018. [Online]. Available: http://www.pcdlegal.com.br/

[34] PUC-Rio, "Centro de estudos em telecomunicações (cetuc)." [Online]. Available: http://www.cetuc.puc-rio.br/

[35] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," 2020.

[36] K. Heafield, "KenLM: Faster and smaller language model queries," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, Jul. 2011, pp. 187–197. [Online]. Available: https://aclanthology.org/W11-2123

[37] [Online]. Available: https://commonvoice.mozilla.org/pt/datasets

[38] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *CoRR*, vol. abs/1912.06670, 2019. [Online]. Available: http://arxiv.org/abs/1912.06670

[39] [Online]. Available: https://www.openslr.org/94/

[40] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "Mls: A large-scale multilingual dataset for speech research," *Interspeech 2020*, Oct 2020. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-2826

[41] [Online]. Available: https://www.linguateca.pt/cetenfolha/index_info.html

[42] Linguateca, "Cetenfolha." [Online]. Available: https://www.linguateca.pt/cetenfolha/

[43] [Online]. Available: https://github.com/nilc-nlp/CORAA

[44] [Online]. Available: https://igormq.github.io/datasets/

[45] Y. Tian, L. Yu, X. Chen, and S. Ganguli, "Understanding self-supervised learning with dual deep networks," *CoRR*, vol. abs/2010.00578, 2020. [Online]. Available: https://arxiv.org/abs/2010.00578

[46] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[47] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[48] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[49] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

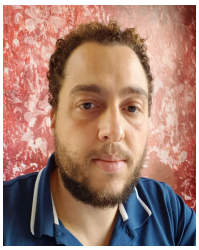
[50] [Online]. Available: https://github.com/jitsi/jiwer

[51] V. I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions and Reversals," *Soviet Physics Doklady*, vol. 10, p. 707, Feb. 1966.

[52] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, pp. 11–21, 1972.

João Alvarenga holds a bachelor's in Computer Science from the Federal University of Ouro Preto (UFOP), in 2019. He is currently Team Lead and Senior Machine Learning Engineer at Stilingue Inteligência Artificial and a master's student in the Graduate Program in Computer Science at UFOP. His research interests include deep learning, natural language processing, and speech recognition.

Luiz Merschmann Luiz H.C. Merschmann is Professor in the Department of Applied Computing at Federal University of Lavras, Brazil. He received the BSc degree in Mining Engineering from Federal University of Ouro Preto, Brazil, MSc degree in Production Engineering from Federal University of Rio de Janeiro, Brazil, and PhD degree in Computer Science from Fluminense Federal University, Brazil. In 2012, he carried out postdoctoral research at University of Kent, UK. He has published several peer reviewed papers in journals and conference proceedings. His research interests include data mining, machine learning, artificial intelligence and natural language processing.



Eduardo Luz holds a bachelor's degree in Electrical Engineering from the Federal University of Minas Gerais (2005), and a Ph.D. in Computer Science from the Federal University of Ouro Preto (2019). He is an Adjunct Professor at the Department of Computing (DECOM) at the Federal University of Ouro Preto and a permanent member of the Graduate Program in Computer Science. His research interests include pattern recognition, machine learning, computer vision, and embedded systems.