UNIVERSIDADE FEDERAL DE OURO PRETO

UMA ABORDAGEM CENTRADA EM DADOS PARA RECONHECIMENTO DE FALA EM PORTUGUÊS: MODELO DE LÍNGUA E SUAS IMPLICAÇÕES

João Paulo Reis Alvarenga

Ouro Preto
Abril de 2023

UNIVERSIDADE FEDERAL DE OURO PRETO

UFOP

decom departamento de computação

UMA ABORDAGEM CENTRADA EM DADOS PARA RECONHECIMENTO DE FALA EM PORTUGUÊS: MODELO DE LÍNGUA E SUAS IMPLICAÇÕES

João Paulo Reis Alvarenga

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação, da Universidade Federal de Ouro Preto, como parte dos requisitos necessários à obtenção do título de Mestre em Ciência da Computação.

Orientador: Eduardo José da Silva Luz

Ouro Preto
Abril de 2023

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

A473a Alvarenga, Joao Paulo Reis.
Uma abordagem centrada em dados para reconhecimento de fala em português: modelo de língua e suas implicações. [manuscrito] / Joao Paulo Reis Alvarenga. - 2023.
66 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Eduardo José da Silva Luz.
Dissertação (Mestrado Acadêmico). Universidade Federal de Ouro Preto. Departamento de Computação. Programa de Pós-Graduação em Ciência da Computação.
Área de Concentração: Ciência da Computação.

1. Reconhecimento automático da voz. 2. Inteligência artificial. 3. Ciência da Computação. I. Luz, Eduardo José da Silva. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004

Bibliotecário(a) Responsável: Elton Ferreira de Mattos - SIAPE: 1.754.007

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS
DEPARTAMENTO DE COMPUTACAO
PROGRAMA DE POS-GRADUACAO EM CIENCIA DA COMPUTACAO

FOLHA DE APROVAÇÃO

João Paulo Reis Alvarenga

Uma abordagem centrada em dados para reconhecimento de fala em português: modelo de língua e suas implicações

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Mestre em Ciência da Computação

Aprovada em 10 de março de 2023

Membros da banca

Prof. Dr. Eduardo José da Silva Luz - Orientador - Universidade Federal de Ouro Preto
Prof. Dr. Luiz Henrique De Campos Merschmann - Universidade Federal de Lavras
Prof. Dr. Rodrigo César Pedrosa Silva - Universidade Federal de Ouro Preto

Prof. Dr. Eduardo José da Silva Luz, orientador do trabalho, aprovou a versão final e autorizou seu depósito no Repositório Institucional da UFOP em 05/04/2023

Documento assinado eletronicamente por **Eduardo Jose da Silva Luz**, **VICE-COORDENADOR(A) DE CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO**, em 12/04/2023, às 14:32, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0507704** e o código CRC **22C88D5F**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: (31)3559-1641 - www.ufop.br

Resumo da Dissertação apresentada à UFOP como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UMA ABORDAGEM CENTRADA EM DADOS PARA RECONHECIMENTO DE FALA EM PORTUGUÊS: MODELO DE LÍNGUA E SUAS IMPLICAÇÕES

João Paulo Reis Alvarenga

Abril/2023

Orientador: Eduardo José da Silva Luz

Programa: Ciência da Computação

Os avanços mais recentes no Reconhecimento Automático de Fala permitem alcançar uma qualidade jamais antes vista em línguas com dados abundantes, tais como o inglês, e em línguas com dados limitados, como o português. Em particular, abordagens baseadas em modelos de *Transformers* permitem realizar a tarefa de reconhecimento de fala diretamente a partir da representação do sinal bruto. Alguns estudos já indicam que a qualidade da transcrição pode ser melhorada ainda mais com o uso de modelos de linguagem. No entanto, o impacto real destes modelos ainda não está claro para o português brasileiro, assim como a importância da qualidade dos dados usados para treinar os modelos. Por isso, este trabalho explora o impacto dos modelos de linguagem aplicados ao reconhecimento de fala para língua portuguesa, tanto em termos de qualidade de dados quanto de desempenho computacional, com uma abordagem centrada em dados. Uma abordagem para medir a similaridade entre conjuntos de dados é proposta para auxiliar na tomada de decisão durante o treinamento. Os resultados mostram que é possível reduzir o tamanho do modelo de linguagem em 80% e ainda alcançar taxas de erro por palavra em torno de 7,17% para o conjunto de dados *Common Voice*.

Abstract of Dissertation presented to UFOP as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

DATA-CENTRIC APPROACH FOR PORTUGUESE SPEECH RECOGNITION: LANGUAGE MODEL AND ITS IMPLICATIONS

João Paulo Reis Alvarenga

April/2023

Advisor: Eduardo José da Silva Luz

Department: Computer Science

Recent advances in Automatic Speech Recognition have enabled remarkable improvements in transcription quality for languages with abundant data, such as English, and for those with limited resources, such as Brazilian Portuguese. Recent approaches address speech recognition problems with *Transformers* based models, which are capable of directly processing raw signals without manual feature extraction. It has been shown that language models can further improve transcription quality. However, the real impact of such language models is still not clear, especially in the Portuguese scenario. Moreover, the quality of the data used for training is known to be critical, yet there are few works in the literature addressing this issue. This work investigates the effect of language models applied to Portuguese speech recognition in terms of data quality and computational performance, with an emphasis on data. We propose an approach to measure the similarity between datasets to inform decision-making during training. Our results show that it is possible to reduce the size of the language model by 80% and still achieve error rates of 7.17% on the Common Voice dataset.

Agradecimentos

Agradeço a *Maria*, minha mãe e *Dayanne*, minha irmã pelo apoio incondicional.
Agradeço a *Julia*, minha namorada, por compartilhar toda essa jornada comigo.
Agradeço aos *meus amigos* por todos os momentos de descontração e pelo companheirismo.
Agradeço aos *amigos* e *colegas* da Stilingue pela parceria e aprendizados de tantos anos.
Agradeço a todos os docentes e servidores da Universidade Federal de Ouro Preto por se esforçarem para manter a melhor infraestrutura possível.
Agradeço ao CSI-Lab por todo suporte para execução dos experimentos.
Agradeço ao *Eduardo*, meu orientador, pela mentoria, confiança, paciência e maestria nessa jornada.

Lista de Figuras

Lista de Tabelas

Nomenclatura

RAF	*Reconhecimento Automático de Fala*
WER	*Word Error Rate*
TF-IDF	*Term Frequency–Inverse Document Frequency*
CER	*Character Error Rate*
CV	*Common Voice*
CETUC	Centro de Estudos em Telecomunicações da PUC-Rio
MFCC	*Mel-Frequency Cepstral Coefficients*
BERT	*Bidirectional Encoder Representations from Transformers*
MLS	*Multilingual LibriSpeech*
MA	Modelo Acústico
MP	Modelo de Pronunciação
ML	Modelo de Língua
PLN	Processamento de Linguagem Natural
GPU	*Graphics Processing Unit*

Sumário

Capítulo 1

Introdução

A tarefa transformação de sinais sonoros em representações textuais, chamada de reconhecimento automático de fala (RAF), tem se desenvolvido nos últimos anos com avanços para a língua inglesa, com trabalhos [4, 17] baixas taxas de erro por palavra, do inglês *Word Error Rate (WER)* , 2.9% a 1.8% de *WER*, quando avaliados no conjunto *LibriSpeech* [38], um conjunto de dados com cerca de 960 horas de áudios em inglês transcrito.

Os sistemas de RAF também podem ser categorizados quanto aos seus componentes como: sistemas 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 [1, 7, 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 [2, 6, 9, 15–17, 19, 20, 30, 39, 54].

As propostas de sistemas de RAF presentes na literatura são implementados utilizando diferentes categorias de aprendizado de máquina, tais como sistemas de reconhecimento de fala supervisionados, auto-aprendizado, não-supervisionados, e com relação aos sistemas podemos ter sistemas híbridos e sistemas de ponta a ponta. Os sistemas supervisionados são aqueles que necessitam de um conjunto de pares de áudio e transcrição para o treinamento dos modelos[2, 6, 9, 15–17, 19, 20, 30, 34, 39, 44, 54]. Já sistemas de reconhecimento 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 [4, 11, 40, 53]. Sistemas de reconhecimento não-supervisionados são aqueles em que nenhum dado é anotado, ou seja, a transcrição dos aúdios não é requerida.

O recente avanço das técnicas de aprendizado auto-aprendizado também tem contribuído para a melhoria da qualidade dos sistemas de reconhecimento de fala em português brasileiro. Os experimentos realizados em [4] 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* [33] em conjunto com modelos de língua [4, 26, 34, 44, 50] e mostraram que a aplicação dessas técnicas reduz a taxa de erro dos modelos de RAF para a língua portuguesa. Entretanto, trabalhos envolvendo RAF e a língua portuguesa ainda são incipientes e diversas questões de pesquisa se encontram em aberto. Para a língua portuguesa, ainda não temos bases com grandes volumes e utilizadas para avaliação em diversos trabalhos como *LibriSpeech*, o que torna o problema de reconhecimento de fala para a língua portuguêsa ainda mais desafiador. 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 (*TF-IDF*) e métricas de distância.
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*.

1.1 Motivação

Os trabalhos que analisam e propões sistemas de reconhecimento de fala frequentemente utilizam modelos de línguas como mecanismo para a diminuição da *WER* final do sistema [6, 10, 26, 31, 39, 50, 54]. Entretanto ainda existe um espaço para investigar a construção desses modelos de língua que são diretamente impactados pelos dados utilizados, principalmente no cenário da língua portuguesa que quando comparada a língua inglesa possui um volume significativamente menor.

1.2 Objetivos

1.2.1 Objetivos Gerais

O principal objetivo desse trabalho é colaborar com o avanço da qualidade dos modelos de reconhecimento automático de fala para o português brasileiro, mais especificamente, explorando uma abordagem centrada em dados.

1.2.2 Objetivos Específicos

O objetivo deste trabalho é investigar o impacto dos dados para treinamento, quantitavivamente e qualitativamente, na etapa de decodificação do processo reconhecimento de fala.
Os objetivos específicos desse trabalho são:

1. Avaliar quantitativamente o impacto da adição e remoção de dados no treinamento de modelos de língua

2. Avaliar qualitativamente o impacto das bases de treinamento no resultado final da transcrição;

3. Analisar o custo computacional de modelos de língua no processo de RAF;

1.3 Publicações

- Este trabalho gerou um artigo aceito na *IEEE Latin America Transactions*, intitulado *A Data-Centric Approach for Portuguese Speech Recognition: Language Model And Its Implications*;

- Participação em evento da comunidade internacional[1];

- Repositório com código fonte e recursos desse trabalho[2].

[1]https://discuss.huggingface.co/t/open-to-the-community-xlsr-wav2vec2-fine-tuning-week-for-low-resource-languages/4467
[2]https://github.com/joaoalvarenga/language-model-evaluation

Capítulo 2

Reconhecimento Automático de Fala: Definições e Abordagens

Neste capítulo, são apresentados os conceitos fundamentais para o entendimento do trabalho bem com a revisão da literatura.

2.1 Reconhecimento Automático de Fala

O Reconhecimento Automático de Fala é uma tarefa com o objetivo de transformar, a partir de um método computacional automático, a representação digital de um sinal sonoro $\mathcal{X}$ na sua representação textual correspondente $\mathcal{Y}$, conforme indicado na Equação 2.1. Geralmente os sistemas de RAF são constituídos de uma grande quantidade de componentes individuais como um modelo acústico para predizer baseado no contexto os sub-fonemas a partir de um áudio, uma estrutura gráfica para mapear os sub-fonemas em fonemas e um modelo de pronúncia para transformar os fonemas em palavras [30].

$$f : \mathcal{X} \mapsto \mathcal{Y} \tag{2.1}$$

2.2 Sistemas de Reconhecimento de Fala

Sistemas de reconhecimento de fala podem ser divididos em duas categorias: sistemas híbridos e sistemas de ponta-a-ponta.

2.2.1 Sistemas Híbridos de Reconhecimento de Fala

Um sistema de reconhecimento de fala pode ser construído a partir de três módulos: o Modelo Acústico (MA), responsável por mapear o sinal sonoro X em fonemas P, representado pela Equação 2.2, o Modelo de Pronunciação (MP, que por sua vez

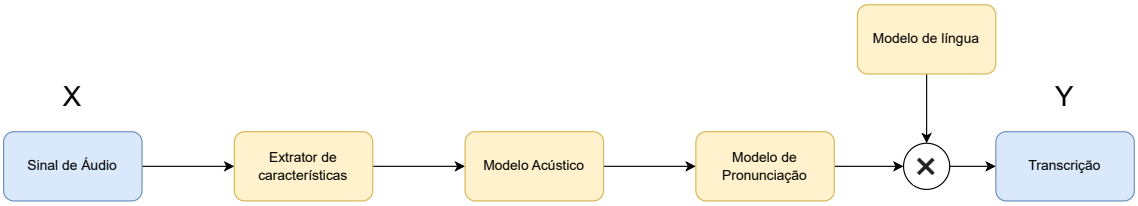

Figura 2.1: Esquema de abordagens híbrida. Fonte: Autor

mapeia os fonemas em representações textuais, representado na Equação 2.3, através de um dicionário de fonemas, e por fim o Modelo de Língua (ML), responsável por modelar a distribuição entre as palavras de um determinado idioma ou uma mistura deles, em alguns trabalhos representado por um modelo de língua casual, em que seu objetivo é associar a probabilidade de uma palavra y_i dado o seu contexto anterior $y_1, \cdots, y_{i-1}$, representado na equação 2.4. A Figura 2.1 exemplifica os componentes de um sistema de RAF híbrido.

$$p_{MA}(X \mid P) \tag{2.2}$$

$$f_{MP} : \mathcal{P} \mapsto \mathcal{Y} \tag{2.3}$$

$$p_{ML}(Y) = p(y_i \mid y_1, \cdots, y_{i-1}) \tag{2.4}$$

Conectando esses três módulos é possível construir um modelo híbrido de reconhecimento de fala, projetando-os separadamente, em alguns casos utilizando técnicas de modelagem diferentes para cada módulo que por fim podem ser utilizados para transcrição conforme indicado nos trabalhos [1, 7, 22]. Sendo assim, um modelo híbrido p_{Hb} pode ser representado por:

$$p_{Hb}(X, Y) = p_{MA}(X \mid f_{MP}(Y))p_{ML}(Y) \tag{2.5}$$

A principal vantagem dos sistemas híbridos é a possibilidade de combinar conjuntos de dados diferentes e técnicas diferentes para o desenvolvimento de cada módulo. A separação dos módulos também possibilita uma manutenção e melhoria mais simplificada, dado que é possível identificar os erros cometidos por cada parte do sistema. Entretanto, quando comparada a qualidade final dos sistemas híbridos com sistemas de ponta-a-ponta, apresentam uma *WER* superior [2, 4, 6, 34].

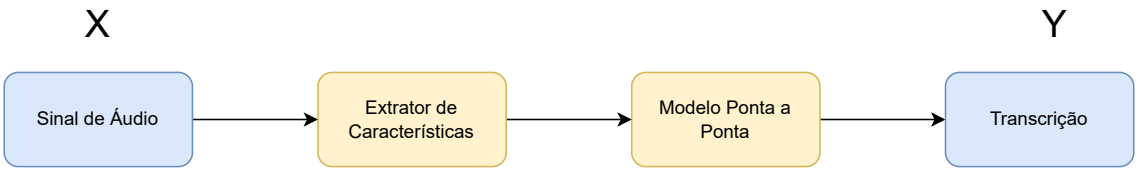

Figura 2.2: Esquema de abordagens de ponta a ponta. Fonte: Autor

2.2.2 Sistemas de Ponta a Ponta para Reconhecimento de Fala

Uma outra forma de construir sistemas de reconhecimento de fala é o projeto de um único modelo responsável por mapear o sinal sonoro de entrada no texto final da transcrição, esses são chamados de modelos ponta-a-ponta (*end-to-end*). Geralmente, esses modelos são construídos utilizando redes neurais, que no processo de aprendizado combinam as características extraídas dos sinais sonoros mapeando em uma sequência de probabilidades para cada *token* do vocabulário de transcrição.
As vantagens desses modelos, principalmente por serem abordagens baseadas em aprendizado profundo, se baseiam na possibilidade de construir modelos que aprendem a extrair e combinar as características necessárias para executar boas transcrições, ou seja não necessitam de uma etapa manual de engenharia de características. Quando comparado com os modelos híbridos, os modelos de ponta a ponta, apresentam uma qualidade superior nos resultados, conforme indicam os trabalhos [2, 6, 9, 15–17, 19, 20, 30, 39, 54]. Entretanto, também por serem baseados em aprendizado profundo e, internamente, unirem vários módulos de reconhecimento de fala, dificultam o entendimento a manutenção dos modelos, além de precisarem de grandes quantidades de dados anotados para atingirem uma boa performance.

2.3 Tipos de Aprendizado de Máquina no Reconhecimento de Fala

Sistemas de RAF podem ser construídos utilizando diferentes técnicas de aprendizado de máquina que utilizam algumas categorias de aprendizado.

2.3.1 Reconhecimento Automático de Fala Supervisionado

A tarefa de reconhecimento automático pode ser resumida como a transformação de uma representação sonora da fala em forma de onda para texto. A abordagem supervisionada consiste em utilizar um conjunto de dados definido na Equação 2.6 contendo N pares de X falas e Y transcrições para treinar um modelo de aprendizado de máquina.

$$\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^{N}, \tag{2.6}$$

Geralmente as falas X são representadas por uma sequência de características, definido na Equação 2.7, em que cada quadro x_t é um vetor contínuo m-dimensional, em uma sequência de T passos, contendo por exemplo os coeficientes *Mel-frequency cepstral* (MFCCs).

$$X = [x_1, \cdots x_T] \in (\mathbb{R}^m)^* \tag{2.7}$$

As transcrições são geralmente definidas como na Equação 2.8, em que B é o vocabulário extraído durante o processo de treinamento e cada elemento da sequência corresponde ou a uma palavra, caso sejam modelos baseados em palavras, ou caracteres e S a quantidade de elementos sejam eles caracteres ou palavras.

$$Y = [y_1, \cdots, y_S] \in B^* \tag{2.8}$$

Dessa forma esses modelos de reconhecimento de fala podem utilizar desses conjuntos de dados para ajustar seus parâmetros e executar a tarefa de transcrição, tendo como objetivo construir um modelo probabilístico, como por exemplo, $p(Y \mid X)$, para dado um conjunto de falas, predizer suas transcrições como mostrado nos trabalhos [2, 6, 9, 15–17, 19, 20, 30, 34, 39, 44, 54].

2.3.2 Reconhecimento Automático de Fala e Autoaprendizado

Os modelos de ponta a ponta de aprendizado supervisionado, muitas vezes necessitam de uma grande quantidade de dados anotados para atingir uma performance próxima ou superior aos estados da arte, entretanto existem alguns trabalhos que propõe o uso do autoaprendizado (*self-learning*) para o pré-treino do modelo. Nesses trabalhos o autoaprendizado se dá pelo pré-treino de um modelo em que o seu objetivo é aprender representações latentes do sinais sonoros, ou seja, mapear os sinais sonoros em representações embutidas em um espaço n-dimensional. Dessa forma, esses modelos podem ser treinados utilizando dados sem anotação manual, de tal forma que, como no trabalho [4], em que o modelo se define por: dado um sinal sonoro X, representando uma sequência de amostras do sinal:

$$X = [x_1, \cdots x_T] \tag{2.9}$$

em que x_i, representa uma amostra do sinal no momento t. Utilizando o redes neurais convolucionais [27], apresentado na Equação 2.10, o sinal X é mapeado em uma

sequência de representações contínuas latente Z, dado que as redes convolucionais geram representações janeladas. Essas representações são mapeadas em representações contextualizadas C utilizando mecanismos de atenção [52], conforme representado na equação 2.11. Por fim, para facilitar o treinamento do modelo é realizada uma quantização Q, através da discretização da representação Z.

$$f_Z : X \mapsto Z \tag{2.10}$$

$$f_C : Z \mapsto C \tag{2.11}$$

$$f_Q : Z \mapsto Q \tag{2.12}$$

O treinamento desse modelo ocorre a partir da representação quantizada Q do sinal sonoro de entrada, para cada segmento q_t da representação, são selecionados K outros segmentos $\hat{q}$ da representação, de forma em que o objetivo do treinamento é que o modelo possa separar entre a representação q_t e os outros segmentos. Esse objetivo pode ser atingido utilizando o erro contrastivo L_c definido por:

$$L_c = -\log \frac{e^{sim(c_t, q_t)/K}}{\sum_{\hat{q} \sim \mathbf{Q}_t} e^{sim(c_t, \hat{q})/K}} \tag{2.13}$$

em que *sim* é a distância de cosseno.

Uma vez tendo esse modelo pré-treinado num conjunto de dados não anotados é possível adicionar uma camada a esse modelo e realizar o ajuste fino dos parâmetros com um conjunto menor de dados anotados conforme mostrado nos trabalhos [4, 11, 40, 53].

Estes trabalhos apresentam uma grande vantagem em relação a cenários que possuem pouco recurso anotado disponível, uma vez em que a partir de um conjunto reduzido de dados anotados é possível obter uma performance bem próxima quando treinado com um conjunto grande de dados anotados. Em contrapartida, conforme indicado nos experimentos dos trabalhos [4, 40, 53], esses modelos exigem uma grande quantidade de dados sem anotação, que embora sejam de fácil acesso, faz com que seja necessário uma grande quantidade de recurso computacional para realizar o treinamento dos modelos.

2.3.3 Reconhecimento Automático de Fala Não-supervisionado

Alguns trabalhos [5, 23] recentes apresentam uma proposta utilizando somente dados não rotulados para a construção de sistemas de RAF, utilizando por exemplo redes generativas adversativas [14] para o treinamento dos modelos e algoritmos de

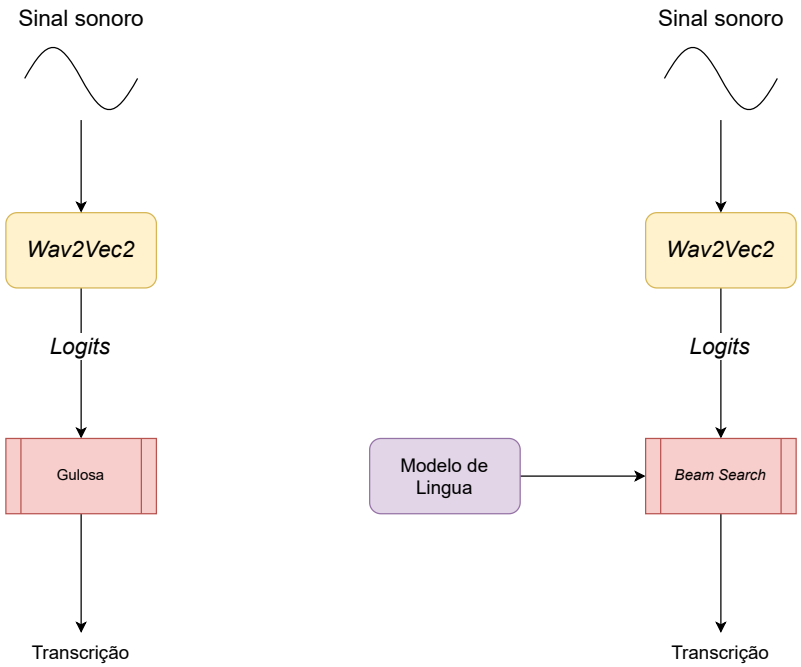

Figura 2.3: Fluxograma do processo de reconhecimento de fala baseado no Wav2Vec2 utilizando para decodificação a heurística gulosa (esquerda) e a heurística *Beam Search* (direita). Fonte: Autor

clusterização, de tal forma em que, nos trabalhos [5, 23] a taxa de erro por palavra foi superior quando comparado aos modelos de auto-aprendizado, entretanto bem próxima.

A principal vantagem desses modelos é que eles não necessitam de nenhum dado anotado, dessa forma podem ser utilizadas qualquer gravação de áudio disponível para a língua em que se deseja realizar a transcrição. Entretanto apresentam a mesma dificuldade de treinamento que os sistemas treinados através de auto-aprendizado, uma grande capacidade computacional para o treinamento dos modelos.

2.4 Algoritmos de decodificação

O processo de transcrição de áudio utilizando um modelo como *Wav2Vec2* ocorre na transformação de um sinal sonoro X através da inferência do modelo em uma matriz $L_{T\times V}$ representando a probabilidade de cada token, podendo ser um caractere e em alguns casos uma palavra, presente no vocabulário V ocorrer no instante de tempo T. Essa matriz pode ser decodificada de várias formas para gerar o texto da transcrição final. Uma das formas mais simples é utilizando a heurística gulosa que seleciona os tokens To do vocabulário com o maior valor de probabilidade em um instante t.

$$f_{w2v} : X \mapsto LTo_t = argmax(L_t) \tag{2.14}$$

2.4.1 Heurística *Beam Search*

Uma heurística bastante popular em sistemas de PLN é a *Beam Search* [33]. O objetivo é encontrar uma sequência mais próxima do resultado ótimo que a busca gulosa. A versão mais direta da heurística apresenta apenas um híper-parâmetro, a largura do feixe w, do inglês *beam width*. Esse parâmetro indica quantos candidatos serão selecionados em cada execução da heurística.
Detalhando um pouco melhor o funcionamento da heurística, no passo inicial $t = 0$, são selecionados os w tokens de maior probabilidade, cada um deles serão candidatos a serem os primeiros tokens no texto final. A cada passo subsequente serão selecionados os w candidatos que maximizam o produto da probabilidade dos tokens de cada sequência candidata.

Exemplificando *Beam Search*

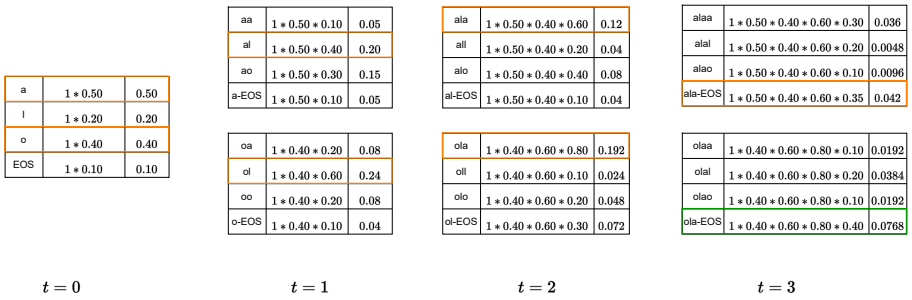

Figura 2.4: Exemplo de execução da heurística *Beam Search* para um tamanho de feixe igual a 2 decodificando a palavra "olá". Fonte: Autor.

Considere o vocabulário $V = \{a, l, o, EOS\}$, em que EOS indique o fim de sentença e uma largura do feixe igual a 2. A partir desse vocabulário é possível gerar uma matriz $L_{T \times V}$ contendo a probabilidade de cada token em cada instante de tempo, conforme a Equação 2.15, em que BOS indica o token de ínicio de sentença. A Figura 2.4 ilustra a execução da *Beam Search* em que cada passo são selecionados os 2 candidatos com a maior probabilidade acumulada até que seja selecionado um candidato contendo um token de fim de sentença. Essa execução foi finalizada em $t = 3$ por ter atingido um candidato contendo um token de fim de sentença, entretanto se ainda existisse um outro candidato com maior probabilidade em outra ramificação, o algoritmo seria executado até atingir o valor máximo de instantes de tempo ou que a ramificação também selecionasse um candidato com o token de fim de sentença.

$$P(To_{t'}|To_1, ..., To_{t'-1}, BOS) \tag{2.15}$$

Melhorando a *Beam Search*

A heurística *Beam Search* no momento de decodificação tem acesso apenas a informações da predição do modelo de transcrição, entretanto existem maneiras de adicionar mais informações para selecionar melhor os candidatos finais. Uma forma de adicionar informações sobre a sintaxe é utilizar um modelo de língua para ajudar na pontuação final de cada candidato. Utilizando o exemplo da subseção 2.4.1, é possível adicionar um modelo de língua para que no passo t', além de considerar apenas $P(To_t|To_1, ..., To_{t-1}, BOS)$, é possível utilizar o texto gerado até então como entrada em um modelo de língua e computar a pontuação final, conforme a equação 2.16, em que α e β são hiper-parâmetros para ajustar o impacto de cada parte na pontuação final de cada candidato.

$$P(final) = \alpha * P(To'_t|To_1, ..., To_{t'-1}, BOS) + \beta * MA(To_{1..t'}) \tag{2.16}$$

2.5 Modelos 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* [8, 45, 46], 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* [12], que apresenta 110 milhões de parâmetros. Todas as 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 [6, 44, 53], o *KenLM* [21]. 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

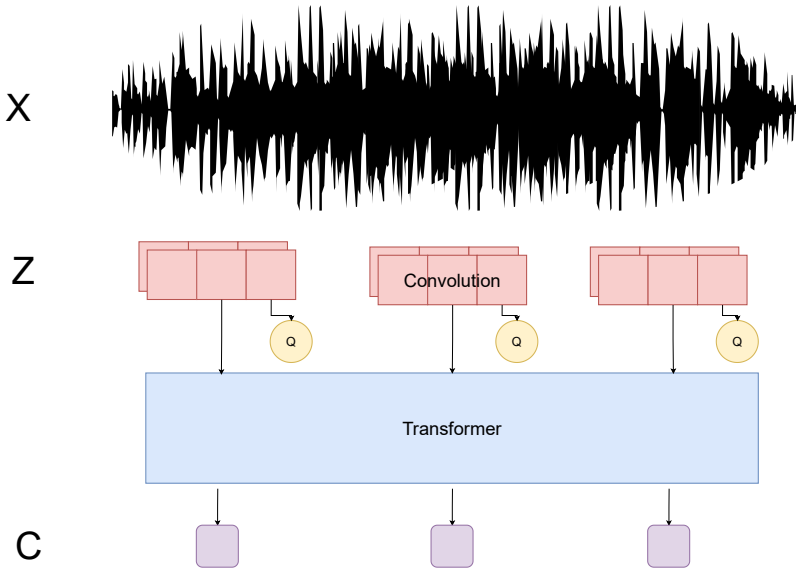

Figura 2.5: Ilustração do *framework* do *Wav2Vec 2.0* que aprende representações de fala contextualizadas. Figura adaptada de [4]

(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:

$$p(w_n|w_1^{n-1}) \tag{2.17}$$

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.

2.6 *Wav2Vec2.0*

A arquitetura proposta em [4] apresenta um modelo composto por um codificador baseado em múltiplas camadas convolucionais, responsável por mapear o sinal

unidimensional de forma de onda X em representações numéricas multidimensionais Z. Essas representações Z alimentam o módulo de *Transformer* com o objetivo de gerar representações finais (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 Z em representações quantizadas Q. A organização da arquitetura está esquematizada na Figura 2.5.

O *Wav2Vec 2.0* apresenta uma vantagem por ser uma proposta de aprendizado auto-supervisionado [51], fazendo com que seja possível treinar boas representações de fala com dados não-supervisionados. A tarefa de pré-treinamento, também conhecida como tarefa de pretexto, é similar ao pré-treino do *BERT* [12], 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 [4] mostrou que é possível atingir *WER* de 2.0% com apenas 10% da base de dados para o *fine-tuning*, um valor bem próximo ao estado-da-arte (1.8% de *WER*), que considera treinamento com 100% da partição de treino, sem a utilização de pré-treino. O trabalho apresentado em [4] mostra que o pré-treino com dados não-supervisionados, inclusive, pode diminuir a quantidade de dados supervisionados necessários para o ajuste fino do modelo.

2.6.1 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 [11], que gerou um modelo de representação de fala treinado em 53 línguas diferentes e apresentou um resultado de 14,7% de *WER* 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 [50], que realizou um *fine-tuning* para tarefa de reconhecimento de fala com um conjunto de dados em português brasileiro. Uma *WER* 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 [26], 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.

2.7 Revisão da Literatura

2.7.1 Reconhecimento Automático de Fala

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* [38], 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) [4, 17, 19, 30, 39] 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 [31], 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* [49], *LapsMail* [37], *CoralBR* [48] e *GoogleVoice* [13] continham dados em português brasileiro, sendo apenas o *LapsMail* e o *CoralBR* conjuntos de dados públicos.

O trabalho apresentado em [6] utilizou os conjuntos de dados *LapsStory* [36], 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 [28] e outro contendo o Código de Defesa do Consumidor [28], 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* [49], 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 [43], 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 [34], os conjuntos de dados *Sid* [34] 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 [34] 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.
Recentemente, [26, 50] avaliaram a performance do *Wav2Vec2* por meio de um *fine-tuning* a partir do modelo pré-treinado disponibilizado em [11]. O modelo foi treinado em 53 línguas incluindo o português. Os trabalhos em [26, 50] 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.

2.7.2 Inteligência Artificial Centrada em Dados

Data-centric AI (DCAI) é um conceito emergente que coloca a qualidade e a dinâmica dos dados à frente das considerações de sistemas de Inteligência Artificial (IA). Isso significa que, em vez de priorizar os modelos, os dados passam a ter a maior importância para o desenvolvimento de sistemas de IA. Esse conceito é cada vez mais difundido e tem sido uma área de pesquisa em crescimento.
Uma abordagem DCAI propõe um método sistemático e iterativo para lidar com problemas de dados. Esta abordagem reconhece a dinâmica dos dados como uma infraestrutura em constante evolução para sistemas de IA [24]. Os princípios da DCAI reconhecem que os dados são uma construção sociotécnica, criados, manipulados e interpretados por seres humanos, de modo que uma abordagem centrada no ser humano é crítica para entender o significado dos dados [24].
Os trabalhos [35, 41] destacam a importância dos dados para o desenvolvimento de sistemas de IA. Enquanto a maioria dos trabalhos de pesquisa em aprendizado de máquina tem se concentrado nos modelos, os dados mais proeminentes têm sido usados para tarefas de aprendizado de máquina cotidianas sem considerar a amplitude, a dificuldade e a fidelidade desses dados em relação ao problema subjacente. O primeiro trabalho propõe o *DataPerf*, um pacote de *benchmarks* para avaliar conjuntos de dados e algoritmos de trabalho com dados de aprendizado de máquina. O objetivo é permitir o "engate de dados"na qual os conjuntos de treinamento ajudam a avaliar conjuntos de teste nos mesmos problemas e vice-versa. O segundo trabalho as lições que podemos transferir da engenharia de dados para DCAI: aplicações de dados e IA precisam ser executadas e treinadas continuamente, não apenas uma vez; os fluxos de

trabalho de implantação de produção são frequentemente centrados em código, não modelo ou dados centrados; o monitoramento de dados deve ser aplicável; o suporte ponta a ponta para versionamento de código e dados é extremamente útil; algumas aplicações não são permitidas mostrar dados para anotadores ou desenvolvedores humanos.
Alguns trabalhos propõem padronizações da DCAI, como o trabalho [55] que propõe alguns princípios para unificar os processos de desenvolvimento de dados treinos, como a coleta de dados, anotação de dados, preparação de dados, redução de dados e aumento de dados e também traz princípios para desenvolvimento de dados de inferência e manutenção de dados. Outro exemplo é o trabalho [18] que propõe algumas formalizações matemáticas para abordagens centradas em dados.

2.8 Conclusão

Os modelos de ponta a ponta com aprendizado autoaprendizado são os atuais estado da arte para o reconhecimento de fala em performance para a língua inglesa. Os modelos ponta a ponta, apesar de necessitarem de uma grande quantidade de recursos computacionais durante a sua fase de pré-treino, uma vez pré-treinados não necessitam de tantos recursos para os ajustes finos e podem ser utilizados para o aprimoramento de modelos em língua portuguesa, por exemplo.
Analisando as taxas de erro, os sistemas de RAF para português brasileiro avançaram nos últimos anos, inclusive em questão de recursos para treino e avaliação, utilizando modelos e técnicas já avaliados para outras línguas. Entretanto existe ainda um espaço para avaliar técnicas para redução das taxas de erro. Alguns trabalhos exploram o uso de modelos de língua para melhoria da taxa de erro entretanto não são exploradas o impacto do treinamento desses modelos de língua no resultado final da transcrição de áudios, deixando espaço para investigar o impacto dos dados para treinamento, quantitativamente e qualitativamente, na etapa de decodificação do processo reconhecimento de fala.

Capítulo 3

Metodologia

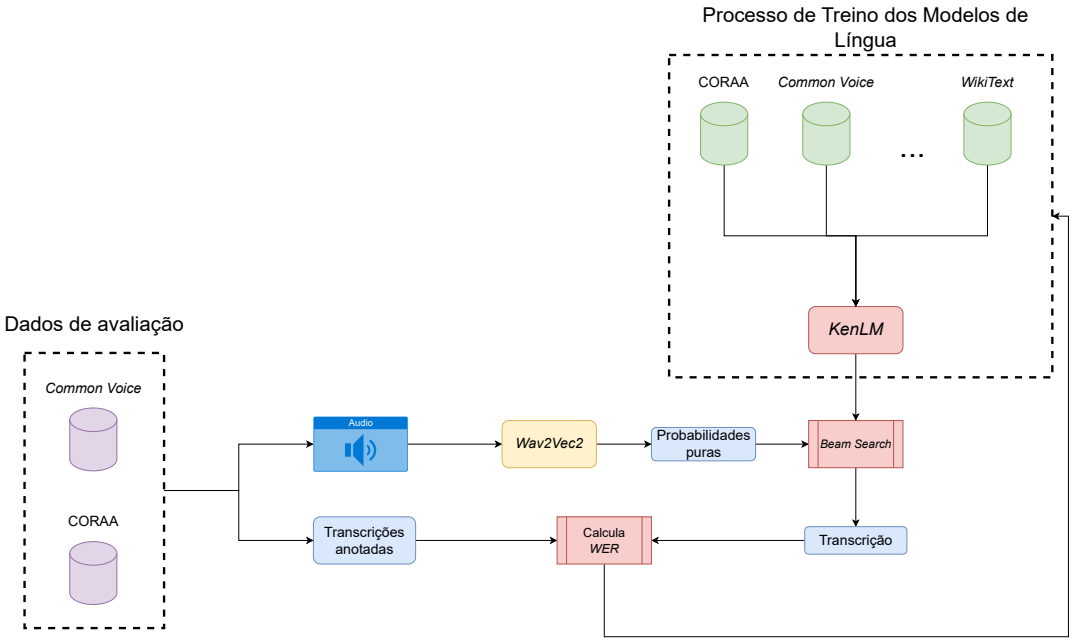

Figura 3.1: Ilustração da metodologia proposta para avaliação das técnicas em abordagem centrada em dados. Fonte: Autor

Esse trabalho faz uma análise centrada em dados, com proposta de metodologia para avaliação do impacto dos dados para treinamento em modelos de língua. A avaliação é feita com um método estado da arte para transcrição de áudios, o *Wav2Vec2* [11].

3.1 Protocolo de Avaliação e Métrica

Nesse trabalho foram avaliados os modelos baseados na *Wav2Vec2*, treinados em português 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 Seção 4.1 e também (ii) variando o a

largura do feixe na heurística *Beam Search*, conforme ilustrado na Figura 3.1. 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 das instâncias das partições de teste dos conjuntos de dados de áudio utilizando o *Wav2Vec2*.
3. Para cada combinação de conjuntos de dados treino 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*[1] 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:

$$WER = \frac{S + D + I}{S + D + C} \tag{3.1}$$

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 $I > C$.

[1]Disponível em: https://github.com/jitsi/jiwer

3.1.1 Normalização

Assim como no trabalho apresentado em [26], para limpeza e normalização dos conjuntos de dados textuais, foi realizado:

- Remoção da capitalização dos textos [26].
- Remoção de espaços duplicados [26].
- Padronização das pausas [26].
- Expansão de números [26].
- Normalização do símbolo % para "porcentagem"[26].
- Remoção de pontuação [26].

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.

3.1.2 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* [29], um método utilizado para comparação entre duas sequências de caractere. Este

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 $distancia$

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.

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* [25], 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.

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 $similaridade$

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:

$$sim = \frac{|Vocab_{teste} \cap Vocab_{treino}|}{|Vocab_{teste}|} \tag{3.2}$$

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

Capítulo 4

Experimentos

Neste capítulo são apresentados a configuração, resultados dos experimentos e análise dos erros.

4.1 Conjunto de dados

Neste trabalho foram utilizados uma variedade de conjuntos de dados a fim de investigar o impacto dos dados de treinamento, tanto em volume de textos quanto em tipo de conteúdo. Todos os conjuntos utilizados estão disponíveis de forma pública para pesquisa. Foram utilizados conjuntos de dados contendo áudio e suas transcrições e conjunto de dados contendo apenas textos. Foram utilizados os seguinte conjuntos de áudio e transcrição:

- *CommonVoice*[1] [3], 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)[2] [42], 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.

- CORAA[3] [26], 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;

[1] https://commonvoice.mozilla.org/pt/datasets
[2] https://www.openslr.org/94/
[3] https://github.com/nilc-nlp/CORAA

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.

Além dos conjuntos de áudio e transcrição, foram utilizado os seguintes conjuntos contendo apenas texto:

- CETENFolha[4] [32] um corpus textual extraído de textos da Folha de São Paulo, contendo cerca de 24 milhões de palavras.
- WikiText PT-BR[5] [34], um *dump* da Wikipedia de 2018 em português contendo cerca de 8,5 milhões de sentenças.

4.2 Configurações dos Experimentos

Algumas bases utilizadas nos experimentos desse trabalho utilizam as partições originais, como o caso das bases *Common Voice* 6.1, *Common Voice* 8.0, *MLS* e CORAA, que foram disponibilizadas com partições de treino e teste pelos autores. Já as bases *WikiText* PT-BR e CETENFolha não foram disponibilizadas em partições e nesse trabalho foram utilizadas completamente para treinamento.
Os experimentos foram realizados combinando as partições de treino entre os 6 conjuntos de dados. A combinação dos conjuntos (1 a 6) é feita conforme a Equação 4.1, 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 avaliada a variação da largura do feixe 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 a largura do feixe foram avaliados os valores: 10, 50 e 100.
Ambos os parâmetros n-gramas e largura do feixe foram escolhidos baseados em testes realizados em trabalhos anteriores[6, 34, 44], acrescentando a variação 4-gram e as variações 50 e 100 para a largura do feixe. Para a execução dos experimentos, foi utilizada uma GPU GeForce RTX 3090, com Processador 24-núcleos AMD Ryzen Threadripper 3960X 2.2 GHz e 128 GB of DDR4 RAM.

$$\sum_{k=1}^{6} \frac{6!}{k!(6-k)!} = 63 \tag{4.1}$$

[4]https://www.linguateca.pt/cetenfolha/index_info.html
[5]https://igormq.github.io/datasets/

Tabela 4.1: Taxa de erro detalhada por ordem do modelo de língua e a largura do feixe 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
Common Voice 6.1	X	X	X	X		X		100	0,07166955
	X	X	X	X		X	3	50	0,07294531
	X	X	X	X				10	0,08116690
	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
CORAA		X	X					100	0,37142562
		X	X				3	50	0,37408322
		X						10	0,39331562
		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

Tabela 4.2: 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%

4.3 Ajuste dos parâmetros

A Tabela 4.1 apresenta o resultado das melhores combinações de conjuntos de treino do modelo de língua, as variações do parâmetro de ordem do modelo de língua e a largura do feixe 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 da largura do feixe 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 $O(w|V|T)$ [33], em que w representa a largura do feixe, $|V|$ o tamanho do vocabulário e T o tamanho da sequência a ser decodificada. Sendo assim é necessário analisar a magnitude de melhoria dos resultados no momento em que for necessário implementar esse sistema em aplicações reais.

4.4 Impacto da qualidade dos dados

As Tabelas 4.2 e 4.3 apresentam o maior ganho, o menor ganho quando comparamos o modelo utilizando-se a heurística gulosa e por meio da Equação 4.2. Além disso as tabelas também apresentam o ganho inicial, quando treinamos o modelo de língua

Tabela 4.3: Melhoria dos modelos de língua em relação a heurística gulosa na partição de teste do CORAA

	Maior Ganho	Menor ganho	Ganho inicial
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 4.4: 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

apenas com um único conjunto de treino.

$$\frac{WER_{greedy} - WER_{beamsearch}}{WER_{greedy}} \tag{4.2}$$

Analisando a similaridade entre os conjuntos de dados de treino e a partição de teste, a Tabela 4.4 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 4.6 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.

4.5 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

Tabela 4.5: 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 4.6: Similaridade entre vocabulários das partições de treino com as partições 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 4.7: 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

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 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 com similaridade abaixo de 77% 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 [26] reporta uma *WER* de 24,18% para a partição de teste do CORAA; O trabalho apresentado em [50] 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.

Capítulo 5

Conclusões e Trabalhos Futuros

Neste Capítulo, dividimos as conclusões em duas seções, a Seção 5.1 apresenta as conclusões dos experimentos e a Seção 5.2 apresenta o cronograma de trabalhos futuros.

5.1 Conclusões

Este trabalho conduz uma investigação centrada em dados visando entender o impacto da qualidade e volume 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.

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.

5.2 Trabalhos Futuros

Esse trabalho investiga modelos de línguas baseados em contagem de palavras, com os notáveis avanços de modelos baseados em *Transformers*, como *BERT*, *GPT-2* e *GPT-3* é necessária a investigação do uso dessas técnicas em trabalhos futuros com a metodologia proposta.
Existe também espaço para investigação do comportamento em novos estado da arte como o *Whisper* [47].

Referências Bibliográficas

[1] ABDEL-HAMID, O., MOHAMED, A.-R., JIANG, H., et al., 2012, "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)*, pp. 4277–4280. doi: 10.1109/ICASSP.2012.6288864.

[2] AMODEI, D., ANANTHANARAYANAN, S., ANUBHAI, R., et al., 2016, "Deep speech 2: End-to-end speech recognition in english and mandarin". In: *International conference on machine learning*, pp. 173–182. PMLR.

[3] ARDILA, R., BRANSON, M., DAVIS, K., et al., 2020, "Common Voice: A Massively-Multilingual Speech Corpus". In: *Proceedings of the 12th Language Resources and Evaluation Conference*, pp. 4218–4222.

[4] BAEVSKI, A., ZHOU, Y., MOHAMED, A., et al., 2020, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations". In: Larochelle, H., Ranzato, M., Hadsell, R., et al. (Eds.), *Advances in Neural Information Processing Systems*, v. 33, pp. 12449–12460. Curran Associates, Inc. Disponível em: <https://proceedings.neurips.cc/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf>.

[5] BAEVSKI, A., HSU, W.-N., CONNEAU, A., et al., 2021, "Unsupervised Speech Recognition". In: Ranzato, M., Beygelzimer, A., Dauphin, Y., et al. (Eds.), *Advances in Neural Information Processing Systems*, v. 34, pp. 27826–27839. Curran Associates, Inc. Disponível em: <https://proceedings.neurips.cc/paper/2021/file/ea159dc9788ffac311592613b7f71fbb-Paper.pdf>.

[6] BATISTA, C., DIAS, A. L., SAMPAIO NETO, N., 2018, "Baseline Acoustic Models for Brazilian Portuguese Using Kaldi Tools". In: *Proc. IberSPEECH 2018*, pp. 77–81. doi: 10.21437/IberSPEECH.2018-17. Disponível em: <http://dx.doi.org/10.21437/IberSPEECH.2018-17>.

[7] BOURLARD, H., MORGAN, N., 1994, *Connectionist Speech Recognition: A Hybrid Approach*. ISBN: 978-1-4613-6409-2. doi: 10.1007/978-1-4615-3210-1.

[8] BROWN, T., MANN, B., RYDER, N., et al., 2020, "Language Models are Few-Shot Learners". In: Larochelle, H., Ranzato, M., Hadsell, R., et al. (Eds.), *Advances in Neural Information Processing Systems*, v. 33, pp. 1877–1901. Curran Associates, Inc. Disponível em: <https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf>.

[9] CHAN, W., JAITLY, N., LE, Q., et al., 2016, "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)*, pp. 4960–4964, .

[10] CHAN, W., JAITLY, N., LE, Q., et al., 2016, "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)*, pp. 4960–4964, .

[11] CONNEAU, A., BAEVSKI, A., COLLOBERT, R., et al., 2021, "Unsupervised Cross-Lingual Representation Learning for Speech Recognition". In: *Proc. Interspeech 2021*, pp. 2426–2430. doi: 10.21437/Interspeech.2021-329.

[12] DEVLIN, J., CHANG, M.-W., LEE, K., et al., 2019, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186.

[13] GONZALEZ-DOMINGUEZ, J., LOPEZ-MORENO, I., MORENO, P. J., et al., 2015, "Frame-by-frame language identification in short utterances using deep neural networks", *Neural Networks*, v. 64, pp. 49–58. ISSN: 0893-6080. doi: https://doi.org/10.1016/j.neunet.2014.08.006. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0893608014002019>. Special Issue on "Deep Learning of Representations".

[14] GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., et al., 2020, "Generative Adversarial Networks", *Commun. ACM*, v. 63, n. 11 (oct), pp. 139–144. ISSN: 0001-0782. doi: 10.1145/3422622. Disponível em: <https://doi.org/10.1145/3422622>.

[15] GRAVES, A., JAITLY, N., 2014, "Towards End-To-End Speech Recognition with Recurrent Neural Networks". In: Xing, E. P., Jebara, T. (Eds.), *Proceedings of the 31st International Conference on Machine Learning*, v. 32, *Proceedings of Machine Learning Research*, pp. 1764–1772, Bejing, China, 22–24 Jun. PMLR. Disponível em: <http://proceedings.mlr.press/v32/graves14.html>.

[16] GRAVES, A., FERNÁNDEZ, S., GOMEZ, F., et al., 2006, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks". In: *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, p. 369–376, New York, NY, USA. Association for Computing Machinery. ISBN: 1595933832. doi: 10.1145/1143844.1143891. Disponível em: <https://doi.org/10.1145/1143844.1143891>.

[17] GULATI, A., QIN, J., CHIU, C.-C., et al., 2020, "Conformer: Convolution-augmented Transformer for Speech Recognition". In: *Proc. Interspeech 2020*, pp. 5036–5040. doi: 10.21437/Interspeech.2020-3015.

[18] HAJIJ, M., ZAMZMI, G., RAMAMURTHY, K. N., et al., 2021, "Data-Centric AI Requires Rethinking Data Notion", *CoRR*, v. abs/2110.02491. Disponível em: <https://arxiv.org/abs/2110.02491>.

[19] HAN, W., ZHANG, Z., ZHANG, Y., et al., 2020, "ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context". In: *Proc. Interspeech 2020*, pp. 3610–3614. doi: 10.21437/Interspeech.2020-2059.

[20] HE, Y., SAINATH, T. N., PRABHAVALKAR, R., et al., 2019, "Streaming end-to-end speech recognition for mobile devices". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6381–6385. IEEE.

[21] HEAFIELD, K., 2011, "KenLM: Faster and Smaller Language Model Queries". In: *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pp. 187–197, Edinburgh, Scotland, jul. Association for Computational Linguistics. Disponível em: <https://aclanthology.org/W11-2123>.

[22] HINTON, G., DENG, L., YU, D., et al., 2012, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups", *IEEE Signal Processing Magazine*, v. 29, n. 6, pp. 82–97. doi: 10.1109/MSP.2012.2205597.

[23] HSU, W.-N., TSAI, Y.-H. H., BOLTE, B., et al., 2021, "Hubert: How Much Can a Bad Teacher Benefit ASR Pre-Training?" In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6533–6537. doi: 10.1109/ICASSP39728.2021.9414460.

[24] JARRAHI, M. H., MEMARIANI, A., GUHA, S., 2022. "The Principles of Data-Centric AI (DCAI)". .

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

[26] JUNIOR, A. C., CASANOVA, E., SOARES, A., et al., 2022, "CORAA ASR: a large corpus of spontaneous and prepared speech manually validated for speech recognition in Brazilian Portuguese", *Language Resources and Evaluation*, (nov.). doi: 10.1007/s10579-022-09621-4. Disponível em: <https://doi.org/10.1007/s10579-022-09621-4>.

[27] LECUN, Y., BOTTOU, L., BENGIO, Y., et al., 1998, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, v. 86, n. 11, pp. 2278–2324. doi: 10.1109/5.726791.

[28] LEGAL, P., 2018. "PCD legal: Acessível para todos". Disponível em: <http://www.pcdlegal.com.br/>.

[29] LEVENSHTEIN, V. I., 1966, "Binary Codes Capable of Correcting Deletions, Insertions and Reversals", *Soviet Physics Doklady*, v. 10 (fev.), pp. 707.

[30] LI, J., LAVRUKHIN, V., GINSBURG, B., et al., 2019, "Jasper: An End-to-End Convolutional Neural Acoustic Model". In: *Proc. Interspeech 2019*, pp. 71–75. doi: 10.21437/Interspeech.2019-1819.

[31] LIMA, T., DA COSTA-ABREU, M., 2019, "A Survey on Automatic Speech Recognition Systems for Portuguese Language and its Variations", *Computer Speech & Language*, v. 62 (12), pp. 101055. doi: 10.1016/j.csl.2019.101055.

[32] LINGUATECA. "CETENFolha". Disponível em: <https://www.linguateca.pt/cetenfolha/>.

[33] LOWERRE, B. T., 1976, *The Harpy Speech Recognition System*. Tese de Doutorado, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213, 4.

[34] MACEDO QUINTANILHA, I., LIMA NETTO, S., PEREIRA BISCAINHO, L., 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, v. 35, n. 1 (Sep.), pp. 230–242. doi: 10.14209/jcis.2020.25. Disponível em: <https://jcis.sbrt.org.br/jcis/article/view/721>.

[35] MAZUMDER, M., BANBURY, C., YAO, X., et al., 2022. "DataPerf: Benchmarks for Data-Centric AI Development". .

[36] NETO, N., PATRICK, C., KLAUTAU, A., et al., 2011, "Free tools and resources for Brazilian Portuguese speech recognition", *Journal of the Brazilian Computer Society*, v. 17, n. 1, pp. 53–68.

[37] OLIVEIRA, R., BATISTA, P., NETO, N., et al., 2012, "Baseline Acoustic Models for Brazilian Portuguese Using CMU Sphinx Tools". In: Caseli, H., Villavicencio, A., Teixeira, A., et al. (Eds.), *Computational Processing of the Portuguese Language*, pp. 375–380, Berlin, Heidelberg. Springer Berlin Heidelberg. ISBN: 978-3-642-28885-2.

[38] PANAYOTOV, V., CHEN, G., POVEY, D., et al., 2015, "Librispeech: An ASR corpus based on public domain audio books". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210.

[39] PARK, D. S., CHAN, W., ZHANG, Y., et al., 2019, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition". In: *Proc. Interspeech 2019*, pp. 2613–2617. doi: 10.21437/Interspeech.2019-2680.

[40] PARK, D. S., ZHANG, Y., JIA, Y., et al., 2020, "Improved Noisy Student Training for Automatic Speech Recognition", *Interspeech 2020*, (Oct). doi: 10.21437/interspeech.2020-1470. Disponível em: <http://dx.doi.org/10.21437/Interspeech.2020-1470>.

[41] POLYZOTIS, N., ZAHARIA, M., 2021, "What can Data-Centric AI Learn from Data and ML Engineering?" *CoRR*, v. abs/2112.06439. Disponível em: <https://arxiv.org/abs/2112.06439>.

[42] PRATAP, V., XU, Q., SRIRAM, A., et al., 2020, "MLS: A Large-Scale Multilingual Dataset for Speech Research", *Interspeech 2020*, (Oct). doi: 10.21437/interspeech.2020-2826. Disponível em: <http://dx.doi.org/10.21437/Interspeech.2020-2826>.

[43] PUC-RIO. "Centro de Estudos em Telecomunicações (CETUC)". Disponível em: <http://www.cetuc.puc-rio.br/>.

[44] QUINTANILHA, I. M., BISCAINHO, L. W. P., NETTO, S. L., 2017, "Towards an end-to-end speech recognizer for Portuguese using deep neural networks", *XXXV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, pp. 709–714.

[45] RADFORD, A., NARASIMHAN, K., SALIMANS, T., et al., 2018, "Improving language understanding by generative pre-training", .

[46] RADFORD, A., WU, J., CHILD, R., et al., 2019, "Language Models are Unsupervised Multitask Learners", .

[47] RADFORD, A., KIM, J. W., XU, T., et al., 2022, "Robust Speech Recognition via Large-Scale Weak Supervision", *arXiv e-prints*, art. arXiv:2212.04356. doi: 10.48550/arXiv.2212.04356.

[48] RASO, T., MELLO, H., 2012, "The C-ORAL-BRASIL I: Reference Corpus for Informal Spoken Brazilian Portuguese". In: Caseli, H., Villavicencio, A., Teixeira, A., et al. (Eds.), *Computational Processing of the Portuguese Language*, pp. 362–367, Berlin, Heidelberg. Springer Berlin Heidelberg. ISBN: 978-3-642-28885-2.

[49] SCHRAMM, M., FREITAS, L., ZANUZ, A., et al., 2006. "CSLU: Spoltech Brazilian Portuguese version 1.0 LDC2006S16". .

[50] STEFANEL GRIS, L. R., CASANOVA, E., DE OLIVEIRA, F. S., et al., 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*, pp. 333–343. Springer.

[51] TIAN, Y., YU, L., CHEN, X., et al., 2020, "Understanding Self-supervised Learning with Dual Deep Networks", *CoRR*, v. abs/2010.00578. Disponível em: <https://arxiv.org/abs/2010.00578>.

[52] VASWANI, A., SHAZEER, N., PARMAR, N., et al., 2017, "Attention is All you Need". In: Guyon, I., Luxburg, U. V., Bengio, S., et al. (Eds.), *Advances in Neural Information Processing Systems*, v. 30. Curran Associates, Inc. Disponível em: <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.

[53] XU, Q., BAEVSKI, A., LIKHOMANENKO, T., et al., 2021, "Self-training and pre-training are complementary for speech recognition". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3030–3034. IEEE.

[54] YANG, X., LI, J., ZHOU, X., 2018, "A novel pyramidal-FSMN architecture with lattice-free MMI for speech recognition", *CoRR*, v. abs/1810.11352. Disponível em: <http://arxiv.org/abs/1810.11352>.

[55] ZHA, D., BHAT, Z. P., LAI, K.-H., et al., 2023. "Data-centric AI: Perspectives and Challenges". .

Apêndice A

Apêndice

A.1 Resultado dos Experimentos

A Tabela A.1 apresenta os resultados de cada combinação de cada conjunto de dados avaliando na partição de teste do *Common Voice 6.1*.
A Tabela A.2 apresenta os resultados de cada combinação de cada conjunto de dados avaliando na partição de teste do CORAA.

Tabela A.1: Taxa de erro de cada combinação de conjunto de treinamento avaliado na partição de teste do *Common Voice 6.1*

CETENFolha	**CORAA**	**CV 6.1**	**CV 8.0**	**MLS**	**WikiText**	**3-gram**	**4-gram**	**5-gram**	**WER**	**Melhoria no WER**	**Tamanho do ML**
X		X	X	X	X			X	7,14%	33,25%	17,31 GB
X			X	X	X			X	7,15%	33,22%	17,19 GB
X	X	X	X	X	X			X	7,16%	33,06%	17,49 GB
X	X	X	X		X	X			7,17%	33,03%	3,43 GB
X	X		X	X	X			X	7,17%	33,03%	17,37 GB
X	X	X	X		X			X	7,17%	33,03%	17,07 GB
X	X		X		X	X			7,17%	33,01%	3,43 GB
X	X		X		X			X	7,17%	33,01%	17,23 GB
X		X	X		X			X	7,17%	32,98%	16,86 GB
X		X	X	X	X		X		7,18%	32,95%	9,19 GB
X	X	X	X		X		X		7,18%	32,93%	9,18 GB
X	X	X	X	X	X		X		7,18%	32,93%	9,22 GB
X			X	X	X		X		7,18%	32,93%	9,19 GB
X			X		X			X	7,18%	32,90%	17,07 GB
X		X	X		X	X			7,18%	32,90%	3,39 GB
X	X		X	X	X		X		7,18%	32,90%	9,29 GB
X	X		X		X		X		7,18%	32,90%	9,21 GB
X		X	X		X		X		7,18%	32,87%	9,04 GB
X			X		X	X			7,19%	32,85%	3,37 GB
X		X	X	X	X	X			7,19%	32,82%	3,41 GB
X			X	X	X	X			7,19%	32,79%	3,41 GB
X			X		X		X		7,19%	32,79%	8,99 GB
	X	X	X		X			X	7,20%	32,77%	15,39 GB
	X		X		X			X	7,20%	32,74%	15,35 GB
X	X	X	X	X	X	X			7,20%	32,74%	3,46 GB

Tabela A.1: Taxa de erro de cada combinação de conjunto de treinamento avaliado na partição de teste do *Common Voice 6.1*

CETENFolha	**CORAA**	**CV 6.1**	**CV 8.0**	**MLS**	**WikiText**	**3-gram**	**4-gram**	**5-gram**	**WER**	**Melhoria no WER**	**Tamanho do ML**
X	X		X	X	X	X			7,20%	32,72%	3,46 GB
		X	X	X	X			X	7,20%	32,69%	15,44 GB
		X	X		X			X	7,21%	32,66%	15,20 GB
			X	X	X			X	7,21%	32,66%	15,37 GB
			X		X			X	7,21%	32,64%	15,27 GB
X	X	X			X	X			7,21%	32,64%	3,41 GB
	X	X	X	X	X			X	7,21%	32,64%	15,62 GB
	X		X	X	X			X	7,21%	32,61%	15,60 GB
X		X		X	X			X	7,21%	32,61%	17,08 GB
		X	X	X	X		X		7,22%	32,56%	8,26 GB
X		X			X	X			7,22%	32,56%	3,39 GB
X	X	X			X			X	7,22%	32,53%	17,34 GB
			X	X	X		X		7,22%	32,53%	8,26 GB
	X		X		X	X			7,22%	32,50%	3,09 GB
X	X	X		X	X			X	7,22%	32,50%	17,50 GB
	X	X	X		X	X			7,22%	32,50%	3,09 GB
		X	X		X		X		7,23%	32,48%	8,19 GB
	X		X	X	X	X			7,23%	32,45%	3,11 GB
	X	X	X	X	X	X			7,23%	32,45%	3,14 GB
			X		X		X		7,23%	32,45%	8,16 GB
X		X		X	X		X		7,23%	32,42%	9,20 GB
X	X	X			X		X		7,23%	32,42%	9,20 GB
X	X	X		X	X		X		7,23%	32,42%	9,23 GB
X		X			X			X	7,23%	32,40%	17,16 GB
X		X			X		X		7,24%	32,37%	9,10 GB
X		X		X	X	X			7,24%	32,37%	3,41 GB

Tabela A.1: Taxa de erro de cada combinação de conjunto de treinamento avaliado na partição de teste do *Common Voice 6.1*

CETENFolha	**CORAA**	**CV 6.1**	**CV 8.0**	**MLS**	**WikiText**	**3-gram**	**4-gram**	**5-gram**	**WER**	**Melhoria no WER**	**Tamanho do ML**
	X	X	X	X	X		X		7,24%	32,37%	8,36 GB
	X	X	X		X		X		7,24%	32,37%	8,30 GB
X	X	X		X	X	X			7,24%	32,37%	3,44 GB
	X		X		X		X		7,24%	32,34%	8,26 GB
	X		X	X	X		X		7,24%	32,34%	8,38 GB
	X	X		X	X	X			7,26%	32,16%	3,12 GB
	X	X			X	X			7,26%	32,16%	3,11 GB
	X	X			X			X	7,27%	32,08%	15,54 GB
			X		X	X			7,27%	32,08%	3,05 GB
		X	X		X	X			7,27%	32,08%	3,07 GB
		X			X			X	7,27%	32,05%	15,37 GB
		X		X	X			X	7,27%	32,05%	15,49 GB
	X	X		X	X			X	7,27%	32,05%	15,62 GB
X	X				X	X			7,28%	32,00%	3,41 GB
			X	X	X	X			7,28%	31,97%	3,09 GB
		X	X	X	X	X			7,28%	31,95%	3,09 GB
		X		X	X		X		7,28%	31,95%	8,26 GB
X				X	X			X	7,29%	31,87%	17,30 GB
X					X	X			7,29%	31,84%	3,37 GB
	X	X		X	X		X		7,30%	31,81%	8,29 GB
X					X		X		7,30%	31,81%	9,10 GB
		X			X		X		7,30%	31,81%	8,19 GB
X				X	X		X		7,30%	31,76%	9,18 GB
	X	X			X		X		7,30%	31,76%	8,27 GB
X	X				X			X	7,31%	31,74%	17,24 GB
X				X	X	X			7,31%	31,74%	3,41 GB

Tabela A.1: Taxa de erro de cada combinação de conjunto de treinamento avaliado na partição de teste do *Common Voice 6.1*

CETENFolha	**CORAA**	**CV 6.1**	**CV 8.0**	**MLS**	**WikiText**	**3-gram**	**4-gram**	**5-gram**	**WER**	**Melhoria no WER**	**Tamanho do ML**
		X			X	X			7,31%	31,74%	3,07 GB
X	X			X	X	X			7,31%	31,74%	3,44 GB
		X		X	X	X			7,31%	31,71%	3,09 GB
X	X			X	X			X	7,31%	31,71%	17,50 GB
X	X			X	X		X		7,31%	31,68%	9,23 GB
X	X				X		X		7,31%	31,66%	9,22 GB
X					X			X	7,31%	31,66%	16,96 GB
	X				X	X			7,33%	31,52%	3,07 GB
	X			X	X	X			7,33%	31,47%	3,12 GB
X	X		X				X		7,34%	31,44%	1,34 GB
X	X	X	X				X		7,34%	31,44%	1,34 GB
X	X	X	X					X	7,34%	31,42%	2,35 GB
X	X		X					X	7,34%	31,42%	2,34 GB
					X			X	7,34%	31,39%	15,28 GB
	X			X	X			X	7,34%	31,39%	15,70 GB
					X	X			7,35%	31,36%	3,05 GB
				X	X			X	7,35%	31,36%	15,42 GB
	X				X			X	7,35%	31,34%	15,53 GB
				X	X		X		7,35%	31,31%	8,19 GB
					X		X		7,36%	31,23%	8,17 GB
				X	X	X			7,37%	31,18%	3,10 GB
	X			X	X		X		7,37%	31,18%	8,31 GB
	X				X		X		7,37%	31,15%	8,27 GB
X	X		X	X			X		7,37%	31,15%	1,43 GB
X	X	X	X	X			X		7,37%	31,13%	1,43 GB
X	X		X	X				X	7,37%	31,13%	2,51 GB

Tabela A.1: Taxa de erro de cada combinação de conjunto de treinamento avaliado na partição de teste do *Common Voice 6.1*

CETENFolha	**CORAA**	**CV 6.1**	**CV 8.0**	**MLS**	**WikiText**	**3-gram**	**4-gram**	**5-gram**	**WER**	**Melhoria no WER**	**Tamanho do ML**
X	X	X	X	X				X	7,37%	31,10%	2,52 GB
X	X	X	X			X			7,39%	30,94%	547,56 MB
X	X		X			X			7,39%	30,94%	547,87 MB
X		X	X				X		7,41%	30,78%	1,22 GB
X	X		X	X		X			7,41%	30,75%	585,53 MB
X			X				X		7,41%	30,75%	1,23 GB
X	X	X	X	X		X			7,41%	30,73%	585,77 MB
X	X	X					X		7,41%	30,73%	1,33 GB
X	X	X						X	7,42%	30,70%	2,35 GB
X	X	X				X			7,44%	30,49%	546,62 MB
X		X	X					X	7,44%	30,49%	2,17 GB
X			X					X	7,44%	30,49%	2,17 GB
X	X	X		X			X		7,44%	30,46%	1,43 GB
X		X	X	X			X		7,45%	30,41%	1,32 GB
X	X	X		X				X	7,45%	30,38%	2,52 GB
X			X	X			X		7,45%	30,36%	1,32 GB
X		X	X	X				X	7,46%	30,28%	2,31 GB
X		X	X			X			7,46%	30,25%	502,90 MB
X			X	X				X	7,46%	30,25%	2,31 GB
X	X	X		X		X			7,46%	30,25%	588,08 MB
X			X			X			7,46%	30,25%	496,32 MB
X		X	X	X		X			7,51%	29,85%	538,26 MB
X			X	X		X			7,51%	29,83%	541,76 MB
X		X					X		7,52%	29,75%	1,22 GB
X		X				X			7,53%	29,62%	498,14 MB
X		X						X	7,53%	29,62%	2,16 GB

Tabela A.1: Taxa de erro de cada combinação de conjunto de treinamento avaliado na partição de teste do *Common Voice 6.1*

CETENFolha	**CORAA**	**CV 6.1**	**CV 8.0**	**MLS**	**WikiText**	**3-gram**	**4-gram**	**5-gram**	**WER**	**Melhoria no WER**	**Tamanho do ML**
X		X		X			X		7,56%	29,38%	1,32 GB
X		X		X				X	7,57%	29,25%	2,30 GB
X	X							X	7,58%	29,19%	2,36 GB
X	X						X		7,58%	29,19%	1,34 GB
X		X		X		X			7,59%	29,09%	540,32 MB
X	X					X			7,59%	29,06%	538,74 MB
X	X			X			X		7,61%	28,87%	1,43 GB
X	X			X		X			7,61%	28,85%	583,48 MB
X	X			X				X	7,62%	28,79%	2,50 GB
X							X		7,71%	27,95%	1,21 GB
X						X			7,71%	27,95%	497,16 MB
X								X	7,72%	27,87%	2,15 GB
X				X			X		7,76%	27,50%	1,31 GB
X				X		X			7,77%	27,44%	539,36 MB
X				X				X	7,77%	27,42%	2,32 GB
	X	X	X	X		X			7,90%	26,17%	119,17 MB
	X	X	X	X				X	7,91%	26,12%	410,34 MB
	X	X	X	X			X		7,91%	26,07%	255,16 MB
	X		X	X		X			7,92%	26,01%	118,15 MB
	X	X	X				X		7,92%	25,99%	153,52 MB
	X	X	X					X	7,92%	25,99%	244,87 MB
	X	X	X			X			7,92%	25,96%	69,95 MB
	X		X	X				X	7,93%	25,93%	407,68 MB
	X		X	X			X		7,94%	25,85%	252,68 MB
	X		X					X	7,94%	25,77%	246,08 MB
	X		X			X			7,95%	25,75%	69,82 MB

Tabela A.1: Taxa de erro de cada combinação de conjunto de treinamento avaliado na partição de teste do *Common Voice 6.1*

CETENFolha	**CORAA**	**CV 6.1**	**CV 8.0**	**MLS**	**WikiText**	**3-gram**	**4-gram**	**5-gram**	**WER**	**Melhoria no WER**	**Tamanho do ML**
	X		X				X		7,95%	25,75%	153,26 MB
	X	X		X				X	8,02%	25,09%	404,41 MB
	X	X		X			X		8,02%	25,03%	251,31 MB
	X	X		X		X			8,03%	24,93%	117,29 MB
	X	X				X			8,08%	24,50%	67,88 MB
	X	X					X		8,09%	24,37%	149,43 MB
	X	X						X	8,10%	24,32%	241,96 MB
	X			X		X			8,37%	21,75%	115,95 MB
	X			X				X	8,38%	21,70%	398,83 MB
	X			X			X		8,39%	21,62%	249,32 MB
	X					X			8,46%	20,98%	66,48 MB
	X							X	8,49%	20,72%	236,08 MB
	X						X		8,50%	20,61%	146,94 MB
		X	X	X			X		8,53%	20,32%	117,73 MB
		X	X	X				X	8,53%	20,26%	185,34 MB
		X	X	X		X			8,54%	20,21%	59,23 MB
		X	X			X			8,54%	20,19%	5,49 MB
		X	X					X	8,55%	20,13%	13,39 MB
		X	X				X		8,55%	20,08%	9,43 MB
			X	X			X		8,57%	19,95%	117,49 MB
			X	X				X	8,57%	19,89%	183,72 MB
			X					X	8,58%	19,87%	13,04 MB
			X	X		X			8,58%	19,87%	58,56 MB
			X				X		8,58%	19,84%	9,17 MB
			X			X			8,59%	19,76%	5,27 MB
		X		X			X		8,83%	17,46%	113,37 MB

Tabela A.1: Taxa de erro de cada combinação de conjunto de treinamento avaliado na partição de teste do *Common Voice 6.1*

CETENFolha	**CORAA**	**CV 6.1**	**CV 8.0**	**MLS**	**WikiText**	**3-gram**	**4-gram**	**5-gram**	**WER**	**Melhoria no WER**	**Tamanho do ML**
		X		X				X	8,84%	17,40%	176,46 MB
		X		X		X			8,84%	17,40%	56,96 MB
		X						X	9,26%	13,48%	6,19 MB
		X				X			9,26%	13,43%	2,51 MB
		X					X		9,27%	13,38%	4,35 MB
				X				X	9,89%	7,55%	173,00 MB
				X		X			9,90%	7,52%	54,14 MB
				X			X		9,90%	7,47%	109,35 MB

Tabela A.2: Taxa de erro de cada combinação de conjunto de treinamento avaliado na partição de teste do CORAA.

CETENFolha	**CORAA**	**CV 6.1**	**CV 8.0**	**MLS**	**WikiText**	**3-gram**	**4-gram**	**5-gram**	**WER**	**Melhoria no WER**	**Tamanho do ML**
	X	X				X			37,14%	17,86%	67,88 MB
	X	X	X				X		37,15%	17,83%	153,52 MB
	X		X				X		37,15%	17,83%	153,26 MB
	X					X			37,15%	17,83%	66,48 MB
	X	X	X			X			37,16%	17,82%	69,95 MB
	X		X			X			37,16%	17,81%	69,82 MB
	X	X					X		37,17%	17,81%	149,43 MB
	X						X		37,17%	17,80%	146,94 MB
	X	X	X					X	37,19%	17,76%	244,87 MB
	X		X					X	37,19%	17,75%	246,08 MB
	X							X	37,20%	17,72%	236,08 MB
	X	X						X	37,21%	17,70%	241,96 MB
X	X		X				X		37,48%	17,11%	1,34 GB

Tabela A.2: Taxa de erro de cada combinação de conjunto de treinamento avaliado na partição de teste do CORAA.

CETENFolha	CORAA	CV 6.1	CV 8.0	MLS	WikiText	3-gram	4-gram	5-gram	WER	Melhoria no WER	Tamanho do ML
X	X	X	X				X		37,48%	17,11%	1,34 GB
	X	X		X		X			37,49%	17,10%	117,29 MB
X	X						X		37,49%	17,09%	1,34 GB
X	X	X					X		37,49%	17,09%	1,33 GB
	X			X		X			37,49%	17,09%	115,95 MB
	X	X	X	X		X			37,49%	17,09%	119,17 MB
X	X		X					X	37,49%	17,08%	2,34 GB
X	X	X	X					X	37,50%	17,08%	2,35 GB
	X		X	X		X			37,50%	17,08%	118,15 MB
X	X	X						X	37,51%	17,05%	2,35 GB
X	X							X	37,51%	17,04%	2,36 GB
	X	X		X			X		37,51%	17,04%	251,31 MB
	X	X	X	X			X		37,52%	17,03%	255,16 MB
	X		X	X			X		37,52%	17,03%	252,68 MB
	X			X			X		37,53%	17,01%	249,32 MB
X	X	X				X			37,54%	16,98%	546,62 MB
X	X		X			X			37,54%	16,98%	547,87 MB
X	X	X	X			X			37,54%	16,98%	547,56 MB
X	X					X			37,55%	16,95%	538,74 MB
	X	X		X				X	37,55%	16,95%	404,41 MB
	X		X	X				X	37,56%	16,93%	407,68 MB
	X	X	X	X				X	37,56%	16,93%	410,34 MB
X	X	X	X	X			X		37,59%	16,87%	1,43 GB
	X			X				X	37,59%	16,87%	398,83 MB
X	X		X	X			X		37,59%	16,87%	1,43 GB
X	X	X		X			X		37,59%	16,86%	1,43 GB

Tabela A.2: Taxa de erro de cada combinação de conjunto de treinamento avaliado na partição de teste do CORAA.

CETENFolha	CORAA	CV 6.1	CV 8.0	MLS	WikiText	3-gram	4-gram	5-gram	WER	Melhoria no WER	Tamanho do ML
X	X			X			X		37,60%	16,85%	1,43 GB
X	X			X				X	37,62%	16,81%	2,50 GB
X	X	X		X				X	37,62%	16,80%	2,52 GB
X	X	X	X	X				X	37,62%	16,80%	2,52 GB
X	X		X	X				X	37,62%	16,79%	2,51 GB
X	X			X		X			37,63%	16,78%	583,48 MB
X	X	X		X		X			37,63%	16,78%	588,08 MB
X	X		X	X		X			37,64%	16,76%	585,53 MB
X	X	X	X	X		X			37,64%	16,75%	585,77 MB
X	X	X			X		X		38,29%	15,33%	9,20 GB
X	X				X		X		38,29%	15,33%	9,22 GB
X	X	X	X		X		X		38,29%	15,32%	9,18 GB
X	X		X		X		X		38,29%	15,32%	9,21 GB
X	X				X	X			38,30%	15,29%	3,41 GB
X	X			X	X		X		38,32%	15,26%	9,23 GB
X	X	X			X			X	38,32%	15,26%	17,34 GB
X	X	X		X	X		X		38,32%	15,25%	9,23 GB
X	X	X	X		X			X	38,32%	15,25%	17,07 GB
X	X		X		X			X	38,32%	15,25%	17,23 GB
X	X	X	X	X	X		X		38,32%	15,25%	9,22 GB
X	X		X	X	X		X		38,32%	15,24%	9,29 GB
X	X	X			X	X			38,33%	15,24%	3,41 GB
X	X	X	X		X	X			38,33%	15,24%	3,43 GB
X	X		X		X	X			38,33%	15,24%	3,43 GB
X	X				X			X	38,33%	15,23%	17,24 GB
X	X	X		X	X			X	38,34%	15,21%	17,50 GB

Tabela A.2: Taxa de erro de cada combinação de conjunto de treinamento avaliado na partição de teste do CORAA.

CETENFolha	CORAA	CV 6.1	CV 8.0	MLS	WikiText	3-gram	4-gram	5-gram	WER	Melhoria no WER	Tamanho do ML
X	X	X	X	X	X			X	38,34%	15,21%	17,49 GB
	X	X			X		X		38,34%	15,21%	8,27 GB
X	X		X	X	X			X	38,34%	15,21%	17,37 GB
	X		X		X		X		38,34%	15,21%	8,26 GB
	X	X	X		X		X		38,34%	15,21%	8,30 GB
X	X			X	X	X			38,34%	15,21%	3,44 GB
	X				X	X			38,34%	15,21%	3,07 GB
X	X			X	X			X	38,34%	15,20%	17,50 GB
	X				X		X		38,35%	15,20%	8,27 GB
X	X	X		X	X	X			38,35%	15,19%	3,44 GB
X	X		X	X	X	X			38,35%	15,18%	3,46 GB
X	X	X	X	X	X	X			38,35%	15,18%	3,46 GB
	X	X			X			X	38,36%	15,17%	15,54 GB
	X	X			X	X			38,36%	15,16%	3,11 GB
	X				X			X	38,36%	15,16%	15,53 GB
	X			X	X		X		38,36%	15,16%	8,31 GB
	X	X	X		X			X	38,37%	15,15%	15,39 GB
	X		X		X			X	38,37%	15,15%	15,35 GB
	X	X	X	X	X		X		38,37%	15,15%	8,36 GB
	X		X	X	X		X		38,37%	15,15%	8,38 GB
	X		X		X	X			38,37%	15,14%	3,09 GB
	X	X	X		X	X			38,37%	15,14%	3,09 GB
	X	X		X	X		X		38,37%	15,14%	8,29 GB
	X			X	X	X			38,38%	15,11%	3,12 GB
	X	X	X	X	X			X	38,39%	15,10%	15,62 GB
	X	X		X	X			X	38,39%	15,10%	15,62 GB

Tabela A.2: Taxa de erro de cada combinação de conjunto de treinamento avaliado na partição de teste do CORAA.

CETENFolha	**CORAA**	**CV 6.1**	**CV 8.0**	**MLS**	**WikiText**	**3-gram**	**4-gram**	**5-gram**	**WER**	**Melhoria no WER**	**Tamanho do ML**
	X		X	X	X			X	38,39%	15,10%	15,60 GB
	X			X	X			X	38,40%	15,09%	15,70 GB
	X	X	X	X	X	X			38,42%	15,04%	3,14 GB
	X	X		X	X	X			38,42%	15,04%	3,12 GB
	X		X	X	X	X			38,42%	15,04%	3,11 GB
X		X				X			38,48%	14,89%	498,14 MB
X						X			38,49%	14,88%	497,16 MB
X		X	X			X			38,49%	14,87%	502,90 MB
X			X			X			38,50%	14,86%	496,32 MB
X		X					X		38,50%	14,85%	1,22 GB
X		X	X				X		38,51%	14,83%	1,22 GB
X			X				X		38,52%	14,82%	1,23 GB
X							X		38,52%	14,81%	1,21 GB
X		X						X	38,54%	14,76%	2,16 GB
X		X	X					X	38,55%	14,75%	2,17 GB
X								X	38,55%	14,75%	2,15 GB
X			X					X	38,55%	14,75%	2,17 GB
X		X	X	X		X			38,59%	14,65%	538,26 MB
X			X	X		X			38,60%	14,65%	541,76 MB
X		X		X		X			38,60%	14,65%	540,32 MB
X				X		X			38,60%	14,63%	539,36 MB
X		X		X			X		38,60%	14,63%	1,32 GB
X		X	X	X			X		38,60%	14,62%	1,32 GB
X			X	X			X		38,61%	14,62%	1,32 GB
X		X		X				X	38,61%	14,62%	2,30 GB
X		X	X	X				X	38,61%	14,61%	2,31 GB

Tabela A.2: Taxa de erro de cada combinação de conjunto de treinamento avaliado na partição de teste do CORAA.

CETENFolha	CORAA	CV 6.1	CV 8.0	MLS	WikiText	3-gram	4-gram	5-gram	WER	Melhoria no WER	Tamanho do ML
X			X	X				X	38,61%	14,60%	2,31 GB
X				X			X		38,62%	14,59%	1,31 GB
X				X				X	38,62%	14,58%	2,32 GB
X					X	X			39,05%	13,63%	3,37 GB
X				X	X		X		39,06%	13,62%	9,18 GB
X					X		X		39,06%	13,62%	9,10 GB
X		X		X	X		X		39,06%	13,61%	9,20 GB
X		X	X	X	X		X		39,06%	13,61%	9,19 GB
X			X	X	X		X		39,06%	13,61%	9,19 GB
X		X			X		X		39,06%	13,61%	9,10 GB
X		X	X		X		X		39,06%	13,61%	9,04 GB
X			X		X		X		39,06%	13,61%	8,99 GB
X		X			X	X			39,08%	13,58%	3,39 GB
X		X	X		X	X			39,08%	13,57%	3,39 GB
X			X		X	X			39,08%	13,57%	3,37 GB
X				X	X	X			39,08%	13,57%	3,41 GB
X					X			X	39,09%	13,56%	16,96 GB
X		X		X	X			X	39,10%	13,53%	17,08 GB
X		X			X			X	39,10%	13,52%	17,16 GB
X		X	X		X			X	39,11%	13,52%	16,86 GB
X				X	X			X	39,11%	13,51%	17,30 GB
X			X		X			X	39,11%	13,51%	17,07 GB
X		X		X	X	X			39,11%	13,50%	3,41 GB
X		X	X	X	X			X	39,11%	13,50%	17,31 GB
X			X	X	X	X			39,11%	13,50%	3,41 GB
X		X	X	X	X	X			39,11%	13,50%	3,41 GB

Tabela A.2: Taxa de erro de cada combinação de conjunto de treinamento avaliado na partição de teste do CORAA.

CETENFolha	CORAA	CV 6.1	CV 8.0	MLS	WikiText	3-gram	4-gram	5-gram	WER	Melhoria no WER	Tamanho do ML
X			X	X	X			X	39,12%	13,50%	17,19 GB
				X	X		X		39,32%	13,05%	8,19 GB
		X	X		X	X			39,32%	13,03%	3,07 GB
			X		X	X			39,32%	13,03%	3,05 GB
		X			X	X			39,33%	13,02%	3,07 GB
		X	X	X	X	X			39,33%	13,02%	3,09 GB
			X	X	X	X			39,33%	13,02%	3,09 GB
					X	X			39,33%	13,02%	3,05 GB
				X	X	X			39,33%	13,01%	3,10 GB
		X	X	X	X		X		39,34%	13,00%	8,26 GB
		X		X	X	X			39,34%	13,00%	3,09 GB
			X	X	X		X		39,34%	13,00%	8,26 GB
		X		X	X		X		39,34%	13,00%	8,26 GB
					X		X		39,35%	12,98%	8,17 GB
				X	X			X	39,35%	12,97%	15,42 GB
					X			X	39,37%	12,94%	15,28 GB
			X		X			X	39,37%	12,93%	15,27 GB
		X	X		X			X	39,37%	12,93%	15,20 GB
		X		X	X			X	39,37%	12,93%	15,49 GB
		X	X		X		X		39,37%	12,93%	8,19 GB
		X			X			X	39,37%	12,93%	15,37 GB
			X		X		X		39,37%	12,93%	8,16 GB
		X			X		X		39,37%	12,92%	8,19 GB
		X	X	X	X			X	39,38%	12,92%	15,44 GB
			X	X	X			X	39,38%	12,92%	15,37 GB
		X	X	X			X		40,13%	11,24%	117,73 MB

Tabela A.2: Taxa de erro de cada combinação de conjunto de treinamento avaliado na partição de teste do CORAA.

CETENFolha	**CORAA**	**CV 6.1**	**CV 8.0**	**MLS**	**WikiText**	**3-gram**	**4-gram**	**5-gram**	**WER**	**Melhoria no WER**	**Tamanho do ML**
		X	X	X		X			40,13%	11,24%	59,23 MB
			X	X		X			40,14%	11,23%	58,56 MB
		X	X	X				X	40,15%	11,21%	185,34 MB
			X	X			X		40,15%	11,20%	117,49 MB
			X	X				X	40,16%	11,18%	183,72 MB
		X		X				X	40,32%	10,84%	176,46 MB
		X		X		X			40,32%	10,83%	56,96 MB
		X		X			X		40,33%	10,81%	113,37 MB
				X				X	40,70%	9,98%	173,00 MB
				X			X		40,71%	9,97%	109,35 MB
				X		X			40,73%	9,92%	54,14 MB
		X	X			X			41,20%	8,88%	5,49 MB
			X			X			41,22%	8,84%	5,27 MB
		X	X				X		41,22%	8,83%	9,43 MB
		X	X					X	41,24%	8,80%	13,39 MB
			X				X		41,26%	8,75%	9,17 MB
			X					X	41,27%	8,74%	13,04 MB
		X				X			42,20%	6,67%	2,51 MB
		X					X		42,24%	6,58%	4,35 MB
		X						X	42,24%	6,58%	6,19 MB