Fine-tuning do Whisper para Reconhecimento de Fala em Português-BR

1st Rodrigo C. Fardin
Programa de Pós-Graduação em Informática (PPGI)
Universidade Federal do Espírito Santo (UFES)
Vitória, ES, Brazil
rodrigo.correa.fardin@gmail.com

2nd Agnelo P. L. Júnior
Programa de Pós-Graduação em Informática (PPGI)
Universidade Federal do Espírito Santo (UFES)
Vitória, ES, Brazil
agnjuniorlima@gmail.com

3nd Luiz R. A. de Araujo
Programa de Pós-Graduação em Informática (PPGI)
Universidade Federal do Espírito Santo (UFES)
Vitória, ES, Brazil
luiz.r.araujo@edu.ufes.br

***Abstract*—Os avanços nos sistemas de Reconhecimento Automático de Fala (ASR) têm impulsionado o desenvolvimento de modelos mais robustos, como o Whisper, da OpenAI. Este trabalho investiga o fine-tuning do modelo Whisper-small para o reconhecimento de fala em português brasileiro, utilizando o framework Hugging Face Transformers. O treinamento foi realizado com um conjunto de dados contendo 31.432 amostras para treinamento e 9.467 para teste. Os experimentos indicaram que o modelo alcançou um *Word Error Rate* (WER) de 17.3%, demonstrando um desempenho promissor na transcrição de áudio em português.**

***Index Terms*—Reconhecimento de Fala, Whisper, Fine-tuning, Modelos de Transformadores, Português-BR**

I. Introdução

Os sistemas de Reconhecimento Automático de Fala (ASR) têm apresentado avanços significativos nos últimos anos, tornando-se essenciais em diversas tecnologias modernas, como assistentes por voz, serviços de transcrição e comunicação em tempo real. Historicamente, o desenvolvimento desses sistemas concentrou-se em idiomas com abundância de recursos – como o inglês – que se beneficiam de extensos conjuntos de dados rotulados e de modelos linguísticos sofisticados. Em contraste, sistemas pré-treinados de forma supervisionada utilizando diversos conjuntos de dados ou domínios demonstram maior robustez e capacidade de generalização em comparação com modelos treinados em uma única fonte [1]–[3].

Um exemplo dessa evolução é o Whisper, da OpenAI [4], que foi concebido com base em transformadores para interpretar uma ampla variedade de idiomas, sotaques e ambientes ruidosos com elevada precisão. Esse modelo utilizou aproximadamente 680.000 horas de dados rotulados, abrangendo mais de 96 línguas, o que ilustra o potencial de combinar grandes conjuntos de dados de alta qualidade para melhorar o desempenho em tarefas de reconhecimento de fala. Entretanto, apesar desse amplo suporte linguístico, desafios persistem no que tange ao desempenho em idiomas com poucos recursos. Em tais casos, torna-se imprescindível realizar um *fine-tuning* (ajuste fino) nesses modelos para aprimorar sua eficácia em contextos práticos, visto que incorporam variações em sotaques, pronúncia, ambiente acústico, gênero e idade, principalmente quando se tratam de segmentos de áudio longos [5]–[7].

Dessa forma, a literatura destaca a importância de combinar grandes volumes de dados com estratégias de ajuste fino para aprimorar a eficácia dos sistemas ASR, especialmente em cenários que envolvem idiomas de baixa disponibilidade de recursos [1], [5], [6], [8].

Este estudo aproveita o modelo Whisper, que é capaz de capturar dependências de longo alcance na fala e lidar com complexidades linguísticas de forma mais eficiente. Embora o Whisper apresente um bom desempenho para o português em geral, este trabalho concentra-se em realizar um ajuste fino específico para o português do Brasil. Essa abordagem visa adaptar o modelo às particularidades fonéticas, lexicais e culturais do português brasileiro, aprimorando sua capacidade de capturar variações regionais e contextos específicos que podem não ter sido completamente contemplados no treinamento original.

A. Trabalhos Correlatos

Estudos recentes têm demonstrado a eficácia de modelos baseados em arquiteturas Transformer para ASR, com destaque para o Whisper, devido à sua capacidade de generalização em múltiplos idiomas. Para avaliar a eficiência do ajuste fino de modelos pré-treinados em contextos linguísticos específicos, diversas abordagens comparativas têm sido desenvolvidas, investigando o impacto da adaptação desses modelos a diferentes cenários.

Oyucu [9] investigou o ajuste fino do modelo Whisper Small para o idioma turco, demonstrando que o modelo ajustado obteve um desempenho superior, alcançando uma taxa de erro de palavras (WER) de 17,98% em um vocabulário amplo e 12,89% ao reconhecer palavras e expressões de diferentes domínios temáticos. Os resultados evidenciaram as vantagens

do Whisper Small em relação aos sistemas ASR turcos existentes, apresentando uma redução significativa na taxa de erro.

Polat et al. [10] também analisaram o impacto do ajuste fino da arquitetura Whisper para o idioma turco, utilizando a técnica Low-Rank Adaptation (LoRA) para otimizar o desempenho do modelo. O estudo avaliou o desempenho inicial do Whisper em cinco conjuntos de dados de fala em turco, onde as taxas de erro de palavras (WER) variaram de 4,3% a 14,2%. Após a aplicação da LoRA, observou-se uma redução na WER de até 52,38%, evidenciando a eficácia da adaptação do modelo Whisper para o turco, um idioma de recursos limitados, e demonstrando a viabilidade de modelos ajustados para a construção de sistemas ASR mais robustos e precisos.

Rijal et al. [8] exploraram a adaptação do Whisper para línguas sub-representadas, como o nepali, demonstrando melhorias significativas na taxa de erro de palavras (WER). Utilizando um conjunto de dados composto por diversas variações de sotaques, dialetos e estilos de fala, o estudo revelou reduções na WER de até 36,2% para o modelo Whisper Small e 23,8% para o modelo Whisper Medium em comparação com os modelos *baselines* do Whisper.

Estudos voltados para línguas indianas também destacaram a eficácia do ajuste fino do Whisper. Bhogale et al. [11] propuseram o modelo *IndicWhisper*, que ajustou o Whisper para 12 línguas indianas utilizando um conjunto de dados de 10,7K horas. O *IndicWhisper* obteve uma redução média de 4,1% na WER, melhorando o desempenho em 39 dos 59 benchmarks avaliados. Esses resultados enfatizam a importância da adaptação de modelos ASR a idiomas específicos, especialmente quando combinada com grandes volumes de dados diversos.

Labied, Belangour e Banane [12] exploraram a adaptação do Whisper para a tradução do dialeto Darija para o Árabe Padrão Moderno (MSA). O estudo realizou um ajuste fino da versão pequena do Whisper no corpus Darija-C, utilizando 5000 passos de treinamento e avaliando o desempenho com a métrica BLEU. O modelo ajustado obteve uma pontuação BLEU de 0,65, representando uma melhoria significativa em relação aos modelos *baselines*.

Esses estudos destacam que, independentemente do idioma, a qualidade dos dados e a metodologia de ajuste fino são fatores determinantes para o sucesso dos sistemas ASR, reforçando a importância da adaptação personalizada para cada contexto linguístico.

II. Metodologia

Esta seção descreve de forma detalhada os procedimentos adotados para o ajuste fino do modelo Whisper (versão pequena) [4] em tarefas de transcrição de fala para o idioma português, abrangendo desde o pré-processamento dos dados até a configuração do treinamento e avaliação.

A. Conjunto de Dados

Para os experimentos, utilizou-se o conjunto de dados *common-voice-17-0* [13], o qual foi carregado e segmentado em dois subconjuntos: o conjunto de treinamento, composto por 31.432 amostras, e o conjunto de teste, com 9.467 amostras.

B. Pré-processamento dos Dados

Para preparar os dados de entrada, foi adotado um fluxo que envolve a extração de características do áudio e a codificação dos rótulos:

1) Extração de Características: Utilizou-se o *WhisperFeatureExtractor* a partir de um modelo pré-treinado (`whisper-small`). Esse extrator realiza duas operações essenciais: primeiramente, procede com o preenchimento ou truncamento dos áudios, de modo que aqueles com duração inferior a 30 segundos são completados com silêncio (zeros) até atingir esse intervalo, enquanto os áudios com duração superior a 30 segundos são truncados; em seguida, converte os sinais de áudio em espectrogramas log-Mel, que são representações visuais do espectro e constituem a forma de entrada esperada pelo modelo.

2) Codificação dos Rótulos: Para a geração das transcrições, o *WhisperTokenizer* foi utilizado para mapear os *ids* de tokens para as respectivas strings de texto. No caso do português, o tokenizador é carregado com os parâmetros de idioma e tarefa (*transcribe*), garantindo a inclusão dos tokens específicos de idioma no início das sequências.

3) Integração dos Processos: Para simplificar a manipulação dos dados, os componentes de extração e tokenização foram integrados na classe *WhisperProcessor*, que centraliza as operações de pré-processamento necessárias tanto para os dados de áudio quanto para as transcrições.

4) Reamostragem dos Áudios: Os áudios originais, amostrados a 48kHz, foram reamostrados para 16kHz, que é a taxa de amostragem esperada pelo modelo Whisper. Esse processo é realizado de forma dinâmica, ajustando os dados durante o carregamento.

C. Configuração do Modelo

Neste trabalho, utilizou-se o modelo pré-treinado `whisper-small`, que conta com 12 camadas, largura de 768, 12 cabeças de atenção e aproximadamente 244 milhões de parâmetros [4]. Para garantir que a transcrição fosse realizada em português, a configuração de geração foi ajustada, definindo explicitamente os argumentos de idioma e tarefa e desabilitando a detecção automática do idioma.

D. Pipeline de Treinamento e Preparação dos Dados para o Modelo

A preparação dos dados para o treinamento envolve o uso de um *data collator* específico para modelos de fala sequência-para-sequência, que trata os *input_features* e os *labels* de forma diferenciada. Os *input_features*, que já foram convertidos para espectrogramas log-Mel de dimensão fixa, são agrupados em tensores PyTorch. Por outro lado, as sequências tokenizadas dos *labels* são preenchidas até o comprimento máximo do lote, sendo que os tokens de preenchimento são substituídos por `-100` para que não influenciem o cálculo da perda, e o token de início de sequência (BOS) é removido,

pois será reinserido posteriormente durante o treinamento. A implementação deste *data collator* faz uso do *WhisperProcessor* para realizar as operações de preenchimento tanto para os *input_features* quanto para os *labels*.

A implementação deste *data collator* faz uso do *WhisperProcessor* para realizar as operações de preenchimento tanto para os *input_features* quanto para os *labels*.

E. Métricas de Avaliação.

Para avaliar a performance do sistema ASR, empregou-se a métrica *Word Error Rate (WER)*, amplamente reconhecida como padrão para sistemas de reconhecimento automático de fala. O WER mede a taxa de erro na transcrição de um áudio, sendo calculado pela soma de palavras inseridas, deletadas e substituídas, dividida pelo número total de palavras da transcrição de referência. Quanto menor o WER, melhor a qualidade do modelo. Essa métrica foi carregada por meio do pacote *evaluate* e integrada a uma função `compute_metrics`, a qual processa as previsões do modelo, decodifica os *ids* para suas respectivas strings e realiza a comparação com os rótulos de referência, permitindo uma análise precisa da acurácia do sistema.

F. Configuração do Treinamento

Os parâmetros de treinamento foram definidos conforme a seguinte configuração:

- Batch size por dispositivo: 16;
- Gradient Accumulation: 2;
- Taxa de Aprendizado: 1e-5;
- Warmup Steps: 1000;
- Épocas de Treinamento: 2 (aproximadamente 2000 passos);
- Gradient Checkpointing e FP16: Habilitados para otimização do uso de memória;
- Estratégia de Avaliação: Execução de avaliações a cada 1000 steps, utilizando WER como métrica principal.

G. Especificações do Ambiente de Treinamento

O treinamento foi conduzido em uma máquina equipada com CPU Intel® i7-10700, 16 GiB de RAM e uma GPU NVIDIA GeForce RTX 2060 SUPER (8 GiB).

III. Experimentos

O treinamento foi conduzido utilizando o modelo pré-treinado `whisper-small`, com o objetivo de avaliar o desempenho do sistema de reconhecimento de fala em português brasileiro, e foi executado em uma máquina equipada com uma GPU de 8 GiB e 16 GiB de RAM, com Python 3.12.2, utilizando o framework Hugging Face Transformers. O conjunto de dados foi segmentado em dois grupos: treinamento (31.432 amostras) e teste (9.467 amostras). O treinamento foi realizado por um total de 2 épocas, com um *batch size* de 16 e *gradient accumulation* de 2 para simular o batch maior. O *batch size* e *gradient accumulation* foram ajustados para a memória disponível na máquina, de forma que não houvesse excesso de uso de memória durante o treinamento, permitindo uma execução mais eficiente sem comprometer o desempenho do modelo.

Por se tratar de um modelo com um grande número de parâmetros a serem ajustados, observou-se uma limitação na capacidade computacional, o que resultou em uma taxa de progresso relativamente baixa. A taxa de amostragem foi de 0.273 amostras por segundo, enquanto a taxa de passos foi de 0.009 passos por segundo, o que, consequentemente, prolongou significativamente o tempo necessário para completar uma época.

O treinamento foi interrompido em algumas ocasiões devido a quedas de energia inesperadas no laboratório, o que comprometeu a continuidade de alguns experimentos e pode ter afetado o desempenho final. A falta de energia levou a perdas de progresso em certos pontos do treinamento, o que resultou em tempos de treinamento mais longos do que o inicialmente previsto. Essas quedas de energia representaram uma limitação importante nos experimentos, impedindo que algumas execuções fossem completadas de forma ideal.

No entanto, foi observada uma boa convergência da *loss* do modelo nos dados de treinamento, que diminuiu de 0.6052 para 0.1831 ao longo das épocas, indicando uma melhoria no desempenho do modelo.

IV. Resultados

O modelo foi avaliado usando a métrica *Word Error Rate* (WER), que mede a precisão dos sistemas de reconhecimento de fala. Durante a primeira época, o modelo obteve um *Training Loss* de 0.1831 e um *Validation Loss* de 0.2593, com o *WER* alcançando 17.38%. Após 1964 passos de treinamento, o *Training Loss* final foi de 0.2389. O tempo total de treinamento foi de aproximadamente 63 horas e 49 minutos, refletindo a complexidade do modelo e as limitações do hardware utilizado.

A avaliação final no conjunto de teste mostrou um desempenho semelhante ao treinamento, com *Validation Loss* de 0.2593 e *WER* de 17.38%, alcançando o melhor *WER* de 17.3%. O tempo de avaliação foi de cerca de 5 horas e 59 minutos, com uma taxa de amostragem de 0.517 amostras por segundo. Apesar dos resultados positivos, quedas de energia e limitações de hardware impactaram o desempenho e a rapidez do processo, sugerindo que melhorias nesses aspectos poderiam acelerar o treinamento e possivelmente melhorar o desempenho do modelo.

Os experimentos realizados demonstraram que o modelo atingiu um bom desempenho na transcrição de áudio em português, com um *WER* de 17.3%, evidenciando sua capacidade de reconhecer e transcrever a fala com precisão razoável. No entanto, ainda há margem para melhorias, especialmente considerando que limitações de hardware e interrupções no treinamento afetaram tanto a duração do processo quanto a eficiência do modelo. Com um ambiente computacional mais robusto e condições mais estáveis, é possível reduzir o tempo de treinamento e potencialmente aprimorar a qualidade das transcrições. Além disso, testes adicionais com diferentes

configurações de hiperparâmetros e estratégias de ajuste fino podem contribuir para um desempenho ainda melhor.

REFERENCES

[1] H. Polat *et al.*, "Implementation of a Whisper Architecture-Based Turkish Automatic Speech Recognition (ASR) System and Evaluation of the Effect of *fine-tuning* with a Low-Rank Adaptation (LoRA) Adapter on Its Performance," *Electronics*, vol. 13, no. 21, p. 4227, 2024.

[2] Y. Wang, J. Li, H. Wang, Y. Qian, C. Wang, e Y. Wu, "Wav2vec-Switch: Contrastive Learning from Original-Noisy Speech Pairs for Robust Speech Recognition," arXiv preprint arXiv:2110.04934, 2022. [Online]. Available: https://arxiv.org/abs/2110.04934

[3] A. Narayanan *et al.*, "Toward domain-invariant speech recognition via large scale training," in *Proc. IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 441–447.

[4] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," arXiv preprint arXiv:2212.04356, 2022. [Online]. Available: https://arxiv.org/abs/2212.04356

[5] A. Bhanushali *et al.*, "Gram Vaani ASR Challenge on spontaneous telephone speech recordings in regional variations of Hindi," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2022, pp. 3548–3552.

[6] A. Parikh, L. Bosch, H. Heuvel, and C. Tejedor-García, "Comparing Modular and End-To-End Approaches in ASR for Well-Resourced and Low-Resourced Languages," in *Proc. 6th Int. Conf. Natural Language and Speech Processing (ICNLSP)*, 2023, pp. 266–273.

[7] W. Hsu *et al.*, "Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training," arXiv preprint arXiv:2104.01027, 2021. [Online]. Available: https://arxiv.org/abs/2104.01027

[8] S. Rijal, S. Adhikari, M. Dahal, M. Awale, and V. Ojha, "Whisper Finetuning on Nepali Language," arXiv:2411.12587, 2024. [Online]. Available: https://arxiv.org/abs/2411.12587

[9] S. Oyucu, "Comparing the *fine-tuning* and Performance of Whisper Pre-Trained Models for Turkish Speech Recognition Task," in *Proc. 7th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Ankara, Turkiye, 2023, pp. 1–4, doi: 10.1109/ISMSIT58785.2023.10304891.

[10] Polat, H., Turan, A. K., Koçak, C., & Ulaş, H. B. (2024). Implementation of a Whisper Architecture-Based Turkish Automatic Speech Recognition (ASR) System and Evaluation of the Effect of *fine-tuning* with a Low-Rank Adaptation (LoRA) Adapter on Its Performance. *Electronics, 13*(21), 4227. https://doi.org/10.3390/electronics13214227

[11] K. S. Bhogale, S. Sundaresan, A. Raman, T. Javed, M. M. Khapra, and P. Kumar, "Vistaar: Diverse Benchmarks and Training Sets for Indian Language ASR," arXiv:2305.15386, 2023. [Online]. Available: https://arxiv.org/abs/2305.15386

[12] Labied, M., Belangour, A., & Banane, M. (2024). Fine-Tuning Whisper for Speech Translation: A Case Study on Translating Darija to Arabic. In *2024 International Conference on Decision Aid Sciences and Applications (DASA)* (pp. 1-6). https://doi.org/10.1109/DASA63652.2024.10836462

[13] Mozilla Common Voice, "Common Voice Dataset Documentation," 2020. [Online]. Available: https://commonvoice.mozilla.org/