Análise de Escalabilidade para Armazenamento e Processamento de Arquivos de Áudio Utilizando Transformers

Heloísa Dias Viotto
hdv23@inf.ufpr.br
Departamento de Informática
Universidade Federal do Paraná
Curitiba, Paraná, Brasil

Cauê Mateus Gonçalves Venturin Samonek
cmgvs23@inf.ufpr.br
Departamento de Informática
Universidade Federal do Paraná
Curitiba, Paraná, Brasil

André Ricardo Abed Grégio
gregio@inf.ufpr.br
Departamento de Informática
Universidade Federal do Paraná
Curitiba, Paraná, Brasil

Fabiano Silva
fabiano@inf.ufpr.br
Departamento de Informática
Universidade Federal do Paraná
Curitiba, Paraná, Brasil

Marcos Sfair Sunye
sunye@inf.ufpr.br
Departamento de Informática
Universidade Federal do Paraná
Curitiba, Paraná, Brasil

Paulo Ricardo Lisboa de Almeida
paulo@inf.ufpr.br
Departamento de Informática
Universidade Federal do Paraná
Curitiba, Paraná, Brasil

ABSTRACT

In a continental-sized country like Brazil, collecting feedback on governmental services such as education, healthcare, and security is challenging and impractical to perform manually, except through sampling techniques. With advancements in machine learning, particularly models based on transformers, it is now possible to automate this process on a large scale, enabling, for instance, the dissemination of health campaign information or the collection of citizen opinions on recently used services. This paper focuses on speech-to-text transcription, a crucial step for enabling large-scale voice-based responses. We explored scalability challenges and evaluated combinations of transcription models and audio formats (WAV, FLAC, and MP3), aiming to balance the computational cost and transcription quality. Our results showed that MP3 files sampled at 14 kHz provide transcription quality comparable to WAV files sampled at 16 kHz while requiring only 11% of the storage size. Furthermore, we demonstrated that smaller models, such as Wav2Vec2-XLSR-53 with 3.17×10^8 parameters, can achieve results similar to larger models, such as Seamless M4T, which has approximately an order of magnitude more parameters.

KEYWORDS

Speech-to-Text, Transformers, Áudio, Deep Learning.

1 INTRODUÇÃO

Em 2021, foi sancionada a lei que lançou a Estratégia Nacional de Governo Digital (ENGD), cujos propósitos incluem o foco no cidadão e sua participação ativa para aprimorar os serviços governamentais [1]. Mais recentemente, em junho de 2024, foi promulgado o decreto que institui a ENGD para o período que compreende os anos de 2024 até 2027, considerando a transformação digital, isto é, atendimento eficiente ao cidadão e proximidade com o mesmo por meio da tecnologia [2].

Porém, para que isso aconteça, é necessário que as tecnologias realmente atendam os cidadãos de forma a superar barreiras econômicas, regionais e sociais, enfrentando os inúmeros desafios que podem surgir na implantação de um sistema que realmente "conversa" com a sociedade. Os desafios dessa comunicação ficam ainda mais evidentes considerando o levantamento do IBGE de 2023 que contabiliza cerca de 55 idosos para cada 100 jovens em um país com expectativa de vida de aproximadamente 76 anos, cujas taxas de escolaridade indicam 9,3 milhões de analfabetos, principalmente entre pessoas com mais de 60 anos [3, 4].

Visto que essas pessoas são as que mais usam e precisam de um serviço público de qualidade, são as mais indicadas para fazer um retorno sobre a situação destes. Sabendo-se do analfabetismo presente nesta parcela da população, a solução é abordar essas pessoas via ligações telefônicas, ligando um certo tempo após a execução de um dado serviço (por exemplo, consulta no posto de saúde) e obtendo sua resposta sobre aspectos do serviço ofertado.

Dadas as dimensões continentais do Brasil e sua massa populacional, é inviável a comunicação humana individual. Logo, faz-se necessária a automatização da geração de áudio e da coleta de respostas dos cidadãos para geração dos indicadores. Os áudios criados para interação e as respostas coletadas, também em áudio, demandam sistemas baseados em Aprendizado de Máquina que possam gerar, processar e armazenar uma massa crescente de arquivos, delineando assim o cenário a ser abordado no presente trabalho.

No escopo de arquivos de áudio, um único minuto de áudio não comprimido amostrado a 44.100 Hz com resolução de 16 bits ocupa aproximadamente 5MB de espaço de armazenamento [5, 6][1]. Apesar da simplicidade em se processar um arquivo como esse, uma vez que não há necessidade de descomprimi-lo na memória, as necessidades de armazenamento podem se tornar proibitivas para implementações em larga escala (por exemplo, alcançar milhões de cidadãos). Para ilustrar a escala, ao se supor 1.000.000 de cidadãos (aproximadamente 0,45% da população brasileira) respondendo perguntas em fala normal provenientes de chamadas telefônicas de um minuto que geram arquivos de áudio com qualidade de CD não comprimido, uma única campanha de coleta de informações pode requerer cerca de 5TB de espaço para armazenamento.

[1] $44,100 \text{ amostras} \times 16 \text{ bits} \times 60 \text{ segundos} = 5,05MB$. Este é o padrão para arquivos de áudio de CD, embora para CDs, o tamanho seja dobrado para áudio estéreo.

Além disso, os arquivos de áudio devem ser armazenados levando-se em consideração a privacidade dos usuários e a Lei Geral de Proteção de Dados Pessoais (LGPD) brasileira, pelo menos até que sejam analisados por um modelo de aprendizado de máquina que transcreva áudio e proveja análise de sentimento geral do público em relação à qualidade dos serviços. Dessa forma, devido à grande quantidade de usuários envolvidos, uma campanha de saúde como a exemplificada acima pode sobrecarregar os servidores em termos de processamento e armazenamento, bem como gerar altos custos financeiros e impacto ambiental devido à emissões de carbono relacionadas ao consumo energético dos sistemas envolvidos.

Um dos objetivos deste trabalho é a análise de escalabilidade, visto que a variação entre compressores de áudios e a taxa de amostragem impactam significativamente no consumo de memória dos arquivos de áudio e na precisão da transcrição dos modelos, os quais também variam em relação ao consumo de memória. Logo, a busca pelo equilíbrio entre a qualidade das transcrições geradas e os recursos consumidos para armazenamento e processamento dos modelos e dos áudios é de extrema importância.

Levando em consideração que modelos de aprendizado de máquina para transcrição de áudio do estado da arte são comumente baseados em *transformers* [7–9], e considerando ainda os desafios de escalabilidade mencionados, e a premissa de que os arquivos de áudio conterão gravações de vozes humanas que serão processadas a fim de extrair informações relevantes para gestores de governo, levantamos as duas questões de pesquisa que norteiam o presente trabalho:

(PP1) Quais algoritmos e propriedades de compressão podem ser usados para armazenar uma quantidade massiva de arquivos de áudio?
(PP2) Quais combinações de algoritmos para compressão de áudio e de modelos de transcrição baseados em transformers implicam em melhor equilíbrio entre a qualidade dos resultados e os recursos consumidos?

A principal contribuição deste trabalho reside na busca por essas respostas, embasada nos experimentos realizados e resultados mostrados e discutidos no contexto dos desafios de viabilidade de implantação de um sistema de comunicação do porte considerado. A execução dos experimentos mostrou que modelos baseados em transformers com mais parâmetros não são necessariamente os que geram os melhores resultados, e que uma redução nas taxas de amostragem de 16 kHz para 14 kHz combinada com compressão MP3 pode reduzir os arquivos para apenas 18% dos seus tamanhos originais com pouco prejuízo aos modelos de transcrição de áudio.

O restante do texto está organizado da seguinte forma: na Seção 2 é apresentada a fundamentação teórica deste trabalho, focando em sinais de áudio e nas definições de tarefas de transcrição utilizando aprendizado de máquina. Na Seção 3 são apresentados os trabalhos relacionados, com foco em modelos atuais que executam a tarefa de Fala para Texto. Na Seção 4 é apresentado o protocolo experimental, incluindo conjuntos de dados, modelos e hardware utilizado nos experimentos. Na Seção 5 são apresentados os resultados dos experimentos, com foco nos resultados gerados utilizando a combinação desses áudios comprimidos e os modelos de transcrição de áudio. Finalmente, na Seção 6 são apresentadas as conclusões, incluindo as respostas para as perguntas de pesquisa, e duas sugestões de combinação de modelos de transcrição de áudio com formatos de arquivos com foco no português brasileiro.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta os principais componentes da fundamentação teórica envolvida no desenvolvimento deste projeto. Na Seção 2.1, é apresentado o conceito de áudio digital e quantização. Já na Seção 2.2 encontra-se a definição e classificação de compressores de áudio. Por fim, a Seção 2.3 apresenta alguns conceitos importantes sobre o reconhecimento de fala para o entendimento geral deste documento.

2.1 Sinal de Áudio Digital

Um áudio digital é a discretização de um áudio analógico, ou seja, uma onda sonora contínua é representada por um conjunto finito de amostras, sendo essas coletadas a cada intervalo de tempo t, como exemplificado na Figura 1a. Porém, não basta somente discretizar os pontos com relação ao tempo, visto que é necessário também quantizar (discretizar) o intervalo de valores possíveis para cada amostra, como exemplificado nas Figuras 1b e 1c [10].

Dessa forma, fica claro que quanto menor o intervalo de tempo t (i.e., maior a frequência), mais próximo o sinal amostrado ficará do sinal real, ao custo de um consumo maior de memória para armazenar as amostras, e de hardware capaz de coletar e reproduzir tais amostras. Uma análise similar pode ser feita para a quantização dos valores amostrados, já que quantidades maiores de bits podem levar à coletas mais próximas dos dados reais, ao custo de maior consumo de memória e processamento.

2.2 Compressão de Áudio com e sem perda

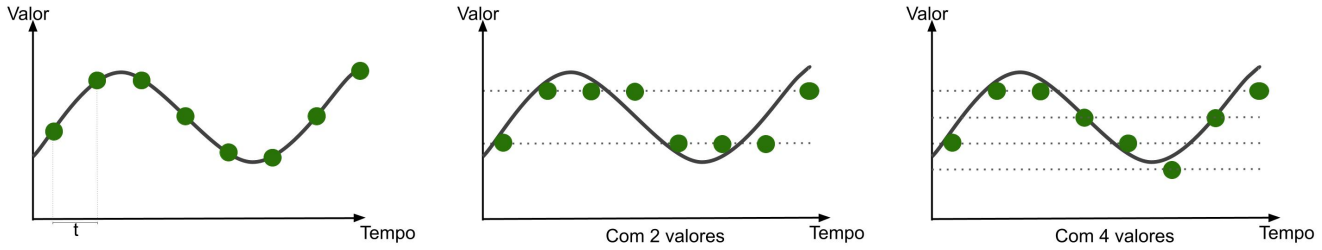
Sendo um dos formatos de áudio mais conhecidos, o WAV (*Waveform Audio File*), desenvolvido em 1991 pela Microsoft e IBM, permite que informações de um áudio sejam armazenadas utilizando os conceitos de amostragem e quantização apresentados na Seção 2.1 (técnica comumente chamada de *Pulse-Code Modulation*), comumente sem compressão, levando a um alto consumo de memória[11]. Neste contexto, torna-se interessante o uso de compressores.

A compressão de um áudio é realizada com o objetivo de reduzir o seu tamanho, proporcionando um armazenamento e transmissão eficientes e podendo impactar a qualidade do arquivo. Devido a isso, os compressores podem ser classificados em *lossy* (com perdas) e *lossless* (sem perdas), dependendo se aceitamos ou não perdas de informação nos arquivos compactados[12].

Compressores *lossy* comumente reduzem o tamanho dos arquivos de forma mais significativa do que compressores *lossless*. No entanto, isso é alcançado através da remoção de algumas informações do arquivo [13]. Como um exemplo de compressor *lossy* amplamente utilizado, pode-se citar o MP3 (MPEG-1/2 Audio Layer 3), desenvolvido por um grupo de pesquisadores do Instituto Fraunhofer na Alemanha[2] que teve algumas de suas patentes expiradas em 2017, tornando o seu uso mais livre e acessível[3].

A compressão *lossless* também proporciona uma redução no consumo de memória, porém mantendo a qualidade do áudio através

[2] https://www.mp3-history.com/en/the_mp3_team.html
[3] https://www.iis.fraunhofer.de/en/ff/amm/consumer-electronics/mp3.html



(a) Pontos amostrados a cada instante t **(b) Quantização de 1 bit** **(c) Quantização de 2 bits**

Figura 1: Amostragem e Quantização

de técnicas como as encontradas em [6]. Um exemplo desse compressor é o FLAC (*Free Lossless Audio Codec*), desenvolvido pela Xiph.org Foundation e com código aberto, o qual utiliza o algoritmo de predição linear para manter a qualidade do áudio armazenado[4].

2.3 Fala para Texto – *Speech-to-Text*

A área do aprendizado de máquina que lida com todas as tarefas relacionadas com a linguística, tanto para traduções como conversões entre textos e falas, é denominada Processamento de Linguagem Natural - *Natural Language Processing* (NLP) [14]. Já a tarefa em que uma fala é transcrita em texto pode ser identificada pelos termos *Speech-to-Text* (STT), *Automatic Speech Recognition* (ASR), ou ainda como *Speech Recognition*, apesar deste também ser utilizado de forma mais geral, referindo-se a qualquer tarefa em que uma fala é reconhecida ou mapeada em um conjunto de palavras [15, 16]. Neste trabalho, utilizaremos o termo *Speech-to-Text* (STT) para nos referir à conversão de fala em texto.

3 TRABALHOS RELACIONADOS

Nesta Seção são apresentados os principais trabalhos relacionados ao trabalho proposto. Vale notar que a área de STT é bastante ampla, e revisões sistemáticas podem ser encontradas em [17–19].

É interessante notar as mudanças nos paradigmas utilizados para a construção de modelos de STT conforme novas técnicas de aprendizado de máquina foram desenvolvidas. Entre o final dos anos de 1980 até o início dos anos 2000, tais sistemas comumente eram modelados utilizando-se *Hidden Markov Models* (HMMs), combinados ou não com Perceptrons Multicamadas – *Multilayer Perceptrons* (MLPs) [19].

Entre o início dos anos 2010 até pouco antes dos anos 2020, as pesquisas em STT apresentavam um grande foco em redes capazes de processar dados sequenciais, como as *Recurrent Neural Networks* (RNNs) e redes do tipo *Long-Short Term Memory* (LSTM) [18, 19]. Apesar de terem maior capacidade de expressar dependências de longo termo quando comparadas aos HMMs, essas redes ainda podem sofrer quando as sequências se tornam muito longas, o que é comum quando realizamos a transcrição de áudio para texto (ex.: uma sequência de áudio pode conter várias horas de gravação) [20].

Com a popularização dos *transformers* para modelagem de dados sequenciais com os artigos de Bahdanau [21] e Vaswani [20], a partir do início dos anos 2020 é notável a prevalência de modelos baseados em transformers para a tarefa de STT devido ao seu alto desempenho nesses trabalhos em relação a outros modelos. Boa parte das redes baseadas em *transformers* atuais são baseadas na arquitetura do *Conformer*, proposta por Gulati et al. [7]. A arquitetura de um *Conformer* combina módulos de múltiplas cabeças de atenção (*Multi-Head Self Attention*), comuns em modelos de *transformers*, com camadas convolucionais, a fim de capturar dependências de curto e longo termo com um custo computacional relativamente baixo. A arquitetura combina ainda MLPs para diminuir a amostragem dos dados antes do processamento. Dentre as redes que possuem arquiteturas baseadas em camadas de *Transformers*, *Conformers*, e suas variantes estão a SeamlessM4T V2 (*Massively Multilingual e Multimodal Machine Translation*) [8], Whisper [22], e Wav2Vec2-XLSR-53 [9], que são foco dos experimentos deste trabalho.

O modelo Whisper, proposto por Radford et al. [22], foca no uso de *transformers* já consolidados no estado da arte para tarefas relacionadas ao STT. A proposta visa mostrar que o uso massivo de dados rotulados auxilia no treinamento de modelos baseados em *transformers*. Os autores coletaram 680,000 horas de áudio rotulado da Internet, utilizando heurísticas para filtrar dados que foram rotulados por outros mecanismos de STT, o que poderia reduzir a qualidade do modelo final treinado.

Já o modelo Seamless M4T V2 (*Massively Multilingual e Multimodal Machine Translation*) é considerado um modelo *foundation*, o qual suporta tarefas de conversão e tradução entre áudios e textos em quase 100 línguas, dependendo da entrada e saída desejada. Em relação a tarefa de conversão de fala em texto, o Seamless M4T V2 aceita 96 línguas, tendo sido treinado em 845 horas de áudios em português [8]. O modelo é formado pela combinação de quatro modelos menores - o Seamless M4T-NLLB, w2v-BERT 2.0, Seamless M4T v2-T2U e Vocoder - responsáveis por pequenas partes da conversão e tradução entre áudios e textos com modelos específicos treinados para codificar e decodificar cada tipo de entrada e saída. Esses modelos menores foram unificados durante o fine-tuning, formando o modelo final [8].

Diferentemente do Whisper e do Seamless M4T V2, o modelo Wav2Vec2-XLSR-53 [9] precisa obrigatoriamente passar por um *fine-tuning* para ser utilizado, visto que ele não possui um *tokenizer*, algoritmo responsável por transformar a entrada em *tokens*, podendo estes ser qualquer tipo de subunidade como palavras e

[4] https://xiph.org/flac/index.html

caracteres. O modelo foi originalmente treinado em 53 línguas, incluindo o português.

Outros modelos notáveis do estado da arte, que no entanto estão fora do escopo deste trabalho, são o Canary [23], e o Parakeet [24, 25]. Na Tabela 1, são exibidas as principais propriedades dos modelos do estado da arte discutidos nesta Seção, incluindo o número de parâmetros (pesos e bias) de cada modelo, a quantidade de memória necessária para carregar os modelos (considerando ponto flutuante IEEE 754 de precisão simples), e suas licenças de uso.

Na Tabela 1, é possível observar que os modelos do estado da arte avaliados variam muito em relação à quantidade de parâmetros, indo de $3,17 \times 10^7$ até $2,3 \times 10^9$, além dos resultados reportados em seus respectivos artigos serem em diversas línguas de acordo com conjunto de dados utilizados em seus experimentos. Dessa forma, se faz necessária uma análise especificamente para a língua portuguesa brasileira, levando em consideração alguns pontos como a influência do compressor de áudio no desempenho das redes e uma avaliação de custo-benefício entre qualidade das transcrições e memória consumida.

4 PROTOCOLO EXPERIMENTAL

Nesta Seção é apresentado o protocolo experimental, que foca na análise de custo-benefício entre a compressão dos arquivos de áudio e a qualidade de suas transcrições geradas pelos modelos, considerando a língua portuguesa. Primeiramente, na Seção 4.1 é apresentado o conjunto de dados que será utilizado para os testes. Na Seção 4.3 são apresentadas brevemente as versões das redes utilizadas nos experimentos. Na Seção 4.4 são apresentadas as métricas que serão utilizadas para avaliar os modelos. Finalmente, na Seção 4.5 é apresentado o hardware que foi utilizado durante os experimentos.

4.1 Conjuntos de Dados utilizados

Muitos conjuntos de dados populares, como o Common Voice [26], foram usados no treinamento ou *fine-tuning* das redes utilizadas nos experimentos. Dessa forma, para evitar possíveis vieses, foi utilizado somente o conjunto de dados CORpus de Áudios Anotados (CORAA) [27] para os experimentos deste artigo, já que esse conjunto não foi utilizado para o treinamento de nenhum dos modelos testados, além de ser um conjunto de dados diverso e fornecer um grande volume de amostras.

Nos experimentos foi empregada a versão 1.1 do CORAA, criado pelo projeto Tarefa de Anotação para o Reconhecimento e Síntese de fala da Língua Portuguesa (TaRsila)[5], do Centro de Inteligência Artificial[6] (C4AI) da Universidade de São Paulo. O CORAA foi criado com o objetivo de fornecer um grande e rico conjunto de áudios anotados em português brasileiro, com cerca de 401 mil arquivos de áudio, totalizando aproximadamente 291 horas, separados em 3 conjuntos distintos: treino, teste e dev. Além de possuir áudios que variam em gênero (monólogo, diálogo, entrevistas, conferência ou palestras) e estilo (fala preparada, espontânea ou lida), as amostras também se diferenciam com relação à clareza, sendo algumas falas em ambientes controlados e outras com ruídos de fundo, variando também com relação ao sotaque dos falantes, sendo a maioria de Minas gerais, São Paulo e Recife [27].

Durante os experimentos, foram utilizados todos os dados dos subconjuntos de teste e treino do CORAA[7], sendo que algumas amostras foram removidas por estarem em português de Portugal. O conjunto final utilizado nos experimentos totaliza 285 horas, com aproximadamente 394 mil arquivos de áudio, com cerca de 62 GB.

O conjunto de arquivos de áudio CRPIH_UVigo-GL-Voices [28] foi utilizado exclusivamente para os experimentos focados no tempo de compressão. Esse conjunto foi escolhido por conter áudios com qualidade de CD, sendo ideal para testar os algoritmos de compressão de áudio por suas características (44.1 kHz e 16-bit PCM (*Pulse Code Modulation*)) as quais garantem que as diferenças sutis do som detectadas pelos humanos sejam registradas e reproduzidas com alta fidelidade [5, 29]. O conjunto de dados utilizado conta com 10.000 arquivos, somando aproximadamente 15 horas de gravação com falantes galegos.

4.2 Compressores Utilizados

Este trabalho analisa os áudios em três formatos: WAV, FLAC e MP3, representando o áudio bruto e os compressores *lossless* e *lossy* (Seção 2.2). Os áudios originais utilizados para os testes estão no formato WAV. Para transformá-los em FLAC e MP3, foi utilizado a biblioteca do Python Pydub, a qual utiliza a implementação do FFmpeg[8] na versão 7.0.1. Para a conversão do formato WAV em FLAC e MP3, foi utilizado apenas o áudio e o formato a ser convertido, sem nenhuma informação adicional e utilizando os parâmetros padrão das bibliotecas.

4.3 Versões das Redes Utilizadas nos Experimentos

São analisadas diferentes versões das arquiteturas de rede Seamless M4T V2, Wav2Vec2-XLSR-53 e Whisper, totalizando 10 modelos testados. As diferentes configurações das redes proporcionam uma análise comparativa entre acurácia e tamanho de cada uma. As redes testadas estão sumarizadas na Tabela 2.

Para o modelo Whisper foram analisadas cinco versões multilinguísticas: Tiny, Base, Small, Medium e Large V3. O modelo Seamless M4T V2 foram analisadas suas versões Medium e Large V2. Já os modelos Wav2Vec2 considerados não variam em número de parâmetros, mas sim em seus treinamentos, sendo eles:

- Facebook: *fine-tuning* utilizando o conjunto LibriSpeech, de 161 horas. O *fine-tuning* foi realizado utilizando a função de perda o CTC loss e taxas de aprendizado fixas [9].
- Jonatasgrosman: encontrado no repositório *HuggingFace* na conta de mesmo nome, este *fine-tuning* foi realizado por Jonatas Grosman no Commom voice 6.1, o qual possui 64 horas[9] [30].
- Lgris: também encontrado no repositório *HuggingFace*, este *fine-tuning* foi realizado por Lucas Gris. Foram utilizados sete conjuntos de dados: CETUC, Commom Voice 7.0, LaPS

[5] https://sites.google.com/view/tarsila-c4ai/
[6] https://c4ai.inova.usp.br/

[7] Os dados de treino foram utilizados para enriquecer os experimentos sem injetar nenhum viés, visto que os modelos analisados neste documento não foram treinados com estes dados.
[8] https://www.ffmpeg.org
[9] https://commonvoice.mozilla.org/pt/datasets

Modelo	Quantidade de Parâmetros	Tamanho (GB)	Ano de Publicação	Licença
Canary	1×10^9	3,7	2024	CC-BY-NC 4.0
Parakeet	$6 \times 10^8 - 1,1 \times 10^9$	2,2 - 4,1	2023	CC-BY-NC 4.0
Seamless M4T V2	$1,2 \times 10^9 - 2,3 \times 10^9$	4,5 - 8,6	2023	CC-BY-NC 4.0
Wav2Vec2-XLSR-53	$3,17 \times 10^7$	0,12	2021	Apache 2.0
Whisper	$3,9 \times 10^7 - 1,55 \times 10^9$	0,2 - 5,8	2020	MIT

Tabela 1: Propriedades do estado da arte. Valores no formato $X - Y$ indicam variações a depender da versão do modelo utilizada.

Modelo	Versão	# Parâmetros	Memória (GB)
Seamless M4T V2 [8]	Medium	$1,2 \times 10^9$	5
	Large V2	$2,3 \times 10^9$	9
Wav2Vec2-XLSR-53 [9, 30, 31]	Facebook Jonatasgrosman Lgris	$3,17 \times 10^8$	1,5
Whisper [22]	Tiny	$3,9 \times 10^7$	0,5
	Base	$7,4 \times 10^7$	0,6
	Small	$2,44 \times 10^8$	1,3
	Medium	$7,69 \times 10^8$	3
	Large V3	$1,55 \times 10^9$	6,5

Tabela 2: Propriedades dos Modelos Testados

Benchmark, LibriSpeech, TEDx, Sidney e VoxForge, totalizando 437,2 horas de áudio em português [31].

4.4 Métricas

No contexto de técnicas de STT, dado determinado áudio, desejamos avaliar a saída produzida pelo modelo em relação à mesma sentença transcrita por um humano, i.e., verificar a diferença entre dois textos. Dentre as possíveis métricas para realizar tal verificação, uma das mais comuns, e que será utilizada neste trabalho, é a Taxa de Erro de Palavras – *Word Error Rate* (WER). A métrica se baseia na distância de Levenshtein, onde é calculado o mínimo de alterações necessárias para transformar um texto em outro com base em três modificações: inserção (I), deleção (D) e substituição (S) relacionadas ao total de palavras (T) no texto de referência, multiplicadas por 100 para se obter uma porcentagem, sendo calculada da seguinte forma [32]:

$$WER = 100 \times \frac{I + D + S}{T}.$$

Como a métrica se refere à taxa de erros, quanto mais próximo de zero, mais próximo do esperado é o texto produzido pelo modelo e, de modo geral, um resultado baixo indica um bom modelo. No entanto, é importante ressaltar que a métrica WER possui certas limitações. Por exemplo, um valor de WER alto não necessariamente significa um modelo ruim, visto que a mudança de um único caractere é o suficiente para acusar uma palavra inteira como errada, tornando-o impreciso em casos onde o modelo tem dificuldade de detectar um fonema específico como, por exemplo, um sotaque.

4.5 Hardware

Para a realização dos experimentos, foi utilizado um nodo disponível em um Computador de Alta Performance (HPC), com oito GPUs Tesla V100-PCIE-32GB. O nodo possui 256GB@1380MHz de memória DRAM e dois processadores Intel Xeon Silver 4110@2.10GHz. Cada modelo foi executado em uma GPU com um *batch* de tamanho 8 devido à grande quantidade de memória consumida pelos modelos.

5 EXPERIMENTOS E RESULTADOS

Inicialmente, na Seção 5.1, são apresentados os experimentos considerando os compressores FLAC e MP3, além de considerar o áudio sem compressão. Já na Seção 5.2, encontram-se os resultados referentes ao WER dos diferentes modelos testados, levando-se em consideração ainda as diferentes compressões experimentadas neste trabalho.

5.1 Compressão do Áudio

Nesta Seção, o subconjunto Sabela do conjunto CRPIH_UVigo-GL-Voices [28], contendo 14,48 horas de gravações com amostragens de 44 kHz e sem compressão, é utilizado para verificar o tempo necessário para compressão, e a quantidade de espaço de armazenamento necessário ao se utilizar cada compressor.

Os resultados estão dispostos na Tabela 3, onde é mostrado o tempo em segundos necessário para comprimir cada hora de áudio, juntamente com o armazenamento necessário (MB) para se armazenar cada hora de áudio, considerando o áudio sem compressão, e os compressores FLAC e MP3. Para as medições de tempo, foi considerado uma única CPU, ou seja, os algoritmos não foram paralelizados.

Amostragem	s/h		MB/h		
	FLAC	MP3	WAV	FLAC	MP3
44 kHz	28,5	61,1	311,2	141,4	29,3
16 kHz	16,0	28,6	113,2	67,2	12,1
14 kHz	15,5	28,3	99,0	60,9	12,1
8 kHz	13,0	17,4	56,4	40,4	5,1

Tabela 3: Tempo de compressão (s) e armazenamento (MB) necessários para cada hora de áudio.

Como pode ser observado na Tabela 3, o compressor FLAC diminui significativamente o tamanho do arquivo de áudio, sua taxa de compressão ($100 \times (1 - \frac{\text{tamanho final}}{\text{tamanho original}})$) varia de acordo com a amostragem utilizada, reduzindo de 54,5% a 44 kHz para 28,4% em 8 kHz. Já o compressor MP3 possui uma taxa em torno de 90% independente da amostragem utilizada. Já quanto ao tempo de compressão, apesar do compressor MP3 sempre demorar mais tempo do que o FLAC, a relação variou de acordo com a amostragem, indo de 2,1 vezes mais lento para 44 kHz, até 1,3 vezes mais lento para 8kHz.

Considerando as redes de fala para texto avaliadas, todas utilizam um áudio de entrada amostrado a 16 kHz. Supondo um exemplo onde 1.000.000 ligações de um minuto são armazenadas, o formato de CD (WAV e 44 kHz) ocuparia 4,8 TB de armazenamento (porém, não seria gasto tempo para compressão). Já o formato FLAC com uma amostragem de 16 kHz, onde não haveriam perdas de informação, ocuparia 1,1 TB, ou seja, apenas 23% do tamanho do dado considerando a qualidade de CD, com um custo de tempo de compressão de cerca de 74 horas. Apesar de parecer proibitivo, o tempo de compressão pode ser facilmente reduzido através de técnicas de paralelização.

5.2 Fala para Texto

Nesta Seção são apresentados os resultados considerando os modelos de STT. Na Tabela 4 é possível observar os valores de WER considerando os compressores FLAC e MP3, para os diferentes modelos de STT testados[10]. Os três melhores valores apresentados encontram-se em negrito.

Taxa de Amostragem	Modelo	Versão	FLAC	MP3
16 kHz	Seamless M4T V2	Medium	56,5	68,0
		Large V2	45,4	61,6
	Wav2Vec2-XLSR-53	Facebook	63,6	65,5
		Jonatasgrosman	57,2	59,1
		Lgris	48,1	49,7
	Whisper	Tiny	201,1	213,7
		Base	172,6	179,6
		Small	101,1	108,2
		Medium	43,1	46,2
		Large V3	**28,7**	30,1
14 kHz	Seamless M4T V2	Medium	57,1	68,2
		Large V2	46,5	61,7
	Wav2Vec2-XLSR-53	Facebook	63,9	65,5
		Jonatasgrosman	57,29	59,04
		Lgris	48,27	49,69
	Whisper	Tiny	202,0	212,7
		Base	171,0	178,4
		Small	101,0	108,0
		Medium	42,8	46,3
		Large V3	**28,9**	**30,0**
8 kHz	Seamless M4T V2	Medium	66,1	107,4
		Large V2	65,8	110,8
	Wav2Vec2-XLSR-53	Facebook	67,7	80,1
		Jonatasgrosman	60,9	75,6
		Lgris	51,4	65,3
	Whisper	Tiny	232,2	449,6
		Base	203,2	392,6
		Small	117,4	227,9
		Medium	48,6	80,8
		Large V3	31,4	50,3

Tabela 4: WER (%) FLAC e MP3

Realizando uma análise com foco nas amostragens, os modelos tiveram melhor desempenho, de forma geral, com áudios de 16kHz, confirmando a relação direta entre os valores de amostragem e a das transcrições. No entanto, a diferença entre os valores obtidos com a amostragem de 16 kHz e com 14kHz é inferior a 2,5% (a maior diferença é entre Seamless M4T, considerando o formato FLAC, que resultou em um acréscimo de WER de 45,4 para 46,5). Alguns modelos, como Whisper Medium, surpreendentemente tiveram melhores valores de WER com áudios de 14kHz. Isso pode ter acontecido devido ao processo de *downsampling*, onde alguma informação presente em 16kHz que comprometia o desempenho do modelo foi retirada.

Com exceção das versões do modelo Seamless M4T, comprimir o áudio utilizando o compressor MP3 resultou em aumentos relativamente pequenos nos valores de WER para as amostragens de 14 e 16 kHz, indicando que pode ser vantajoso armazenar os dados utilizando o formato MP3 em cenários onde a memória de armazenamento é um fator limitante. O aumento no valor de WER para as amostragens de 14 e 16 kHz para o formato MP3, quando comparado ao formato FLAC, foi de 4,3% e 4,6% em média, respectivamente (desconsiderando o modelo Seamless M4T). Dados os resultados da Seção 5.1, conclui-se que pode ser vantajoso optar pela amostragem de 14 kHz com compressão MP3, dado que haverá apenas um pequeno aumento no valor de WER, mas gerando arquivos que ocupam apenas 18% do espaço de armazenamento quando comparados aos arquivos FLAC de 16 kHz.

Finalmente, considerando os modelos específicos testados, o modelo Whisper Large V3, de $1,55 \times 10^9$ parâmetros, destaca-se por ter o melhor resultado com áudios em formato FLAC e uma amostragem de 16kHz, obtendo um WER de 28,7. As versões menores do modelo Whisper aumentam muito os valores de WER, sendo perceptível a piora dos valores conforme o tamanho do modelo diminui, chegando a um WER de 201,1 para a versão Tiny de $3,9 \times 10^7$ parâmetros, considerando a amostragem de 16 kHz. É interessante notar que, como observado anteriormente quanto às amostragens, considerando o modelo Whisper Large V3, há apenas um pequeno aumento no valor de WER utilizando áudio amostrado a 14 kHz, utilizando tanto o compressor FLAC (WER de 28,9) quanto o MP3 (WER de 30,0).

Apesar de apresentar resultados consideravelmente inferiores ao Whisper, com WER próximos a 50 para todas as amostragens, a versão Lgris do modelo Wav2Vec2-XLSR-53 pode ser um bom custo-benefício em termos de processamento, já que o modelo possui apenas $3,17 \times 10^8$ parâmetros, sendo cerca de 5 vezes menor que o Whisper Large V3, o que pode reduzir custos com hardware, consumo de energia e emissões.

Por fim, é interessante notar que apesar de ser o maior modelo em número de parâmetros, o Seamless M4T Large V2 foi capaz de gerar resultados comparáveis apenas a modelos menores, como a versão Lgris do modelo Wav2Vec2-XLSR-53. Esse resultado indica que apenas considerar modelos grandes, com muitos parâmetros, não necessariamente traz benefícios.

6 CONCLUSÃO

Visando um balanço entre precisão de modelos *Speech-to-Text* (STT) e armazenamento do dados necessários, neste artigo apresentamos uma comparação entre diferentes formatos de áudio: WAV, FLAC e MP3, sendo respectivamente, o áudio bruto, comprimido sem perda e comprimido com perda, analisando informações da própria

[10] Os resultados utilizando o formato WAV (sem compressão) foram omitidos por serem equivalentes ao FLAC.

compressão, como o tempo levado para executar seus algoritmos e o tamanho final ocupado pelos arquivos de áudio, e os efeitos que esses formatos possuem em diferentes redes de STT.

Levando em conta os experimentos realizados neste trabalho, chegamos às seguintes respostas para nossas perguntas de pesquisa:

PP1 – Quais algoritmos e propriedades de compressão podem ser usados para armazenar uma quantidade massiva de arquivos de áudio? R.: Quando a qualidade das transcrições é o fator mais crítico a ser considerado, concluímos que armazenar os áudios utilizando compressão FLAC e amostragem de 16 kHz é o recomendado. No entanto, se pequenos aumentos nas taxas de WER são aceitáveis (cerca de 5% de aumento), armazenar os arquivos no formato MP3 e com amostragem de 14 kHz pode ser vantajoso por reduzir significativamente o tamanho dos arquivos, ocupando apenas 18% do espaço de armazenamento quando comparado ao FLAC de 16 kHz. Amostragens maiores que 16 kHz não são indicadas considerando os modelos testados, já que todos requerem uma amostragem de entrada de 16 kHz (ou seja, amostragens maiores que 16 kHz são reduzidas para 16 kHz pelos modelos).

PP2 - Quais combinações de algoritmos para compressão de áudio e de modelos de transcrição baseados em transformers implicam em melhor equilíbrio entre a qualidade dos resultados e os recursos consumidos? R.: O modelo Whisper em sua versão Large V3 apresentou os melhores resultados para ambos os formatos de arquivo discutidos na PP1 (FLAC de 16 kHz ou MP3 de 14 kHz). No entanto, esse modelo possui $1,55 \times 10^9$ parâmetros. Em sistemas onde é necessário um melhor equilíbrio entre qualidade dos resultados e custo computacional, a versão Lgris do modelo Wav2Vec2-XLSR-53 pode ser um melhor custo-benefício, apesar do aumento considerável nos valores de WER.

Dessa forma, levando em consideração os formatos de arquivos e modelos de STT testados neste documento, e as respostas para ambas perguntas de pesquisa, apresentamos duas configurações de sistemas STT para o português brasileiro:

Configuração 1, com foco na qualidade da transcrição: armazenamento dos arquivos utilizando o formato FLAC e amostragem de 16 kHz, utilizando o modelo Whisper Large V3 para transcrição.

Configuração 2, com foco no equilíbrio entre qualidade da transcrição e custo computacional: armazenamento dos arquivos utilizando o formato MP3 e amostragem de 14 kHz, utilizando o modelo Wav2Vec2-XLSR-53 Lgris para transcrição.

Este trabalho abre oportunidades para estudos futuros, como o ajuste fino (*fine-tuning*) específicos para áudios de diferentes amostragens e compressores, com foco no custo-benefício entre qualidade da transcrição e custo computacional, e análise da qualidade da transcrição levando em consideração áudios enviados em diferentes modais (e.g., ligações de voz, áudios enviados por aplicativo de mensagem, áudio gravado durante entrevista, …).

ACKNOWLEDGMENTS

Este projeto foi financiado pelo Ministério da Saúde através de uma TED para PD&I entre SAPS/MS e C3SL/UFPR.

REFERÊNCIAS

[1] Ministério da Gestão e da Inovação em Serviços Públicos. Estratégia nacional de governo digital. www.gov.br/governodigital/pt-br/estrategias-e-governanca-digital/estrategianacional/estrategia-nacional-de-governo-digital, 2021. [Online; acessado 06/Dez/2024].

[2] Presidência da República. Decreto no 12.069, de 21 de junho de 2024. www.planalto.gov.br/ccivil_03/_ato2023-2026/2024/decreto/D12069.htm, 2024. [Online; acessado 06/Dez/2024].

[3] Instituto Brasileiro de Geografia e Estatística. Em 2023, expectativa de vida chega aos 76,4 anos e supera patamar pré-pandemia. agenciadenoticias.ibge.gov.br/agencia-noticias/2012-agencia-de-noticias/noticias/41984-em-2023-expectativa-de-vida-chega-aos-76-4-anos-e-supera-patamar-pre-pandemia, 2024. [Online; acessado 06/Dez/2024].

[4] Instituto Brasileiro de Geografia e Estatística. Pesquisa nacional por amostra de domicílios contínua – educação 2023. https://agenciadenoticias.ibge.gov.br/media/com_mediaibge/arquivos/baf49b4ab43ec70bcba5f01d7f512ffd.pdf, 2024. [Online; acessado 06/Dez/2024].

[5] T. Painter and A. Spanias. Perceptual coding of digital audio. *Proceedings of the IEEE*, 88(4):451–515, 2000. doi: 10.1109/5.842996.

[6] Mat Hans and Ronald W Schafer. Lossless compression of digital audio. *IEEE Signal processing magazine*, 18(4):21–32, 2001.

[7] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

[8] Seamless Communication, Loïc Barrault, Yu-An Chungand Mariano Coria Meglioli, David Dale, Ning Dong, Mark Duppenthaler, Paul-Ambroise Duquenne, Brian Ellis, Hady Elsahar, Justin Haaheim, John Hoffman, Min-Jae Hwang, Hirofumi Inaguma, Christopher Klaiber, Ilia Kulikov, Pengwei Li, Daniel Licht, Jean Maillard, Ruslan Mavlyutov, Alice Rakotoarison, Kaushik Ram Sadagopan, Abinesh Ramakrishnan, Tuan Tran, Guillaume Wenzek, Yilin Yang, Ethan Ye, Ivan Evtimov, Pierre Fernandez, Cynthia Gao, Prangthip Hansanti, Elahe Kalbassi, Amanda Kallet, Artyom Kozhevnikov, Gabriel Mejia, Robin San Roman, Christophe Touret, Corinne Wong, Carleigh Wood, Bokai Yu, Pierre Andrews, Can Balioglu, Peng-Jen Chen, Marta R. Costa-jussà, Maha Elbayad, Hongyu Gong, Francisco Guzmán, Kevin Heffernan, Somya Jain, Justine Kao, Ann Lee, Xutai Ma, Alex Mourachko, Benjamin Peloquin, Juan Pino, Sravya Popuri, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, Anna Sun, Paden Tomasello, Changhan Wang, Jeff Wang, Skyler Wang, and Mary Williamson. Seamless: Multilingual expressive and streaming speech translation. *ArXiv*, 2023.

[9] Alexis Conneau, Alexei Baevski, Ronan Collober, Abdelrahman Mohamed, and Michael Auli. Unsepervised cross-lingual representation learning for speech recognition. *ArXiv*, 2020.

[10] T.L. Floyd. *Sistemas Digitais: Fundamentos e Aplicações*. Bookman, 2007. ISBN 9788560031931.

[11] Mikhail V. Belodedov, Roman V. Fonkants, and Rustam R. Safin. Development of an algorithm for optimal encoding of wav files using genetic algorithms. In *2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE)*, volume 5, pages 1–6, 2023. doi: 10.1109/REEPE57272.2023.10086837.

[12] Bongjun Kim and Zafar Rafii. Lossy audio compression identification. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2459–2463, 2018. doi: 10.23919/EUSIPCO.2018.8553611.

[13] Karlheinz Brandenburg. MP3 and AAC explained. In *Audio Engineering Society Conference: 17th International Conference: High-Quality Audio Coding*. Audio Engineering Society, September 1999.

[14] Sushant Singh and Ausif Mahmood. The nlp cookbook: Modern recipes for transformer based deep learning architectures. *IEEE Access*, 9:68675–68702, 2021. doi: 10.1109/ACCESS.2021.3077350.

[15] Akshi Kumar, Sukriti Verma, and Himanshu Mangla. A survey of deep learning techniques in speech recognition. In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 179–185, 2018. doi: 10.1109/ICACCCN.2018.8748399.

[16] Pedram Aliniaye Asli and Anna Zumbansen. Performance of speech recognition algorithms in musical speech used for speech-language pathology rehabilitation. In *2023 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, pages 1–5, 2023. doi: 10.1109/MeMeA57477.2023.10171898.

[17] Sadeen Alharbi, Muna Alrazgan, Alanoud Alrashed, Turkiayh Alnomasi, Raghad Almojel, Rimah Alharbi, Saja Alharbi, Sahar Alturki, Fatimah Alshehri, and Maha Almojil. Automatic speech recognition: Systematic literature review. *Ieee Access*, 9:131858–131876, 2021.

[18] Jinyu Li et al. Recent advances in end-to-end automatic speech recognition. *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022.

[19] Mishaim Malik, Muhammad Kamran Malik, Khawar Mehmood, and Imran Makhdoom. Automatic speech recognition: a survey. *Multimedia Tools and Applications*, 80:9411–9457, 2021.

[20] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[21] Dzmitry Bahdanau. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[22] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022. URL https://arxiv.org/abs/2212.04356.

[23] Elena Rastorgueva and Nithin Rao Koluguri. New standard for speech recognition and translation from the nvidia nemo canary model. *Nvidia Developer Blog*, 2024.

[24] Somshubra Majumdar and Nithin Rao Koluguri. Pushing the boundaries of speech recognition with nvidia nemo parakeet asr models. *Nvidia Developer Blog*, 2024.

[25] Hainan Xu, Nithin Rao Koluguri, and Somshubra Majumdar. Turbocharge asr accuracy and speed with nvidia nemo parakeet-tdt. *Nvidia Developer Blog*, 2024.

[26] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber. Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4211–4215, 2020.

[27] Arnaldo Candido Junior, Edresson Casanova, Anderson Soares, Frederico Santos de Oliveira, Lucas Oliveira, Ricardo Corso Fernandes Junior, Daniel Peixoto Pinto da Silva, Fernando Gorgulho Fayet, Bruno Baldissera Carlotto, Lucas Rafael Stefanel Gris, et al. Coraa asr: a large corpus of spontaneous and prepared speech manually validated for speech recognition in brazilian portuguese. *Language Resources and Evaluation*, pages 1–33, 2022.

[28] Centro Ramón Piñeiro para a Investigación en Humanidades (CRPIH) and Multimedia Technology Group (GTM) – atlanTTic Research Center for Telecommunication Technologies. Crpih_uvigo-gl-voices: Galician tts dataset, June 2023. URL https://doi.org/10.5281/zenodo.8027725.

[29] P. S. Sathidevi and Y. Venkataramani. Perceptual audio coding using sinusoidal/optimum wavelet representation. *Circuits, Systems and Signal Processing*, 21: 511–524, 2002. doi: 10.1007/s00034-002-0402-8.

[30] Jonatas Grosman. Fine-tuned XLSR-53 large model for speech recognition in Portuguese. https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-portuguese, 2021.

[31] Lucas Rafael Stefanel Gris, Edresson Casanova, Frederico Santos de Oliveira, Anderson da Silva Soares, and Arnaldo Candido Junior. Brazilian portuguese speech recognition using wav2vec 2.0, 2021. URL https://arxiv.org/abs/2107.11414.

[32] H. Nanjo and T. Kawahara. A new asr evaluation measure and minimum bayes-risk decoding for open-domain speech understanding. In *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 1, pages I/1053–I/1056 Vol. 1, 2005. doi: 10.1109/ICASSP.2005.1415298.