Phonetic segmentation for Brazilian Portuguese based on a self-supervised model and forced-alignment

Eduardo S. e S. Buarque[2], Joel F. F. Gomes[12], Ubiratan da S. Tavares[2], Mário Uliani Neto[1], Fernando O. Runstein[1], Ricardo P. V. Violato[1], Marcus Lima[2]

[1]CPQD - Centro de Pesquisa e Desenvolvimento, Campinas, SP, Brasil

[2]Pontifícia Universidade Católica de Campinas, SP, Brasil

{sensit.info, ust1973, marcuslima3}@gmail.com,
{jfgomes, uliani, runstein, rviolato}@cpqd.com.br

Abstract. *A phonetic segmentation system for Brazilian Portuguese was developed using the Wav2Vec2 model, a self-supervised learning framework. The study explores the application of Wav2Vec2 in automatically determining phonetic boundaries within speech signals. By leveraging the rich acoustic representations learned by Wav2Vec2, we aim to improve the accuracy of phonetic segmentation. The system's performance was compared with the Montreal Forced Aligner (MFA), demonstrating notable effectiveness in various speech conditions, including neutral and expressive voices. Our methodology involves preprocessing phonetic transcriptions, utilizing the model for alignment, and post-processing to determine precise phonetic boundaries. Results indicate significant advancements in phonetic boundary detection, especially in challenging contexts such as expressive speech.*

Resumo. *Um sistema de segmentação fonética para o português brasileiro foi desenvolvido utilizando o modelo Wav2Vec2, uma estrutura de aprendizado auto-supervisionado. O estudo explora a aplicação do Wav2Vec2 na determinação automática de fronteiras fonéticas dentro de sinais de fala. Aproveitando as representações acústicas ricas aprendidas pelo Wav2Vec2, buscamos melhorar a precisão da segmentação fonética. O desempenho do sistema foi comparado com o Montreal Forced Aligner (MFA), demonstrando eficácia notável em várias condições de fala, incluindo vozes neutras e expressivas. Nossa metodologia envolve pré-processamento de transcrições fonéticas, utilização do modelo para alinhamento e pós-processamento para determinar limites fonéticos precisos. Os resultados indicam avanços significativos na detecção de fronteiras fonéticas, especialmente em contextos desafiadores como a fala expressiva.*

1. Introdução

O objetivo geral de qualquer sistema de alinhamento é determinar um mapeamento preciso entre diversas representações que compartilham informações subjacentes comuns. Por exemplo, enquanto o alinhamento áudio-partitura visa recuperar os tempos de início e término de cada evento relatado em uma partitura musical para uma determinada performance, o alinhamento letra-áudio é focado na localização temporal das palavras pronunciadas em uma gravação.

No contexto da segmentação fonética, os sinais de fala são analisados para identificar e marcar os instantes de tempo das fronteiras entre os diferentes fones, isto é, os pontos de início e fim de cada fone dentro do fluxo contínuo da fala.

Dado um arquivo de áudio com fala e a transcrição fonética do texto nele contido, espera-se que um sistema de segmentação fonética determine automaticamente os limites de tempo para cada fone, fazendo assim a divisão do sinal de fala em unidades distintas correspondentes aos fones, ou seja, realizando a segmentação dos fones.

A segmentação fonética é um desafio na área de tecnologias de fala. A complexidade surge devido à variação na pronúncia das palavras em uma fala natural, a presença de coarticulação e variações individuais na fala.

Neste contexto, os algoritmos de segmentação fonética buscam identificar fronteiras precisas entre os fonemas, contribuindo para a melhoria da precisão em tarefas como síntese de fala e transcrição de fala.

Este trabalho apresenta a aplicação do modelo auto-supervisionado Wav2Vec2 na determinação automática das fronteiras fonéticas em sinais de fala no idioma português do Brasil. Está organizado da seguinte forma: na seção 2 são apresentados os trabalhos relacionados à tarefa de alinhamento fonético que fazem uso de modelos de inteligência artificial; a seção 3 apresenta a metodologia utilizada, com a descrição do método de segmentação baseado em alinhamento forçado, a adaptação do modelo baseado em caractere para fone, e a estratégia de pós-processamento para ajuste das fronteiras fonéticas; a seção 4 apresenta detalhes do método utilizado para avaliar o resultado da segmentação fonética; a seção 5 apresenta uma discussão sobre os resultados apresentados; e, por fim, a seção 6 apresenta algumas conclusões e sugestões de trabalhos futuros.

2. Trabalhos Relacionados

2.1. Montreal Forced Aligner (MFA)

Em segmentação fonética, o *Montreal Forced Aligner* (MFA) [McAuliffe et al. 2017] destaca-se como uma linha de base fundamental, devido à sua eficácia comprovada no alinhamento forçado de fala.

O MFA é uma ferramenta de ponta em linguística computacional e tecnologias de fala, que utiliza modelos baseados em *Hidden Markov Models* (HMMs) para alinhar automaticamente transcrições fonéticas com gravações de áudio. A escolha do MFA como *baseline* neste projeto decorre de sua alta precisão, eficiência e flexibilidade, essenciais para lidar com a complexidade da segmentação fonética.

Sua capacidade de processar fala contendo variações de sotaques e estilos de fala, aliada a um robusto treinamento em múltiplos corpora, incluindo o corpus TIMIT [Garofolo et al. 1993], torna-o uma referência para avaliar e aprimorar novas metodologias e técnicas no campo da segmentação fonética de fala.

2.2. Wav2Vec2

O Wav2Vec2 [Baevski et al. 2020] é um modelo de treinamento de aprendizado auto-supervisionado desenvolvido pelo Facebook AI Research (FAIR) para a tarefa de reconhecimento automático de fala. Foi treinado com grande quantidade de fala bruta não

rotulada e projetado para lidar com a representação acústica de áudio de forma eficiente; o modelo tem se mostrado muito eficaz em várias tarefas de processamento de fala.

Wav2Vec2 é uma extensão do modelo original Wav2Vec [Schneider et al. 2019], que foi projetado para aprender representações acústicas de áudio de forma não-supervisionada. A principal inovação do Wav2Vec2 é a sua capacidade de pré-treinar representações de áudio brutas antes de ajustá-las para tarefas específicas como o reconhecimento de fala.

O Wav2Vec2 se destaca na segmentação fonética devido à sua capacidade de aprender representações acústicas ricas e detalhadas através de seu pré-treinamento com grandes volumes de fala não rotulada. Isso permite que o modelo identifique fones com alta precisão, capturando nuances sutis na fala. A abordagem de mascaramento usada no pré-treinamento força o modelo a aprender profundamente as características acústicas do áudio, o que é essencial para distinguir entre os diferentes fones do idioma português do Brasil.

Além disso, a arquitetura *transformers* [Vaswani et al. 2017] do Wav2Vec2 é eficaz em modelar dependências temporais, facilitando a identificação das fronteiras entre os fones em uma sequência de fala contínua. A combinação dessas características torna o Wav2Vec2 altamente eficiente na segmentação fonética, apresentando um desempenho superior em comparação com métodos anteriores e mostrando ótimos resultados mesmo em contextos de dados limitados ou com varíações nas características da fala (natural, com emoções ou com diferentes estilos expressivos).

3. Metodologia

3.1. Abordagem Utilizada

Este trabalho utilizou um modelo pré-treinado baseado no Wav2Vec2, treinado com mais de 31.000 horas de dados em 1.130 idiomas do *Scaling Speech Technology to 1,000+ Languages* [Pratap et al. 2024], denominado MMS_FA [Hwang et al. 2023], para alinhar áudios com transcrições ortográficas de maneira eficiente. A aplicação de técnicas de tokenização e alinhamento temporal, em conjunto com um processamento dinâmico, permite realizar uma segmentação ortográfica precisa.

O fluxo de execução da segmentação inicia com o carregamento dos arquivos de áudio e sua transcrição fonética. Essa transcrição fonética é traduzida para uma transcrição ortográfica equivalente, a qual é passada, juntamente com o áudio, para o MMS_FA. Este último fará o alinhamento ortográfico utilizando o modelo pré-treinado, o qual será convertido para o alinhamento fonético utilizando a tradução fonética-ortográfica previamente definida. Após essa tradução são feitos ajustes entre as fronteiras de cada fone, para capturar a sequëncia de fala contínua, finalizando com a exportação das segmentações ajustadas para um novo arquivo. O pipeline detalhado pode ser visto na Figura 1 e a descrição de cada etapa pode ser vista nas seções seguintes.

3.2. Modelo de Segmentação Baseado em Redes Neurais

O modelo MMS_FA é fornecido pelo *torchaudio.pipelines* [Hwang et al. 2023] e é especializado na tarefa de alinhamento forçado. Foi treinado para realizar o alinhamento ortográfico do texto com o áudio. Este modelo foi desenvolvido como parte do projeto de

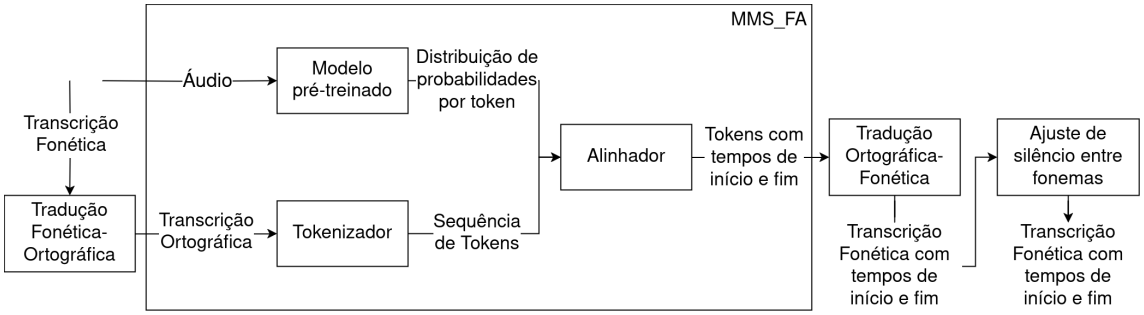

Figura 1. Pipeline Proposto - Segmentação fonética

pesquisa *Scaling Speech Technology to 1,000+ Languages* e é de código aberto. No contexto deste trabalho, o MMS_FA é usado para gerar emissões de padrões e características acústicas a partir do áudio fornecido, que são posteriormente utilizadas para realizar o alinhamento fonético.

3.3. Alinhamento Forçado

O alinhamento forçado é uma técnica utilizada para mapear sequências de áudio com suas transcrições correspondentes. O algoritmo CTC (*Connectionist Temporal Classification*) [Graves et al. 2006] é amplamente utilizado para este propósito. A biblioteca PyTorch [Paszke et al. 2019], através do módulo torchaudio, oferece uma API que facilita a implementação do alinhamento forçado, utilizando o algoritmo de segmentação CTC descrito em "*CTC-segmentation of large corpora for German end-to-end speech recognition*" [Kürzinger et al. 2020].

Para realizar o alinhamento forçado utilizando a biblioteca *torchaudio* e o modelo pré-treinado, são necessários dois componentes: o áudio e sua transcrição ortográfica. O *pipeline* de alinhamento pode ser visto na Figura 2, e segue os seguintes passos:

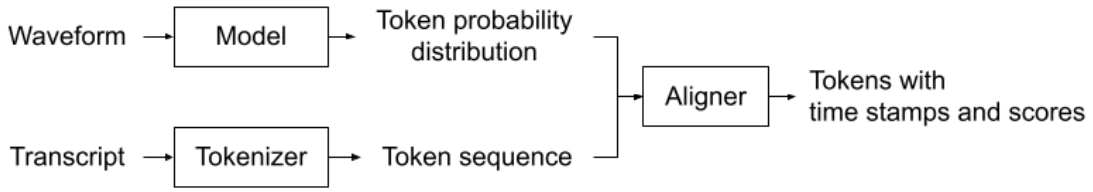

Figura 2. Pipeline Wav2Vec2 - Forced Alignment [1]

1. Entrada de Áudio e Transcrição: A forma de onda (*waveform*) do áudio é passada para o modelo acústico que produz a sequência de distribuição de probabilidade dos tokens. Simultaneamente, a transcrição ortográfica é passada para o tokenizador (*tokenizer*), que a converte em uma sequência de tokens.
2. Alinhamento: O alinhador (*aligner*) utiliza os resultados do modelo acústico e do tokenizador para gerar um instante temporal específico (*timestamps*) para cada token.

[1] https://pytorch.org/audio/stable/tutorials/forced_alignment_for_multilingual_data_tutorial.html#portuguese

O modelo acústico e o tokenizador utilizam o mesmo conjunto de tokens. O MMS_FA associa um modelo acústico pré-treinado, um tokenizador (que utiliza o mesmo conjunto de tokens que o modelo) e um alinhador. O tokenizador mapeia os caracteres normalizados para números inteiros, de acordo com os tokens definidos no pré-treinamento e seus códigos correspondentes especificados pela bibiloteca torchaudio.

3.4. Adaptação para Segmentação Fonética

Para utilizar o pipeline com a transcrição fonética ao invés da transcrição ortográfica, foram necessárias duas etapas adicionais: pré-processamento e pós-processamento.

1. Pré-processamento: Esta etapa converte a transcrição fonética em uma transcrição ortográfica correspondente, utilizando os mesmos tokens usados pelo modelo acústico e pelo tokenizador. Um conjunto de regras mapeia cada fone do português para uma representação ortográfica correspondente.
2. Pós-processamento: Após rodar o pipeline, os tokens são convertidos de volta em fone, utilizando o alinhamento por tokens para calcular o timestamp de início e fim de cada fone. A conversão é baseada nas mesmas regras que mapeiam a conversão de fones para a representação ortográfica.

Por exemplo, a seguinte sentença:

- *O discurso de encerramento foi brilhante*

Com a seguinte representação da transcrição fonética:

- *uc - dz ii ss kk uu rf ss uc - dz ic - en ss ee rx aa mm en tt uc - ff oo ij - bb rd ii lh an ts ic*

Foi convertida para a seguinte representação de tokens:

- *u diskursu di ensserramentu foi brilhanti*

Tabela 1. Alinhamento por Token

Token	Tempo (ms)
e	[640, 660)
n	[700, 740)
s	[740, 760)
e	[820, 840)
r	[880, 900)
r	[900, 920)
a	[940, 960)
m	[1.000, 1.020)
e	[1.080, 1.100)
n	[1.160, 1.200)
t	[1.260, 1.280)
u	[1.300, 1.320)

Tabela 2. Alinhamento por Fone

Token	Tempo (ms)
en	[640, 740)
ss	[740, 760)
ee	[820, 840)
rr	[880, 920)
aa	[940, 960)
mm	[1.000, 1.020)
en	[1.080, 1.200)
tt	[1.260, 1.280)
uc	[1.300, 1.320)

Após o alinhamento, cada token terá um timestamp de início e fim, conforme a Tabela 1, permitindo o cálculo dos timestamps dos fone correspondentes, conforme a Tabela 2. Com isso, obtém-se a segmentação fonética do áudio fornecido.

3.5. Pós-Processamento das fronteiras fonéticas

Dado o comportamento do CTC de estimar o padrão central dos fones [Zeyer et al. 2021], podemos verificar na Tabela 2 que o modelo pode não caracterizar a fronteira exata entre os fones, uma vez que o alinhador produz fronteiras entre fone com lacunas como entre os fonemas “ss”, que termina em 760, e o seguinte “ee” que começa no 820.

Por isso, foram verificadas três diferentes abordagens para determinar a fronteira entre os fones vizinhos:

- **Início**: Utiliza a marcação de início dos fones como fronteira.
- **Fim**: Utiliza a marcação de final dos fones como fronteira.
- **Média**: utiliza o ponto médio entre o fim de um fone e o início do fone seguinte como fronteira.

Cada abordagem pode ser visualizada conforme as Tabelas 3, 4 e 5, respectivamente, e tiveram como base a Tabela 2, para o cálculo do ajuste das fronteiras fonéticas.

Tabela 3. Início

Token	Tempo (ms)
en	[640, 740)
ss	[740, 820)
ee	[820, 880)
rr	[880, 940)
aa	[940, 1.000)
mm	[1.000, 1.080)
en	[1.080, 1.260)
tt	[1.260, 1.300)
uc	[1.300, 1.320)

Tabela 4. Fim

Token	Tempo (ms)
en	[640, 740)
ss	[740, 760)
ee	[760, 840)
rr	[840, 920)
aa	[920, 960)
mm	[960, 1.020)
en	[1.020, 1.200)
tt	[1.200, 1.280)
uc	[1.280, 1.320)

Tabela 5. Médio

Token	Tempo (ms)
en	[640, 740)
ss	[740, 790)
ee	[790, 860)
rr	[860, 930)
aa	[930, 980)
mm	[980, 1.050)
en	[1.050, 1.230)
tt	[1.230, 1.290)
uc	[1.290, 1.320)

4. Resultados

Os resultados do estudo mostram que o desempenho do modelo pré-treinado Wav2Vec2 em comparação ao *Montreal Forced Aligner* (MFA) varia significativamente dependendo da métrica de avaliação utilizada para as fronteiras dos fones. As análises foram realizadas considerando as três abordagens propostas na seção 3.5.

Foram utilizados três conjuntos de dados para avaliação de métricas: (1) **Corpus Rosana**: Conjunto de dados com 500 áudios com voz neutra segmentados manualmente. (2) **Corpus Adriana**: Conjunto de dados com 500 áudios com vozes expressivas segmentados manualmente. (3) **Corpus Adriana (Com Falha)**: Conjunto com 19 áudios com estilo de fala expressiva que apresentaram erro utilizando o MFA. Ou seja, o MFA não foi capaz de realizar a segmentação fonética desse conjunto de arquivos.

Na Tabela 6 é apresentada a distribuição do erro de segmentação por corpus, abordagem e classe de tolerância, e na Tabela 7 a acurácia do modelo por corpus, abordagem e classe de tolerância. Os erros foram classificados calculando a diferença entre a previsão do modelo e arquivos rotulados e revisados e computando-os de forma não cumulativa, considerando as faixas encontradas nas tabelas.

[1]O MFA não conseguiu segmentar 8 áudios do Corpus Rosana.

[2]O MFA não conseguiu segmentar 11 áudios do Corpus Adriana (Com Falha).

Tabela 6. Distribuição do erro de segmentação (em ms) por classe de tolerância.

Corpus	Abordagem	10ms	25ms	50ms	100ms
Rosana	MFA[1]	41.038%	41.920%	12.106%	3.625%
	Wav2Vec2 (Ajuste Início)	13.828%	28.721%	43.646%	13.476%
	Wav2Vec2 (Ajuste Fim)	30.815%	35.021%	25.363%	8.349%
	Wav2Vec2 (Ajuste Média)	39.733%	40.572%	18.003%	1.602%
Adriana	MFA	26.970%	41.818%	17.340%	9.158%
	Wav2Vec2 (Ajuste Início)	16.555%	29.246%	36.454%	16.649%
	Wav2Vec2 (Ajuste Fim)	23.518%	32.660%	25.363%	28.867%
	Wav2Vec2 (Ajuste Média)	37.929%	37.700%	20.547%	3.569%
Adriana (C/ Falha)	MFA[2]	19.186%	23.837%	7.558%	0.000%
	Wav2Vec2 (Ajuste Média)	36.481%	39.914%	19.456%	4.006%

Tabela 7. Acurácia em diferentes tolerâncias para diferentes abordagens.

Corpus	Abordagem	10ms	25ms	50ms	100ms
Rosana	MFA[1]	**41.038%**	**82.958%**	95.064%	98.689%
	Wav2Vec2 (Ajuste Início)	13.828%	42.549%	86.195%	99.671%
	Wav2Vec2 (Ajuste Fim)	30.815%	65.836%	91.200%	99.549%
	Wav2Vec2 (Ajuste Média)	39.733%	80.305%	**98.308%**	**99.910%**
Adriana	MFA	26.970%	68.788%	86.128%	95.286%
	Wav2Vec2 (Ajuste Início)	16.555%	45.800%	82.254%	98.903%
	Wav2Vec2 (Ajuste Fim)	23.518%	56.179%	85.046%	97.916%
	Wav2Vec2 (Ajuste Média)	**37.929%**	**75.629%**	**96.177%**	**99.746%**
Adriana (C/ Falha)	MFA[2]	19.186%	43.023%	50.581%	50.581%
	Wav2Vec2 (Ajuste Média)	**36.481%**	**76.395%**	**95.851%**	**99.857%**

Além disso, para melhor análise, podemos ver a distribuição do erro médio de cada fonema, para a abordagem Wav2Vec2 (Ajuste Média): na Figura 3 para o Corpus Rosana; e na Figura 4 para o Corpus Adriana.

5. Discussão

Os resultados indicam que a escolha da métrica de avaliação das fronteiras dos fonemas impacta significativamente o desempenho do modelo Wav2Vec2, principalmente devido ao comportamento do CTC em retornar um intervalo entre fones vizinhos ao invés da fronteira exata [Zeyer et al. 2021].

A abordagem de ajuste pela média foi a mais eficaz, com alta precisão em menores tolerâncias. Para o corpus Rosana, foram analizados 22.158 fones, com um erro médio de 15,732 ms e desvio padrão de 12,916 ms. A distribuição do erro médio por fone mostra que os fones de silêncio apresentaram os maiores erros, sugerindo a necessidade de ajuste para melhorar a acurácia.

Para áudios expressivos, a abordagem de ajuste pela média também se mostrou mais eficaz. No corpus Adriana, analisamos 20.061 fonemas, com um erro médio de 17,715 ms e desvio padrão de 16,344 ms. A distribuição do erro médio por fonema revela que os fonemas de silêncio apresentaram os maiores erros, enquanto outros fonemas não

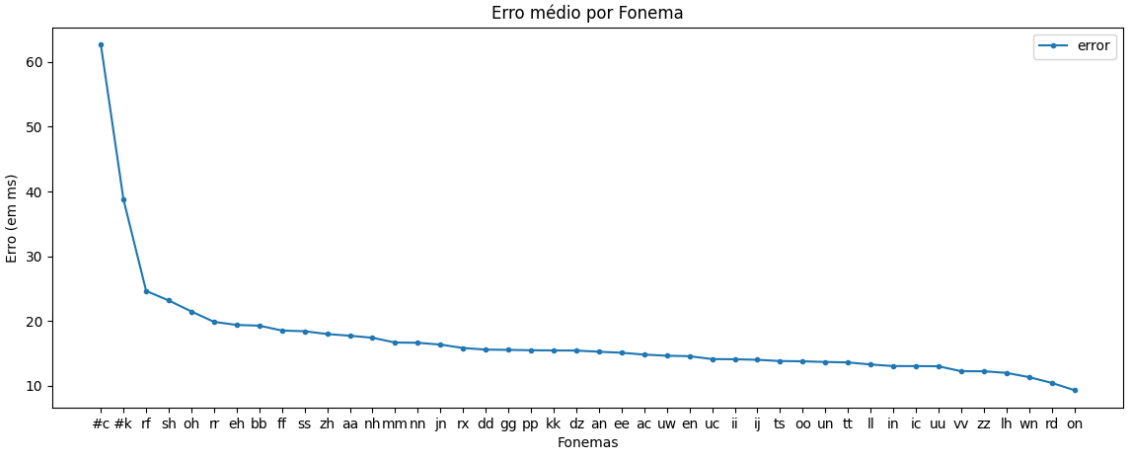

Figura 3. Erro Médio por Fonema - Corpus Rosana

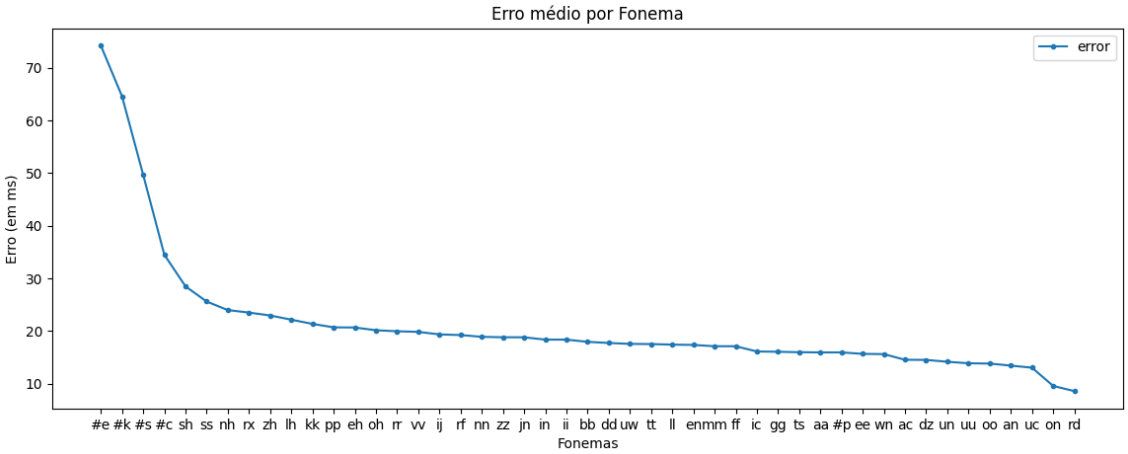

Figura 4. Erro Médio por Fonema - Corpus Adriana

ultrapassaram um erro médio de 25 ms.

Os resultados mostram que o Wav2Vec2 supera significativamente o MFA no Corpus Adriana (com Falha), com uma precisão de 99,857% em até 100 ms e 95,851% em até 50 ms, enquanto o MFA falha em 49,419% dos fones dentro de 100 ms de tolerância.

Foi observado que o intervalo entre fones vizinhos impactam negativamente as métricas de segmentação do Wav2Vec2, com erros entre 25 e 50 ms. Essa característica se reflete principalmente nos fones de silêncio, que apresentaram os maiores erros médios, conforme as figuras 3 e 4.

Esses resultados indicam que o Wav2Vec2, especialmente com ajuste pela média, oferece uma melhoria significativa na segmentação fonética, particularmente em áudios expressivos, superando a *baseline* estabelecida com o MFA. Ajustes adicionais para os intervalos entre os fones podem melhorar ainda mais a precisão do modelo na determinação do instante de tempo que separa dois fones.

6. Conclusão

Os resultados deste estudo demonstram que o modelo pré-treinado Wav2Vec2, quando aplicado ao alinhamento fonético para o idioma português do Brasil, apresenta um desempenho superior ao obtido com o MFA, especialmente com os ajustes das fronteiras fonéticas proposto neste trabalho igual ou superiores a 25 ms. O MFA só apresentou resultado superior na faixa de erro de 10 ms. Isso ocorreu devido a que a precisão da inferência de emissão dos estados de duração do modelo wav2vec2 usa passos de 20 ms. No entanto, uma marcação de fronteira com variância de 10 ms pode ser subjetiva em sinais de fala, pois essa diferença até aparece nas marcações manuais de segmentação dos arquivos de teste feita por pessoas diferentes. O problema maior nas tarefas relacionadas ao processamento de fala estão relacionados com erros iguais ou maiores do que 25 ms.

A avaliação utilizando o corpus Rosana, com dados rotulados e segmentados manualmente, permitiu uma análise precisa do desempenho do Wav2Vec. Foram exploradas três abordagens diferentes para determinar as fronteiras dos fonemas: início, fim e ponto médio entre os fones. Os resultados mostraram que a abordagem de ajuste pela média teve um desempenho próximo ao do MFA para faixas de erros menores (10 e 25 ms) e superou em casos de faixas de erros maiores (50 e 100 ms). Esta abordagem mostrou-se promissora, pois minimiza os erros de alinhamento superiores a 50 ms, que indicam desvios significativos.

A abordagem de ajuste pela média com o Wav2Vec2 apresentou um erro médio de 15,732 ms e um desvio padrão de 12,916 ms para o corpus Rosana, envolvendo um total de 22.158 fonemas. A análise da distribuição do erro médio por fonema destacou que os fonemas de silêncio, marcados como #c e #k, tiveram os maiores erros médios. Esses resultados sugerem que o tratamento do fones de silêncio é crucial para melhorar a precisão do alinhamento fonético.

Por fim, como informado anteriormente, o MFA apresentou erros de alinhamento em 8 dos 500 arquivos da base “Rosana” e em 11 dos 19 arquivos da base “Adriana (Com Falha)”, enquanto o Wav2Vec2 executou 100% dos alinhamentos sem erros, mostrando maior robustez na tarefa da segmentação fonética.

6.1. Trabalhos Futuros

Sugestões para aprimorar os resultados e superar as limitações observadas:

- Aprimoramento do Tratamento de Silêncios: Desenvolver técnicas específicas para lidar com os fones de silêncio que impactam significativamente no alinhamento. Isso pode envolver a introdução de modelos ou heurísticas adicionais que identifiquem e ajustem as pausas de silêncio de maneira mais precisa, como o uso de um detector de atividade de voz (VAD, do inglês *Voice Active Detection*).
- Treinamento Personalizado do Wav2Vec2: Realizar um *fine-tuning* do modelo Wav2Vec2 com um corpus específico que inclua variações de silêncio entre palavras e diferentes estilos expressivos de fala. Isso pode ajudar o modelo a aprender melhor as características dos silêncios e reduzir os erros de alinhamento.
- Avaliação com Diversos Corpus: Expandir a avaliação do desempenho do Wav2Vec2 utilizando outros corpus para verificar a generalização dos resultados e ajustar os métodos conforme necessário.

- Desenvolvimento de Métricas Adicionais: Criar e aplicar novas métricas de avaliação que considerem não apenas a precisão temporal das fronteiras dos fonemas, mas também a percepção auditiva dos erros de alinhamento, para fornecer uma avaliação mais holística do desempenho do modelo.
- Análise de Impacto de Variações Linguísticas: Investigar como variações linguísticas, como sotaques e dialetos, afetam o desempenho do Wav2Vec e ajustar o modelo para melhorar a robustez e a precisão em diferentes contextos linguísticos.

Através destes trabalhos futuros, esperamos não apenas melhorar a precisão do alinhamento fonético com o modelo Wav2Vec, mas também avançar no estado da arte em segmentação fonética de fala, contribuindo para aplicações mais robustas e precisas em reconhecimento de fala, síntese de fala e processamento de linguagem natural.

7. Agradecimentos

Este projeto foi apoiado pelo programa PPI Softex, Acordo de Parceria nº 0200-120/2022, financiado pelo Ministério da Ciência, Tecnologia e Inovações com recursos da Lei nº 8.248, de 23 de outubro de 1991.

Referências

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., and Pallett, D. S. (1993). Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon technical report n*, 93:27403.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Hwang, J., Hira, M., Chen, C., Zhang, X., Ni, Z., Sun, G., Ma, P., Huang, R., Pratap, V., Zhang, Y., et al. (2023). Torchaudio 2.1: Advancing speech recognition, self-supervised learning, and audio processing components for pytorch. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–9. IEEE.

Kürzinger, L., Winkelbauer, D., Li, L., Watzel, T., and Rigoll, G. (2020). Ctc-segmentation of large corpora for german end-to-end speech recognition. In *International Conference on Speech and Computer*, pages 267–278. Springer.

McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., and Sonderegger, M. (2017). Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, volume 2017, pages 498–502.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Pratap, V., Tjandra, A., Shi, B., Tomasello, P., Babu, A., Kundu, S., Elkahky, A., Ni, Z., Vyas, A., Fazel-Zarandi, M., et al. (2024). Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(97):1–52.

Schneider, S., Baevski, A., Collobert, R., and Auli, M. (2019). wav2vec: Unsupervised Pre-Training for Speech Recognition. In *Proc. Interspeech 2019*, pages 3465–3469.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Zeyer, A., Schlüter, R., and Ney, H. (2021). Why does ctc result in peaky behavior? *arXiv preprint arXiv:2105.14849*.