UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS DE QUIXADÁ

CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

LIOMAR RENNER ARAUJO RABELO

UM ESTUDO DE CASO DO MODELO DE RECONHECIMENTO DE VOZ WHISPER PARA TRANSCRIÇÃO DE CONFERÊNCIAS TEDX VIA APRENDIZADO FRACO

QUIXADA

2022

LIOMAR RENNER ARAUJO RABELO

UM ESTUDO DE CASO DO MODELO DE RECONHECIMENTO DE VOZ WHISPER PARA TRANSCRIÇÃO DE CONFERÊNCIAS TEDX VIA APRENDIZADO FRACO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Orientador: Prof. Me. Francisco Erivelton Fernandes de Aragão.

QUIXADA

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

R114e Rabelo, Liomar Renner Araujo.
Um estudo de caso do modelo de reconhecimento de voz whisper para transcrição de conferências tedx via aprendizado fraco. / Liomar Renner Araujo Rabelo. – 2023.
36 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Software, Quixadá, 2023.
Orientação: Prof. Dr. Francisco Erivelton Fernandes de Aragão.

1. Inteligência artificial. 2. Aprendizagem profunda. 3. Reconhecimento automático da voz. I. Título.
CDD 005.1

LIOMAR RENNER ARAUJO RABELO

UM ESTUDO DE CASO DO MODELO DE RECONHECIMENTO DE VOZ WHISPER PARA TRANSCRIÇÃO DE CONFERÊNCIAS TEDX VIA APRENDIZADO FRACO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Aprovada em:

BANCA EXAMINADORA

Prof. Me. Francisco Erivelton Fernandes de Aragão (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Wladimir Araújo Tavares
Universidade Federal do Ceará (UFC)

Prof. Dr. Fabio Carlos Sousa Dias
Universidade Federal do Ceará (UFC)

Prof. Dr. Alberto Sampaio Lima
Universidade Federal do Ceará (UFC)

À Deus, por me dar forças nos dias aonde ninguém poderia. À minha família, por seu grande amor. Mãe, por nunca perder a esperança em mim. À minha esposa amada, por sempre me impulsionar para o melhor.

AGRADECIMENTOS

À Deus, pelo seu eterno amor.

À minha mãe, Maria Vauberlene de Araujo, que tanto sonhou com esse momento e tanto fez para tornar realidade.

À meu falecido avô, João Machado Rabelo e minha querida avó, Doroteia Aquino dos Santos, que me deram as bases necessárias para minha formação como pessoa.

À meu tio João José Aquino Machado, pelo apoio incondicional durante todo o processo da graduação.

Ao amor da minha vida, minha esposa Sabrina Hellen Andrade Oliveira, que sempre me fez acreditar que esse dia chegaria.

Ao Prof. Me. Francisco Erivelton Fernandes de Aragão, por ser um mestre como orientador e um ser humano de grande coração.

Ao Prof. Dr. Davi Sena, que extraiu o melhor de mim nas disciplinas e pelos conselhos pragmáticos.

Aos professores participantes da banca examinadora Prof. Dr. Wladimir Araújo Tavares, Prof. Dr. Fabio Carlos Sousa Dias e Prof. Dr. Alberto Sampaio Lima pelo tempo, pelas valiosas colaborações e sugestões.

À Universidade Federal do Ceará, por oferecer um ensino público de qualidade em um país que dificulta tanto o acesso ao ensino.

Ao meu amigo Luiz Henrique Mederios de Souza pelo valioso compartilhamento de conhecimento sobre a ferramenta Google Colab no qual foi essencial para a execução dos experimentos deste trabalho.

"Se você falar com um homem numa linguagem que ele compreende, isso entra na cabeça dele. Se você falar com ele em sua própria linguagem, você atinge seu coração" (Nelson Mandela)

RESUMO

Modelos de aprendizado de máquina para tarefas de reconhecimento de voz são geralmente treinados com aprendizado não supervisionado, devido a dificuldade de coletar dados rotulados de alta qualidade. O modelo de reconhecimento de voz multitarefa e multilinguagem chamado Whisper nos fornece a possibilidade de utilizar gratuitamente um modelo estado-da-arte na área de processamento de linguagem natural. A grande inovação deste modelo foi utilizar Aprendizado supervisionado fraco, misturando uma grande quantidade de dados rotulados e não rotulados, com um alto grau de diversidade de conteúdo, demonstrando que Aprendizado supervisionado fraco pode performar no mesmo nível que outros modelos estado-da-arte. Foi estudado as capacidades de reconhecimento de voz e transcrição para texto na língua portuguesa, do modelo Whisper ao aplicarmos o dataset Multilingual TEDx que contém mais de 150 horas de áudio de alta qualidade no formato .flac e totalmente na língua portuguesa, além das transcrições dos mesmos áudios no formato .vtt. Conseguimos observar uma taxa de erro abaixo de 1, utilizando a métrica Word Error Rate, flutuando entre 0.3 e 0.7, essa taxa demonstra que o modelo consegue quando exposto a entradas pequenas ter uma taxa de erro abaixo do registrado no treinamento quando exposto a entradas maiores.

Palavras-chave: Inteligência artificial; Aprendizagem profunda; Reconhecimento automático da voz.

ABSTRACT

Machine learning models for speech recognition tasks are often trained with unsupervised learning because of the difficulty of collecting high-quality labeled data. The multitasking and multilingual speech recognition model called Whisper gave us the possibility to use a state-of-art model in the area of natural language processing for free. The great innovation of this model was to use weak learning, mixing a large amount of labeled data and unlabeled data, with a high degree of content diversity, demonstrating that weak learning can perform at the same level as other state-of-art models. The Whisper model's voice recognition and transcription-to-text capabilities in Portuguese were studied by applying the Multilingual TEDx dataset, which contains more than 150 hours of high-quality audio in .flac format and entirely in Portuguese, in addition to their transcripts audios in .vtt format. We were able to observe an error rate below 1, using the Word Error Rate metric, fluctuating between 0.3 and 0.7. This rate demonstrates that the model can, when exposed to small inputs, have an error rate below that recorded in training when exposed to larger inputs.

Keywords: Artificial Intelligence; Deep Learning; Speech Recognition.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ASR	*Automatic Speech Recognition*
BERT	*Bidirectional Encoder Representations from Transformers*
Colab	*Google Colaboratory*
DL	*Deep Learning*
GPT	*Generative Pre-Training Transformer*
GPU	*Graphics Processing Units*
ML	*Machine Learning*
TEDx	*Technology, Entertainment and Design*
WER	*Word Error Rate*
Wav2Vec2.0	*Wav2Vec2 model*
IA	Inteligência Artificial
PLN	Processamento de linguagem natural

SUMÁRIO

1 INTRODUÇÃO

Recentemente pesquisadores têm se interessado cada vez mais pelo *Automatic Speech Recognition* (*ASR*), também conhecido como reconhecimento autómatico de fala, uma vez que a fala é um método de comunicação entre as pessoas (YU; DENG, 2015). *ASR* começou com sistemas simples que respondiam a um número limitado de sons e evoluiu para sistemas sofisticados que respondem fluentemente à linguagem natural. Devido o interesse de automatizar tarefas simples que necessitam da interação homem-máquina tem aumentado o interesse crescente na tecnologia *ASR* (JUANG; RABINER, 2005).

ASR pode ser definida como o processo de derivação da transcrição da fala, conhecida como sequência de palavras, em que o foco está na forma da onda de fala (YU; DENG, 2015).

O reconhecimento de voz é difícil devido à diversidade de sinais de fala (YU; DENG, 2015). Atualmente, *ASR* é amplamente aplicado em muitas funções, como boletins meteorológicos, tratamento autómatico de chamadas, cotações de ações e sistemas de consulta (JUANG; RABINER, 2005). O progresso no reconhecimento de voz foi estimulado pelo desenvolvimento de técnicas de pré-treinamento não supervisionado exemplificadas pelo framework estado da arte *Wav2Vec2 model* (*Wav2Vec2.0*) (BAEVSKI *et al.*, 2020).

Como esses métodos aprendem diretamento do áudio bruto sem a necessidade de rótulos humanos, eles podem usar produtivamente grandes conjuntos de dados de reconhecimento de voz não rotulados e tem rapidamente ampliado para mais de 1 milhão de horas de dados de treinamento (ZHANG *et al.*, 2022). Significativamente mais que as 1000 horas típicas de um conjunto de dados acadêmicos supervisionados, quando ajustada ao *benchmark* padrão, essa abordagem melhorou o estado da arte, especialmente em um ambiente com poucos dados rotulados (RADFORD *et al.*, 2022).

Esses codificadores de áudio pré-treinados aprendem representações de fala de alta qualidade, mas, como são puramente não supervisionados, não possuem um mapeamento de decodificador de desempenho equivalente para essas representações de saídas utilizáveis, necessitando de um ajuste fino para realmente executar a tarefa (BAEVSKI *et al.*, 2021). *Wav2Vec2.0* é uma exceção que vem empolgando a comunidade tendo um sistema completo de *ASR* totalmente não supervisionado (RADFORD *et al.*, 2022).

Os esforços recentes criaram *datasets* maiores para *ASR*, no entanto é necessário reconhecer que dados de alta qualidade são limitados, quando relaxamos as exigências de transcrições de alta qualidade validados por humanos (CHEN *et al.*, 2016) podemos fazer uso de

sofisticados sistemas automatizados (GALVEZ *et al.*, 2021) utilizando o método de supervisão fraca, no qual, gerou um aumento significativo em *datasets* de treinamento ruidosos, que são aqueles dados nos quais a qualidade do áudio não é das melhores e/ou possui muita interferência e ruído.

Em geral, os treinamentos clássicos acabam tendo que escolher entre quantidade e qualidade, um equilibrio entre os dois normalmente é uma boa escolha. Conjuntos de dados utilizando o método de supervisionamento fraco melhoram significativamente a robustez e a generalização dos modelos (MAHAJAN *et al.*, 2018).

No entanto, um novo modelo chamado *Whisper* aumenta para a faixa de 680mil horas de treinamento utilizando supervisão fraca. Os resultados sugerem que supervisão fraca tem sido subestimado como método para reconhecimento de voz, tendo alcançado resultados sem a necessidade das técnicas mais recentes de *ASR* (RADFORD *et al.*, 2022).

Atualmente temos uma grande quantidade de dados de áudios disponíveis em plataformas abertas, seja em formato de vídeo, *podcasts* ou outros formatos, constituindo uma alta demanda por métodos de Aprendizado supervisionado fraco (KUEHNE *et al.*, 2017). Rotular dados de reconhecimento de voz e transcrições é caro e demorado, pois requer não apenas o rótulo, mas, informações temporais também precisam ser rotuladas, tornando difícil cobrir grandes quantidades de dados em larga escala (KUEHNE *et al.*, 2017).

O modelo utilizado como base para este trabalho fornece um enorme passo em direção a uma solução. Apesar do *dataset* multiliguagem elevado do modelo, a maior parte do treinamento foi feito em inglês, sugerindo que o modelo não foi suficientemente treinado em algumas linguagens sendo uma delas o português. (RADFORD *et al.*, 2022).

Neste contexto, o objetivo do presente trabalho é utilizar o *Open AI Whisper* para analisar como o modelo pode performar em português quando seu *dataset* é aumentado, a métrica utilizada é a *Word Error Rate* (*WER*), o resultado dessa comparação é usado como base para entender melhor como o modelo *Whisper* de supervisão fraca pode performar com uma língua de padrões tão distintos do inglês.

Este trabalho está organizado da seguinte forma. O Capítulo 2 apresenta conceitos necessários para o desenvolvimento e entendimento do trabalho, a saber: *Machine Learning* (*ML*), *Deep Learning* (*DL*), Processamento de linguagem natural (PLN), *ASR*, Arquitetura *Transformers*, *WER*. O Capítulo 3 apresenta metodologia utilizada, duas questões de pesquisa, procedimentos e passos para coleta de dados, aplicação do experimento e procedimentos para análise dos dados. No Capítulo 4 é apresentado os resultados das questões de pesquisa dentro das métricas pré-estabelecidas. No Capítulo 5 apresenta as conclusões e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Machine Learning

ML é o estudo de algoritmos que fornece aos sistemas a capacidade de aprender e melhorar com a experiência. Geralmente é visto como um subcampo da Inteligência Artificial (IA)(SAH, 2020). Algoritmos de aprendizado de máquina permitem que os sistemas tomem decisões de forma autônoma sem qualquer suporte externo (SAH, 2020).

Às vezes, depois de visualizar os dados, não podemos interpretar as informações extraídas dos dados, nesse caso, aplicamos *ML*(MAHESH, 2020). Com a abundância de conjuntos de dados disponíveis a demanda está aumentando, muitos setores aplicam *ML* para extrair dados relevantes. O objetivo com *ML* é aprender com os dados (MAHESH, 2020).

Existem 4 categorias primárias de algoritmos de *ML*, são elas: Aprendizado supervisionado, Aprendizado não-supervisionado, Aprendizado por reforço e Aprendizado semi-supervisionado, conforme a figura 1:

Figura 1 – Desmistificando termos em Machine Learning - tipos de aprendizado

Fonte: Raphaell (2021).

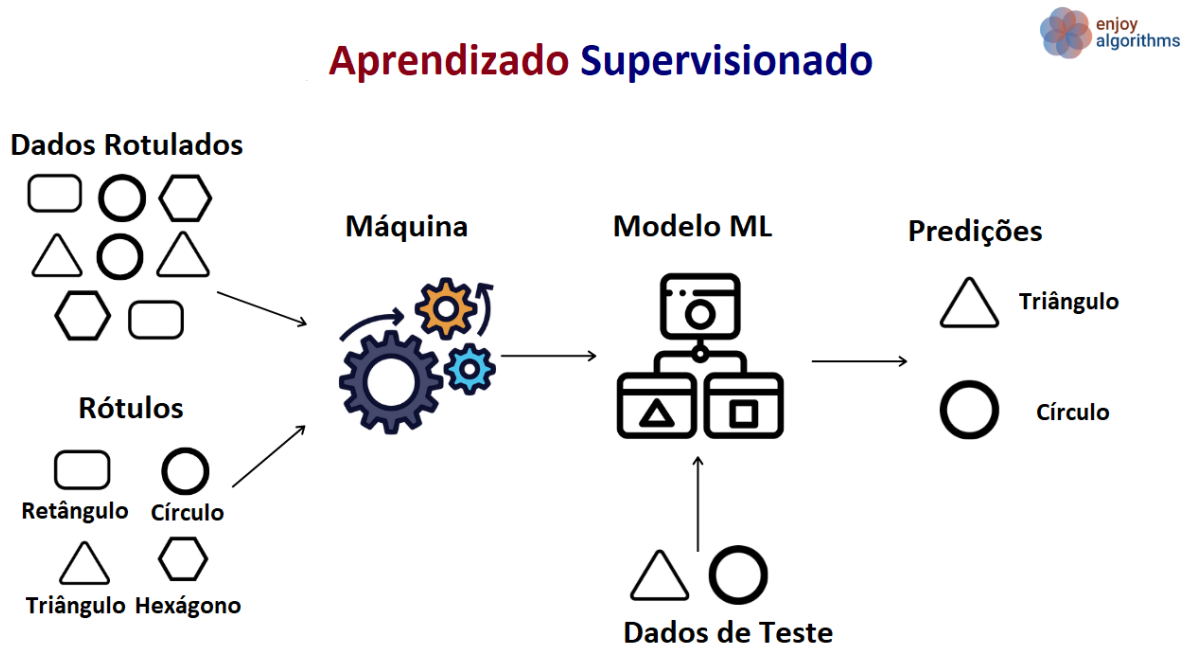
As abordagens principais contem sub métodos e técnicas ampliando ainda mais o leque de possibilidades de treinamento. Nas seções a seguir os métodos são descritos brevemente, contanto iremos descrever em mais detalhes o método semi-supervisionado, assim como a sua

estratégia de Aprendizado supervisionado fraco na seção 2.1.4.1, que são os mais relevantes para o entendimento deste trabalho.

2.1.1 Aprendizado supervisionado

O aprendizado supervisionado é a tarefa de *ML* de aprender uma função que mapeia uma entrada para uma saída com base em pares de entrada-saída de exemplo. Ele infere uma função a partir de dados de treinamento rotulados que consistem em um conjunto de exemplos de treinamento (MAHESH, 2020).

Figura 2 – Aprendizado supervisionado



Fonte: Adaptado de Raj (2021).

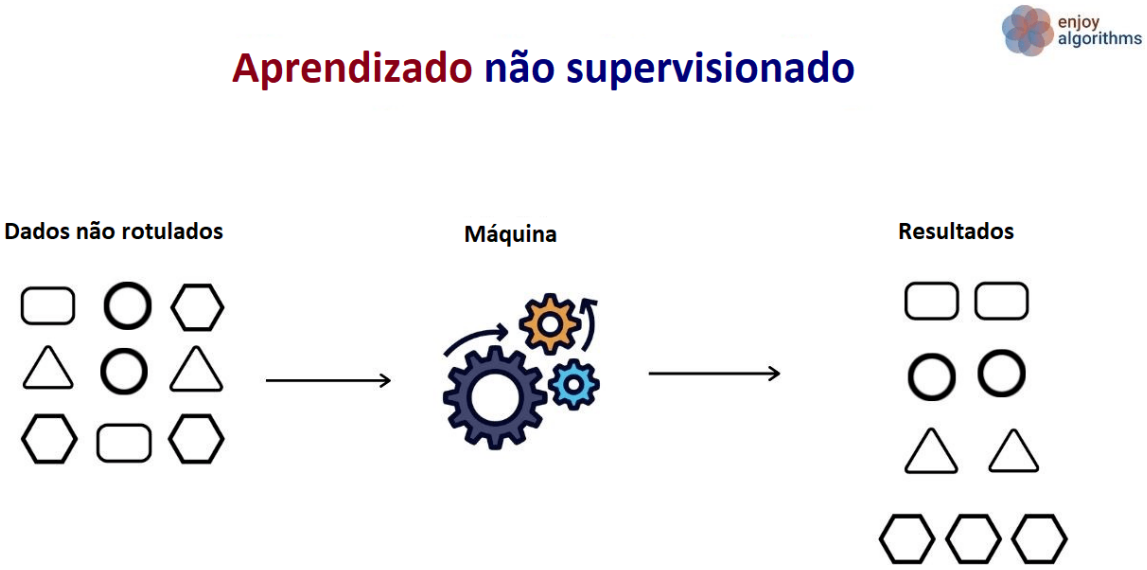
Os problemas de aprendizado supervisionado são divididos em dois tipos de problemas, os quais são: Problemas de Classificação, no qual a variável dependente é categórica; e Problemas de Regressão, no qual a variável dependente é numérica (MARSLAND, 2011). Como este trabalho não utiliza aprendizado supervisionado evitaremos entrar nos detalhes sobre este método de aprendizado.

2.1.2 Aprendizado não-supervisionado

Aprendizado não supervisionado refere-se a algortimos para identificar padrões em conjuntos de dados que não são nem classificados e nem rotulados (HASTIE *et al.*, 2009).

Em outras palavras, nenhum rótulo é dado ao algoritmo de aprendizado, deixando-o sozinho para encontrar a estrutura no conjunto de entrada (BARLOW, 1989).

Figura 3 – Aprendizado não-supervisionado



Fonte: Adaptado de Raj (2021).

No aprendizado não-supervisionado um sistema de IA agrupa informações não classificadas de acordo com semelhanças e diferenças, mesmo que não aja categorias fornecidas (BARLOW, 1989).

O principal objetivo do aprendizado não-supervisionado é descobrir padrões ocultos e interessantes em dados não rotulados (DRIDI, 2021). Existem quatro tipos de tarefas não supervisionadas: *Clustering*, *Anomaly detection*, *Principal component analysis*, *Autoencoders* (DRIDI, 2021). Este método também não é utilizado neste trabalho e novamente evitaremos entrar em detalhes.

2.1.3 Aprendizado por reforço

O aprendizado por reforço é aplicado quando a tarefa em questão é tomar uma sequência de decisões em direção a uma recompensa final. Durante o processo de aprendizado, um agente artificial recebe recompensas ou penalidades pelas ações que executa (FRANÇOIS-LAVET *et al.*, 2018). O objetivo é maximizar a recompensa total. Um exemplo inclui agentes de aprendizagem para jogar jogos de computador (FRANÇOIS-LAVET *et al.*, 2018). De todas as

formas de Aprendizado, essa é a que mais se distancia da forma de aprendizado que este trabalho tem como base, sendo as duas anteriores mais próximas do Aprendizado semi-supervisionado, no qual entraremos em detalhes a seguir.

2.1.4 Aprendizado semi-supervisionado

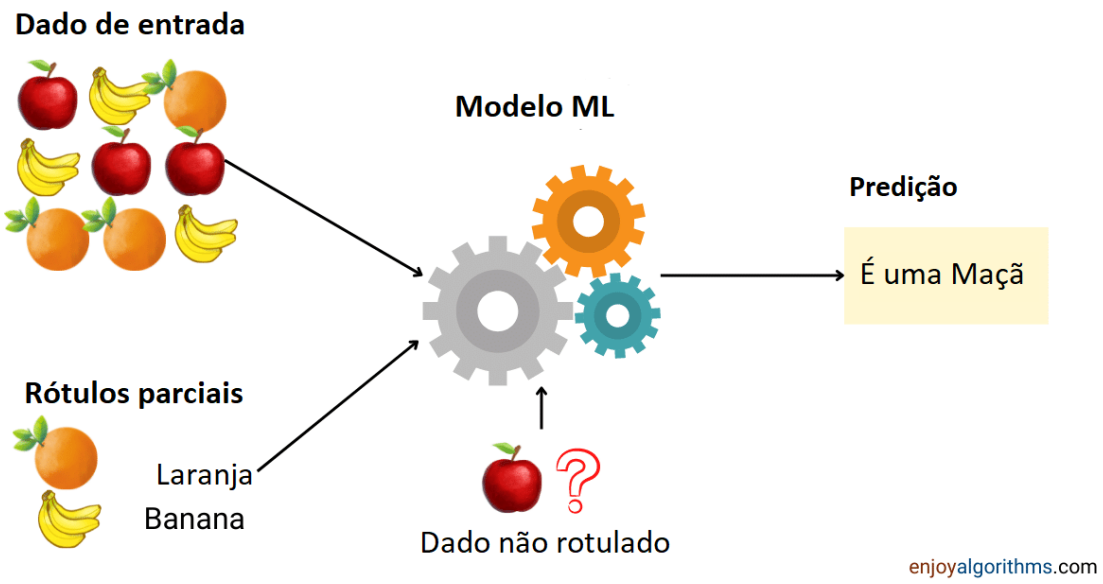
O aprendizado semi-supervisionado é um tipo de aprendizado em *ML* que se apresenta como um meio termo entre supervisionado e não-supervisionado, no qual o conjunto de dados é apenas parcialmente rotulado (REDDY *et al.*, 2018).

O objetivo principal do aprendizado semi-supervisionado é superar as limitações do aprendizado supervisionado e do aprendizados não-supervisionado (REDDY *et al.*, 2018).

No aprendizado supervisionado uma forte limitação é a exigência de grande quantidade de informações rotuladas de alta qualidade validada por humanos especialistas de um determinado sub-dominio. Embora o aprendizado não-supervisionado não precise de nenhuma informação prévia sobre os dados, esse aprendizado ainda sim é capaz de agrupar os dados pela verossimilhança, no entanto, pode encontrar dificuldades de precisão para dados desconhecidos (MEHYADIN; ABDULAZEEZ, 2021).

A figura a seguir demonstra conceitualmente um exemplo de dados parcialmente rotulados, dados de entrada, caso de teste e predição.

Figura 4 – Aprendizado semi-supervisionado



Fonte: Adaptado de Raj (2021).

2.1.4.1 Aprendizado supervisionado fraco

Aprendizado supervisionado fraco é um termo genérico abrangendo uma variedade de estudos que tentam construir modelos preditivos para diferentes tarefas de *ML*. É comum categorizá-lo em três tipos: Aprendizado supervisionado fraco incompleto, Aprendizado supervisionado fraco inexato e Aprendizado supervisionado fraco impreciso (ZHOU, 2018).

Aprendizado supervisionado fraco incompleto é quando recebemos uma pequena quantidade de dados rotulados, enquanto dados não rotulados são abundandes. Aprendizado supervisionado fraco inexato é quando algumas informações de supervisão são fornecidas, mas, não tão exatas quanto o desejado. Aprendizado supervisionado fraco impreciso é quando a informação nem sempre é verdade fundamental, em outras palavras, algumas informações na rotulagem podem conter erros (ZHOU, 2018).

2.2 Processamento de Linguagem Natural

A PLN é na sua concepção a interseção da IA e da linguística (NADKARNI *et al.*, 2011), com a intenção de fazer as máquinas compreenderam contextos complexos da linguagem natural humana. (MANNING; SCHUTZE, 1999) define a linguística como a tentativa de compreender a forma como os seres humanos conversam, assim como nossa escrita, ou seja, como adquirimos, produzimos e entendemos diversas línguas. E acrescenta que, o objetivo da PLN é analisar a linguagem, seja essa linguagem escrita ou falada.

(BEYSOLOW, 2018) diz que o objetivo do PLN é fornecer aos computadores a capacidade de entender o sentido de um texto, reconhecimento de voz, e geração de resposta a questões. Originalmente a PLN dependia puramente de regras simbólicas, entretanto a natureza irrestrita da linguagem natural, sua ambiguidade e tamanho extremamente grande, levaram a problemas ao se usar a abordagem de análise puramente símbolica (NADKARNI *et al.*, 2011).

Uma reorientação fundamental no campo foi feita e resultou na chamada PLN estatística. A análise estatística substitui regras detalhadas por regras mais amplas, constroi regras probabilísticas a partir de dados rotulados semelhante aos algoritmos de *ML*. As abordagens estatísticas dão bons resultados na prática simplesmente porque, aprendendo com dados reais abundantes, utilizam os casos mais comuns: quanto mais abundantes e representativos os dados, melhor eles ficam (NADKARNI *et al.*, 2011).

2.2.1 Speech Recognition

A principal tarefa do *ASR* é converter sinais de voz em transcrições de texto. É um dos campos de pesquisa mais importantes da PLN. Com mais de meio século de esforço, a métrica *WER*, que é uma unidade métrica para desempenho de transcrição, foi significativamente reduzida (LU *et al.*, 2020).

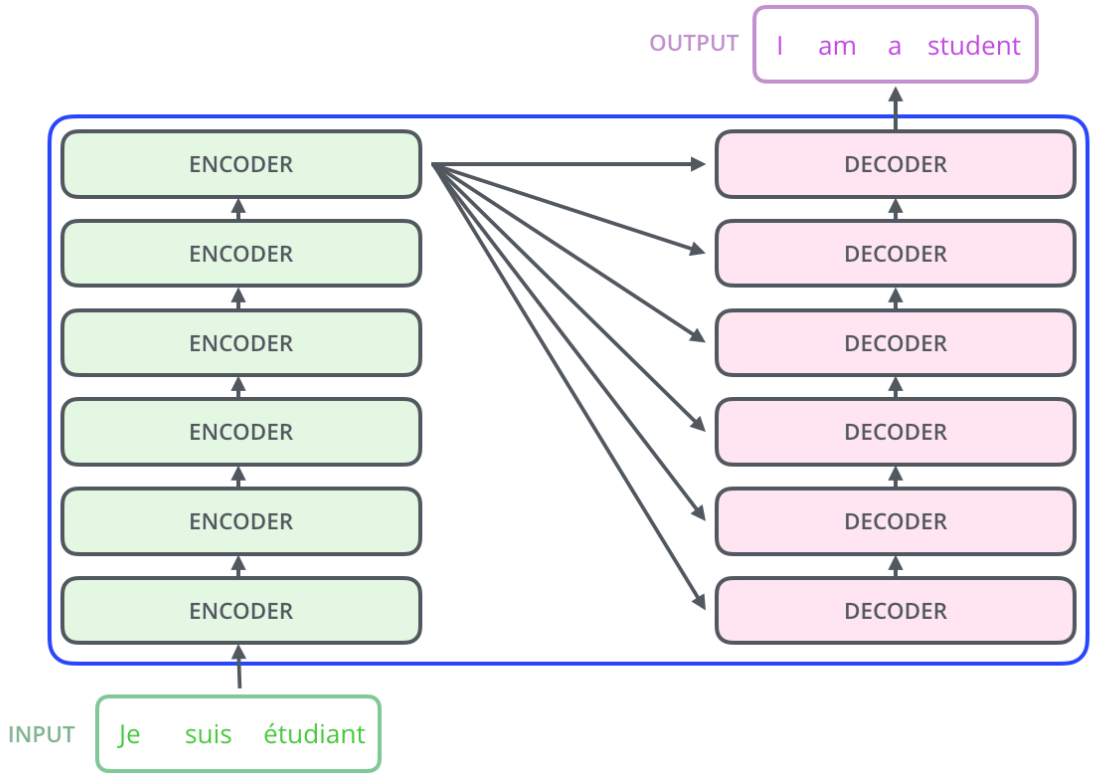
Particularmente nos últimos anos, devido ao aumento do poder computacional, grande quantidade de dados coletados e algoritmos de *DL*, o poder dominante da tecnologia de *DL* aumentou ainda mais o desempenho dos sistemas *ASR* a um nível prático (LU *et al.*, 2020).

2.3 Arquitetura Transformers

Na PLN existe uma arquitetura famosa chamada *Transformer*, cujo objetivo é executar atividades chamadas de *sequence-to-sequence*(VASWANI *et al.*, 2017) sem a necessidade de processar os dados sequenciais em ordem. O mecanismo de atenção identifica o contexto para qualquer posição na sequência de entrada (FERREIRA, 2022). Atualmente é utilizada também como arquitetura base para modelos como o *Bidirectional Encoder Representations from Transformers* (*BERT*) e *Generative Pre-Training Transformer* (*GPT*). Os transformadores gradativamente substituíram modelos de *DL* recorrente pois facilita o treinamento com conjuntos de dados maiores do que era possível anteriormente (FERREIRA, 2022).

(VASWANI *et al.*, 2017) propõe uma arquitetura que tem como base principal codificadores(*encoders*) e decodificares(*decoders*). O codificador processa a entrada atráves de suas camadas, uma camada após a outra, o mesmo processo é feito no conjunto de camadas de decodificação. O codificador transfere as informações sobre partes da entrada que são consideradas mais relevantes para a camada seguinte, enquanto o decodificador recebendo os dados gerados pelas codificações processa estes dados usando informações contextuais para gerar uma sequência de saída. O fator chave nos dois processos são os mecanismos de atenção implementados com objetivo de analisar a relevância de todas as entradas e gerar informações que ajudaram a produzir uma saída.

Figura 5 – Arquitetura transformer codificadores e decodificadores em uma tarefa de tradução



Fonte: Allamar (2018).

2.4 Word error rate

WER é uma métrica para medir performance em sistemas de *ML* e *ASR*. O desempenho dos sistemas de reconhecimento de voz geralmente são especificados em termos de precisão e velocidade. A precisão pode ser medida no que chamamos de precisão de desempenho, que geralmente é avaliada com *WER*, enquanto a velocidade é medida baseada no fator de performance em tempo real do algoritmo. Existem outras medidas de precisão, porém, não pretendemos usar neste trabalho, essas medidas são *Single Word Error Rate (SWER) ae Command Success Rate (CSR)* (NAGATA *et al.*, 1964).

A dificuldade geral ao se medir o desempenho reside no fato de que a sequência de palavras reconhecidas pode ter um comprimento diferente da sequência de palavras de referência (supostamente a correta) (GAIKWAD *et al.*, 2010).

O *WER* é derivado da *distância de Levenshtein*, trabalhando no nível da palavra ao invés do nível do fonema. Este problema é resolvido alinhando primeiro a sequência de palavras reconhecidas com a sequência de palavras de referência (falada) usando o alinhamento dinâmico de strings (RABINER; JUANG, 1993). *WER* pode então ser calculado como:

$$WER = (S + D + I) / N$$

No qual: *S* é o número de substituições, *D* é o número de exclusões, *I* é o número de inserções, *N* é o número de palavras na referência.

2.4.1 Limitações e viés

O *WER* é uma métrica valiosa para comparar diferentes sistemas, bem como para avaliar melhorias dentro de um sistema. Esse tipo de medição, no entanto, não fornece detalhes sobre a natureza dos erros de tradução e, portanto, é necessário mais trabalho para identificar as principais fontes de erro.

3 METODOLOGIA

Um dos objetivos desse trabalho é transcrever de forma automatizada áudios na língua portuguesa em textos. Este é um problema que exige abordagem multitarefa para o processo de reconhecimento de voz e posterior transformação em texto. O presente trabalho utiliza um modelo *estado-da-arte* de *ASR* multitarefa e multilinguagem treinado em larga escala em diversas tarefas de PLN, a saber: reconhecimento de voz multilinguagem, transcrição de áudio e identificação de linguagem.

Portanto, será necessário um *dataset* em português de áudios que serão extraídos do *Multilingual TEDx corpus*. Considerando que o modelo foi treinado em supervisão fraca em larga escala e já possui como parte do seu treinamento alguns dados rotulados na língua portuguesa, não será necessário rotular nenhum dos áudios utilizados, no entanto, o corpus escolhido fornece as transcrições de todos os áudios do seu *dataset*. Os arquivos possuem extensão *.flac* para os áudios e *.vtt* para as transcrições.

Como o presente trabalho utiliza um modelo já construído e faz uso de um *dataset* validado, seu principal foco será analisar os resultados das questões de pesquisa que o trabalho irá apresentar na seção a seguir.

3.1 Questões de Pesquisa

Como dito anteriormente, o *Whisper* possui treinamento em multilinguagem e multitarefa, sendo uma das linguagens treinadas o português. Contudo, foi observado que algumas linguagens possuem baixo número de horas de treinamento, trazendo dúvidas sobre a performance do modelo quando comparado com a quantidade de horas de treinamento que o modelo teve em inglês.

No entanto o português possui aproximadamente 10 mil horas de treinamento com supervisão fraca, sendo uma das linguagens com maior percentual de horas de treinamento pelo modelo e um índice *WER* de aproximadamente 5, para a tarefa de transcrição de áudio. Neste caso foi definido algumas questões de pesquisa:

- QP1: *Como o modelo Whisper performa na tarefa do reconhecimento de voz e a subsequente transcrição em texto na lingua portuguesa quando submetido a uma quantidade de horas de áudio inferior à 10h?* Com a resposta da **QP1** será possível dizer ao final se a acurácia do modelo melhora, piora ou permanece praticamente equivalente para esse cenário.

- QP2: *É possível afirmar que o modelo Whisper de supervisão fraca performa pior conforme o corpus se torna maior?* Com a resposta da **QP2** será possível comparar as diversas performances do modelo com a performance do mesmo modelo quando submetido com o corpus massivo de aproximadamente 10 mil horas, tendo como base para comparação de ambos a métrica *WER*.

3.2 Passos do estudo

Nesta seção, são apresentados os procedimentos adotados para coleta de áudios da conferência *Technology, Entertainment and Design* (*TEDx*), os procedimentos para execução do experimento e procedimentos para análise dos resultados com a métrica *WER*.

3.2.1 Procedimentos para a Coleta dos dados

O *dataset* escolhido para ser utilizado neste trabalho foi o *Multilingual TEDx corpus*, um corpus multilinguagem composto de palestras de temas variados contendo mais de 150mil horas somente na língua portuguesa (FERREIRA, 2022). As palestras *TEDx* podem ser em uma variedade de línguas, mas, compartilham do mesmo formato das tradicionais paletras *TED*, sendo estas exclusivamente em inglês. Atualmente existem mais de 150 mil *TEDx* em mais de 100 idiomas e mais de 3000 novas gravações são adicionadas a cada ano (FERREIRA, 2022).

No entando o presente trabalho fez uso apenas de áudios em português. O *corpus Multilingual TEDx* é lançado sob uma licença *CC BY-NC-ND 4.0* e pode ser baixado gratuitamente em *http://www.openslr.org/100*. O *dataset* em português atende pelo nome de *mtedxpt.tgz* e possui tamanho de *29 Gb* podendo ser baixado e inserido manualmente ou diretamente no *Google Colaboratory* (*Colab*) via linha de comando.

3.2.2 Procedimentos para Execução do experimento

O experimento irá selecionar os 5 primeiros áudios do nosso *dataset* e suas devidas transcrições, o algoritmo então será submetido a tarefa de reconhecimento de voz e transcrição do áudio. Antes do ínicio da tarefa é importante efetuarmos um processo chamado *Text Cleaning* ou limpeza de texto. Embora Whisper pareça ser muito bom em capitalização, existem usos incomuns de capitalização no texto original que Whisper poderia não conhecer. Não queremos penalizar o Whisper por isso, então converteremos todo o texto em letras minúsculas. Também

vamos remover as aspas, apesar do Whisper parecer muito bom em colocá-las no lugar certo, vamos nos concentrar nas palavras apenas no presente trabalho, evitaremos processos adicionais de *Text Cleaning*.

O modelo é então executado, quando a tarefa termina um novo arquivo de texto é gerado com o resultado da transcrição e as marcações de tempo, este arquivo será por fim utilizado juntamente com a transcrição original, sendo por fim comparados.

O código do experimento foi escrito em *Python*, utilizando o modelo *Open AI Whisper*. O experimento foi feito dentro do ambiente conhecido como *Colab*, fazendo uso do *dataset* previamente escolhido na seção 3.2.1. O download do *dataset* foi efetuado com o comando *!wget -N nomedoarquivo.tgz*, sendo posteriormente descompactado com o comando *!tar -xvzf nomedoarquivo.tgz*.

O comando para rodar o modelo necessário é *!whisper "caminho/doarquivo.flac- -model large –language Portuguese*, mais detalhes do código estará acessível ao final deste trabalho, assim como o arquivo base salvo no *Google Drive*, sendo necessário solicitar permissão ao autor para acesso ao código.

3.2.3 Procedimentos para a Análise dos dados

A análise dos dados se fez dentro do ambiente *Colab*, utilizando a métrica *WER* explicada na seção 2.5, com o resultado ao final tentaremos responder as questões de pesquisas levantadas na seção 3.1.

3.3 Ferramentas

Nesta seção abordamos brevemente sobre as ferramentas essenciais para a realização do experimento deste trabalho. Todas as ferramentas escolhidas são gratuitas, permitindo que o experimento seja facilmente replicado ou até mesmo melhorado para trabalhos futuros.

3.3.1 Python

Primeiramente, porque Python? a resposta é simples: muito poderosa, muito acessível (RASCHKA; MIRJALILI, 2019). Python tem características como: simplicidade, uma comunidade online imensa, muitas bibliotecas e *frameworks*, além de ferramentas de visualização muito úteis no campo da IA, *ML* e *DL* (OGOTI, 2021).

3.3.2 Open AI Whisper

O *Whisper* é um sistema de *ASR* treinado com mais de 680mil horas de dados multilinguagem e multitarefa coletados da web, tendo mostrado que o uso de um conjunto tão grande e diversificado leva a uma maior robutez para sotaques, ruído de fundo e linguagem técnica, além disso permite transcrição em vários idiomas (RADFORD *et al.*, 2022).

Whisper é *open-source* e é utilizado como modelo base para tarefa do seguinte trabalho de transcrever em textos os áudios extraídos do *Multilingual TEDx corpus*.

3.3.3 Google Colaboratory

O *Colab* é uma ferramenta online produto de pesquisas científicas do Google que permite escrever código na linguagem Python, sendo bastante adequado para *ML*. O *Colab* não requer configuração para usar e fornece acesso a recursos de computação como *Graphics Processing Units* (*GPU*) (FERREIRA, 2022). O *Colab* possui uma versão gratuita e duas versões pagas, na versão gratuita os tempos de execução são limitados a 12 horas e a *RAM* também é limitada a *16 GB*.

Existem duas variantes *PRO* no presente momento em que este trabalho foi executado, que são *Colab PRO* com *32 GB* de *RAM* e *Colab PRO+* que têm *52 GB* disponíveis de alta memória. No entanto, os usuários *Pro+* são priorizados se os recursos forem escassos. O tempo de execucão permitido na variante *PRO* é de 24 horas, no entando a variante *PRO+* possui uma funcionalidade que permite a execução em segundo plano, não exigindo que o navegador fique aberto, além disso, usuários *PRO+* possuem acesso às melhores *GPU*s e é o recomendado para executar algoritmos de *DL* ou *ML* em larga escala.

O presente trabalho faz uso da versão gratuita, um dos incovenientes da versão gratuita é que ao utilizar o serviço com frequência, por muito tempo e processando dados massivamente, o usuário corre o risco de ser bloqueado e esse bloqueado pode perdurar por dias e até semanas, dependendo do quão recorrente seja o uso massivo do serviço gratuito.

4 RESULTADOS

À seguir iremos abordar sobre os resultados do experimento, demonstrando o WER que tivemos e comparando com o do autor (RADFORD *et al.*, 2022) no seu trabalho *Robust Speech Recognition via Large-Scale Weak Supervision*, dessa forma, analisamos nossos dados e respondemos as questões de pesquisa propostas.

Submetemos 5 arquivos diferentes para reconhecimento de voz, transcrição de texto e posterior análise da taxa de erro, todos os arquivos escolhidos são parte do *dataset* que determinamos previamente na seção 3, iremos chamar eles pelo nome original de cada arquivo, para facilitar posterior replicação do experimento, os arquivos se chamam: *"BaBl6qoK0", "CqkAk3H9K0", "GU6AHeyv8k", "BtfBTQJoM", "cCqIYediyg"*.

4.1 QP1: Como o modelo Whisper performa na tarefa do reconhecimento de voz e a subsequente transcrição em texto na lingua portuguesa quando submetido a uma quantidade de horas de áudio inferior à 10h?

Cada arquivo que rodamos foi analisado individualmente, vejamos como ficou o resultado final na imagem abaixo:

Figura 6 – WER dos 5 arquivos usados no experimento

Fonte: Imagem gerada pelo próprio autor.

Os 5 arquivos totalizam 1 hora e 44 minutos de áudio, respectivamente o tamanho de cada áudio é: 22:49m, 18:16m, 19,14m, 14:34m, 12:55m. Podemos notar na figura 6 uma taxa de erro abaixo de profissionais humanos capacitados na tarefa de transcrição, sendo a taxa média de WER destes entre 7.61 e 10.5 (RADFORD *et al.*, 2022).

A tabela é a diferença de horas entre o treinamento e o experimento, essa pequena quantidade de horas no experimento coincide com a demanda de transcrições, quando por vezes o arquivo da aula, conferência, vídeo no youtube, tende a ser uma quantidade abaixo de 10 horas de áudio:

Tabela 1 – Horas de áudio

Horas de áudio autor	Minhas Horas de áudio
10k	1.44h

Fonte: Tabela gerada pelo próprio autor.

O WER médio do modelo durante o treinamento quando submetido à 10 mil horas na lingua portuguesa fica ligeiramente abaixo de 5 considerando apenas a tarefa de transcrição. É possível observar na figura 6 a ligeira variação na taxa de erro, sendo que no maior áudio testado(22:49m) foi onde obtivemos a maior taxa de erro(0.73...).

4.2 QP2: É possível afirmar que o modelo Whisper de supervisão fraca performa pior conforme o corpus se torna maior?

Nosso experimento consistiu da elaboração de um algoritmo no qual pode ser encontrado a síntese no apendice desse trabalho e no link do Google Drive na seção 5. Exucutamos um total de 1 hora e 44 minutos de áudio na língua portuguesa. Em seguida submetemos nosso resultado à metrica *WER* que nos responde com a taxa de erro. Replicamos o experimento 5 vezes e comparamos com a *WER* do autor (RADFORD *et al.*, 2022). A síntese está na Tabela: 1

Tabela 2 – Comparação entre Métodos

réplica	**minha WER**	**WER do autor**
I	0.732	5
II	0.342	5
III	0.386	5
IV	0.388	5
V	0.428	5

Fonte: Tabela gerada pelo próprio autor.

Na primeira réplica do experimento obtemos a taxa de 0.732, na segunda 0.342, na terceira 0.386, na quarta 0.388, na quinta 0.428. Observamos que a variabilidade das nossas replicações do experimento apresentam uma alta variação se comparada com a do autor. Indicando uma performance do modelo melhor do que seu caso médio quando diante de entradas menores. No entanto, a variação de erros aumentou no caso aonde foi submetido à maior entrada, levando a hipotése de que o modelo piora levemente sua performance quanto maior a entrada. Por fim, no caso de entradas pequenas o modelo teve uma performance de altissíma, podendo ser facilmente utilizado para transcrições com um teor mais profissional.

5 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, utilizamos um modelo de *ASR* com a intenção de reconhecer a voz humana e transcrevê-la em texto. Foi selecionada uma métrica com a intenção de aferir a performance do modelo submetido a um *dataset* na língua portuguesa. No total 5 áudios foram testados e seus resultados coletados para análise, tendo um acurácia média de *WER* de 0.4.

Os resultados do nosso experimento apresentam uma baixa variação quando os resultados são comparados entre si, porém quando comparados aos resultados da *WER* com os resultados do autor(RADFORD *et al.*, 2022) no seu trabalho *Robust Speech Recognition via Large-Scale Weak Supervision*, podemos notar uma forte diferença positiva nos resultados do nosso experimento. Concluímos que o modelo é estável e performático para entradas pequenas.

Apontamos também que em todas as réplicas do nosso experimento nossa métrica superou a do autor autor(RADFORD *et al.*, 2022). Assim concluímos neste estudo de caso que o modelo proposto pelo autor, assim como o algoritmo proposto se adequa à transcrição de áudio para texto da fala humana em lingua portuguesa.

Como trabalhos futuros sugerimos a replicação do experimento elaborado neste texto numa quantidade suficiente para a realização de uma análise estatística visando a corroboração dos nossos resultados. Para tal, nosso código fica disponível no link *https://colab.research.google.com/drive/1325fFC4eBzb3a1VYHToawg4oFxQ4-4qK?usp=sharing*, sendo necessário autorização do autor. O dataset pode ser encontrado no link *http://www.openslr.org/100/*.

Por outro lado, como continuidade científica sugerimos a reprodução do algoritmo para tratar de áudios problemáticos, como aqueles contendo ruído e aqueles produzidos por pessoas com algum tipo de deficiência vocal.

O experimento também pode ser ampliado fazendo uso de todo o *dataset* sugerido, nesse caso sugerimos também que utiliza um versão mais robusta do *Colab* ou possua um *hardware* mais robusto para melhor execução do experimento.

Por fim, temos uma questão de pesquisa que amplia os resultados deste trabalho: Considerando que nossos resultados apontaram para uma direção de maior precisão conforme o tamanho do áudio diminui, nossa hipótese é que o tamanho do áudio impacta diretamente e negativamente na precisão do resultado conforme o tamanho da entrada cresce.

REFERÊNCIAS

ALLAMAR, J. **The Illustrated Transformer**. 2018. Disponível em: https://jalammar.github.io/illustrated-transformer/. Acesso em: 10 set. 2022.

BAEVSKI, A.; HSU, W.-N.; CONNEAU, A.; AULI, M. Unsupervised speech recognition. **Advances in Neural Information Processing Systems**, v. 34, p. 27826–27839, 2021.

BAEVSKI, A.; ZHOU, Y.; MOHAMED, A.; AULI, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. **Advances in Neural Information Processing Systems**, v. 33, p. 12449–12460, 2020.

BARLOW, H. B. Unsupervised learning. **Neural computation**, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 1, n. 3, p. 295–311, 1989.

BEYSOLOW, T. **Applied Natural Language Processing with Python: Implementing Machine Learning and Deep Learning Algorithms for Natural Language Processing**. [*S. l.*]: Apress, 2018.

CHEN, T.; XU, B.; ZHANG, C.; GUESTRIN, C. Training deep nets with sublinear memory cost. **arXiv preprint arXiv:1604.06174**, 2016.

DRIDI, S. Supervised learning-a systematic literature review. OSF Preprints, 2021.

FERREIRA, M. P. **Estudos de algoritmos de aprendizagem profunda no contexto de processamento de linguagem natural para desenvolvimento de assistentes virtuais**. Dissertação (B.S. thesis) – Universidade Federal do Rio Grande do Norte, 2022.

FRANÇOIS-LAVET, V.; HENDERSON, P.; ISLAM, R.; BELLEMARE, M. G.; PINEAU, J. *et al.* An introduction to deep reinforcement learning. **Foundations and Trends® in Machine Learning**, Now Publishers, Inc., v. 11, n. 3-4, p. 219–354, 2018.

GAIKWAD, S. K.; GAWALI, B. W.; YANNAWAR, P. A review on speech recognition technique. **International Journal of Computer Applications**, International Journal of Computer Applications, 244 5 th Avenue,# 1526, New . . . , v. 10, n. 3, p. 16–24, 2010.

GALVEZ, D.; DIAMOS, G.; CIRO, J.; CERÓN, J. F.; ACHORN, K.; GOPI, A.; KANTER, D.; LAM, M.; MAZUMDER, M.; REDDI, V. J. The people's speech: A large-scale diverse english speech recognition dataset for commercial usage. **arXiv preprint arXiv:2111.09344**, 2021.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **Unsupervised learning. The elements of statistical learning (pp. 485-585)**. [*S. l.*]: Springer, New York, NY, 2009.

JUANG, B.-H.; RABINER, L. R. Automatic speech recognition–a brief history of the technology development. **Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara**, v. 1, p. 67, 2005.

KUEHNE, H.; RICHARD, A.; GALL, J. Weakly supervised learning of actions from transcripts. **Computer Vision and Image Understanding**, Elsevier, v. 163, p. 78–89, 2017.

LU, X.; LI, S.; FUJIMOTO, M. Automatic speech recognition. In: **Speech-to-speech translation**. [*S. l.*]: Springer, 2020. p. 21–38.

MAHAJAN, D.; GIRSHICK, R.; RAMANATHAN, V.; HE, K.; PALURI, M.; LI, Y.; BHARAMBE, A.; MAATEN, L. V. D. Exploring the limits of weakly supervised pretraining. In: **Proceedings of the European conference on computer vision (ECCV)**. [*S. l.*: *s. n.*], 2018. p. 181–196.

MAHESH, B. Machine learning algorithms-a review. **International Journal of Science and Research (IJSR).[Internet]**, v. 9, p. 381–386, 2020.

MANNING, C.; SCHUTZE, H. **Foundations of statistical natural language processing**. [*S. l.*]: MIT press, 1999.

MARSLAND, S. **Machine learning: an algorithmic perspective**. [*S. l.*]: Chapman and Hall/CRC, 2011.

MEHYADIN, A. E.; ABDULAZEEZ, A. M. Classification based on semi-supervised learning: A review. **Iraqi Journal for Computers and Informatics**, v. 47, n. 1, p. 1–11, 2021.

NADKARNI, P. M.; OHNO-MACHADO, L.; CHAPMAN, W. W. Natural language processing: an introduction. **Journal of the American Medical Informatics Association**, BMJ Group BMA House, Tavistock Square, London, WC1H 9JR, v. 18, n. 5, p. 544–551, 2011.

NAGATA, K.; KATO, Y.; CHIBA, S. Spoken digit recognizer for the japanese language. **Journal of the Audio Engineering Society**, Audio Engineering Society, v. 12, n. 4, p. 336–342, 1964.

OGOTI, L. **Why Python is Good for Machine Learning**. 2021. Disponível em: https://www.section.io/engineering-education/why-python-is-good-for-machine-learning/. Acesso em: 20 ago. 2022.

RABINER, L.; JUANG, B.-H. **Fundamentals of speech recognition**. [*S. l.*]: Prentice-Hall, Inc., 1993.

RADFORD, A.; KIM, J. W.; XU, T.; BROCKMAN, G.; MCLEAVEY, C.; SUTSKEVER, I. Robust speech recognition via large-scale weak supervision. **arXiv preprint arXiv:2212.04356**, 2022.

RAJ, R. **Supervised, Unsupervised and Semi-supervised learning with Real-life Usecase**. 2021. Disponível em: https://www.enjoyalgorithms.com/blogs/supervised-unsupervised-and-semisupervised-learning. Acesso em: 25 ago. 2022.

RAPHAELL, B. **Desmistificando termos em Machine Learning - tipos de aprendizado**. 2021. Disponível em: https://www.alura.com.br/artigos/desmistificando-termos-machine-learning-tipos-aprendizado. Acesso em: 28 ago. 2022.

RASCHKA, S.; MIRJALILI, V. **Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2**. [*S. l.*]: Packt Publishing Ltd, 2019.

REDDY, Y.; VISWANATH, P.; REDDY, B. E. Semi-supervised learning: A brief review. **Int. J. Eng. Technol**, v. 7, n. 1.8, p. 81, 2018.

SAH, S. Machine learning: a review of learning types. Preprints, 2020.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, Ł.; POLOSUKHIN, I. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017.

YU, D.; DENG, L. **Automatic Speech Recognition: A Deep Learning Approach**. London: Springer, 2015. (Signals and Communication Technology). ISSN 1860-4862. ISBN 978-1-4471-5778-6.

ZHANG, Y.; PARK, D. S.; HAN, W.; QIN, J.; GULATI, A.; SHOR, J.; JANSEN, A.; XU, Y.; HUANG, Y.; WANG, S. *et al.* Bigssl: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition. **IEEE Journal of Selected Topics in Signal Processing**, IEEE, v. 16, n. 6, p. 1519–1532, 2022.

ZHOU, Z.-H. A brief introduction to weakly supervised learning. **National science review**, Oxford University Press, v. 5, n. 1, p. 44–53, 2018.

APÊNDICE A – CÓDIGOS-FONTE BASE UTILIZADO PARA OS EXPERIMENTOS

Código-fonte 1 – Na primeira linha instalamos Whisper!

```
!pip install git+https://github.com/openai/whisper.git
```

Código-fonte 2 – Fazemos o download do dataset e descompactamos os arquivos!

```
# Get test data

!wget -N https://www.openslr.org/resources/100/mtedx_pt.tgz
!tar -xvzf mtedx_pt.tgz
```

Código-fonte 3 – Agora o Whisper irá transcrever o áudio!

```
import os
!pip install beautifulsoup4

!whisper "/content/sample_data/_-BaBl6qoK0.flac" --model
   large --language Portuguese
```

Código-fonte 4 – Agora iremos ler o arquivo de texto que o Whisper gerou. Algumas linhas descrevem o que foi gravado, iremos retirar essas linhas.!

```
with open('/content/_-BaBl6qoK0.flac.txt','r') as f:
  whisper_lines1 = [l.strip() for l in f]




whisper_lines1 = whisper_lines1[4:]
whisper_lines1 = whisper_lines1[:-2]
```

Código-fonte 5 – Com os arquivos originais que fizemos download previamente, precisaremos ler eles para posterior comparação:!

```
with open('/content/sample_data/_-BaBl6qoK0.pt.vtt','r') as
    f:
  true_lines1 = [l.strip() for l in f]
```

Código-fonte 6 – Text Cleaning!

```
whisper_text1 = " ".join(whisper_lines1)
whisper_text1 = whisper_text1.replace('"','')
whisper_text1 = whisper_text1.lower()
```

Código-fonte 7 – Tudo pronto, vamos instalar e calcular o WER!

```
!pip install jiwer

from jiwer import wer
print("audio _-BaBl6qoK0", wer(true_text1,whisper_text1))
```