Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Reconhecimento automático de fala aplicada ao controle de tráfego aéreo

Matheus F. Castro

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientador
Prof. Dr. Li Weigang

Brasília
2023

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Reconhecimento automático de fala aplicada ao controle de tráfego aéreo

Matheus F. Castro

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Prof. Dr. Li Weigang (Orientador)
CIC/UnB

Prof.a Dr.a Alba Cristina Magalhaes Alves de Melo
CIC/IE, Universidade de Brasília

Dr. João Luiz Azevedo de Carvalho
ENE/FT, Universidade de Brasília

Prof. Dr. João Luiz Azevedo de Carvalho
Coordenador do Curso de Engenharia da Computação

Brasília, 17 de Janeiro de 2023

Dedicatória

Dedico este trabalho à minha família e amigos que me apoiaram.

Agradecimentos

À Deus por me permitir realizar este trabalho.

Aos meus pais Daise Feitosa de Castro e Jehu Alves de Castro por sempre me apoiarem e incentivarem em minha carreira acadêmica.

À minha avó Nair por sua gentileza e carinho. À minha irmã Camila, por sua inspiração e companheirismo. À Mariana por estar ao meu lado e me apoiar nos momentos difíceis. Ao meu cunhado Rodrigo pelo estímulo a continuar estudando. Aos meus amigos Rodrigo Demetrio e Rodrigo Xavier por me acompanharem nesta jornada.

Ao meu orientador Prof. Dr. Li Weigang por me orientar e guiar nas decisões deste trabalho.

Aos professores da banca Prof. Dr.a Alba Cristina Magalhaes Alves de Melo e Prof. Dr. João Luiz Azevedo de Carvalho pela revisão e aconselhamentos no trabalho.

Ao Cristiano Garcia e Lucas Monteiro por me ajudarem e apoiarem no desenvolvimento do trabalho.

À UnB pela infraestrutura oferecida.

Resumo

A fala é o principal meio de comunicação entre as pessoas, e um importante pilar da aviação atual na comunicação entre controlador de tráfego aéreo e piloto. Por meio da comunicação falada, um controlador informa ao piloto pistas de pouso, decolagem, realiza o controle da altitude e realiza comunicações de emergência. Por isso, novas tecnologias para se evitarem ruídos de comunicação entre controlador e piloto são essenciais para a aviação. Existem diversos exemplos reais onde problemas de comunicação geraram acidentes catastróficos, como a colisão de Charki Dadri, que gerou 347 fatalidades, e o desastre aéreo de Tenerife que causou 583 fatalidades. Por esse motivo, este trabalho implementa e avalia um sistema de reconhecimento automático de fala que possa ser utilizado em sistemas de controle de tráfego aéreo. No trabalho foi utilizado o modelo Whisper, um modelo *sequence-to-sequence*, baseado no modelo *encoder-decoder* em transformer, pré-treinado em várias configurações, e então realizado aprendizado por transferência em cima desses modelos pré-treinados para terem suas performances e taxas de erro avaliadas. A base de dados utilizada possui aproximadamente 10 horas de áudio falado e foi dividida em aproximadamente 6 horas de treino, 2 horas de validação e 2 horas de teste. Com uma base de treino tão pequena, seria de se esperar que o resultado do aprendizado por transferência fosse pequeno, porém nos testes realizados, apesar da base pequena, foi possível obter melhoria de até 25% na *word error rate* (WER).

Palavras-chave: reconhecimento automático de fala, controle de tráfego aéreo, transformer, modelo Whisper, deep learning, aprendizado por transferência

Abstract

Speech is the primary method of communication in society and an important pillar of current aviation as it is used in the communication between air traffic controllers and pilots. Through spoken communication, a controller informs the pilot of landing and takeoff tracks, performs altitude control, and performs emergency communications. Therefore, new technologies to avoid communication noise between driver and pilot are essential for aviation. There are several examples where communication problems led to catastrophic accidents such as the Charki Dadri collision, which led to 347 fatalities, and the Tenerife airport disaster which caused 583 fatalities. For this reason, this work implements and evaluates an automatic speech recognition system that can be used in air traffic control systems. The Whisper, a sequence-to-sequence model, based on the encoder-decoder model of the transformer, was used in several pre-trained configurations, and then fine-tuned and adjusted to have their performances and error rating rates evaluated. The database used has approximately 10 hours of spoken audio and was divided into approximately 6 hours of training, 2 hours of validation, and 2 hours of testing. With such a small training base, it would be expected that the fine-tuning improvement would be small, but despite the small base, it was possible to obtain an improvement of up to 25% in the word error rate (WER).

Keywords: automatic speech recognition, air traffic control, Transformer, Whisper, deep learning, transfer learning

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A fala é o principal meio de comunicação entre as pessoas. Por razões que vão desde a curiosidade tecnológica sobre os mecanismos para a realização mecânica das capacidades de fala humana, até o desejo de automatizar tarefas simples inerentemente exigindo interações homem-máquina, a pesquisa em reconhecimento automático de fala e síntese de fala por máquina tem atraído muita atenção recentemente [2].

O reconhecimento automático de fala (ASR, da sigla em inglês para *automatic speech recognition*), que visa permitir a interação natural homem-máquina, tem sido uma área de pesquisa intensiva há décadas. Muitas tecnologias centrais como modelos de mistura Gaussiana (GMM, da sigla em Inglês para *Gaussian mixture models*) e modelos mcultos de Markov (HMM, da sigla em inglês para *hidden Markov models*) foram desenvolvidas ao longo do caminho, principalmente antes do novo milênio. Essas técnicas avançaram muito no estado da arte em ASR e em seus campos relacionados. Comparado a essas conquistas anteriores, o avanço na pesquisa e aplicação de ASR na década anterior a 2010 foi relativamente lento e menos empolgante, embora técnicas importantes, como o treinamento discriminativo de sequência GMM-HMM, tenham funcionado bem em sistemas práticos durante esse período [3].

As técnicas de aprendizado profundo estimularam o surgimento de uma alternativa, que é o modelo de ponta-a-ponta (vide seção 2.3.5). Comparado com o modelo baseado em HMM, o modelo de ponta-a-ponta usa um único modelo para mapear diretamente o áudio para caracteres ou palavras. Ele substitui o processo de engenharia pelo processo de aprendizado e não precisa de conhecimento de domínio, portanto, o modelo de ponta-a-ponta é mais simples para construção e treinamento. Essas vantagens fazem com que o modelo de ponta-a-ponta se torne rapidamente uma área de pesquisa importante em reconhecimento de fala contínua de grande vocabulário [4].

Os métodos de aprendizado profundo, por sua vez, viram um crescimento explosivo, atenção e disponibilidade de ferramentas após seu sucesso em visão computacional na

década que iniciou em 2010. O processamento de linguagem natural logo experimentou muitos dos benefícios da visão computacional. O reconhecimento de fala, tradicionalmente um campo dominado pela engenharia de recursos e técnicas de ajuste de modelo, incorporou aprendizado profundo em seus métodos de extração de recursos, resultando em fortes ganhos de qualidade. A grande disponibilidade de dados é outro fator que contribui para os ganhos de desempenho com o aprendizado profundo. Ao contrário de muitos algoritmos de aprendizado tradicionais, os modelos de aprendizado profundo continuam a melhorar com a quantidade de dados fornecidos [5].

No controle de tráfego aéreo (ATC, da sigla em inglês para *air traffic control*), as comunicações trocadas entre um piloto e um controlador incluem uma riqueza de informações de contexto situacional, ou seja, são fortemente influenciadas pela dinâmica de tráfego em tempo real. Esse é um aspecto crucial na tomada de decisões eficientes em relação às operações de tráfego aéreo. Acredita-se que entender a intenção de controle incorporada pelo discurso do ATC é a chave para garantir a segurança do voo. Entre as possíveis aplicações das técnicas de ASR para traduzir falas ATC estão a redução erros de comunicação, a melhoria da eficiência operacional e o alívio na carga de trabalho dos controladores [6].

Devido à comunicação ser um aspecto tão importante na aviação atual, novas tecnologias são constantemente introduzidas ao cenário da aviação afim de minar as possibilidades de ruídos de comunicação. Uma destas tecnologias é o *Controller Pilot Data Link Communications* (CPDLC) [7], tecnologia já usada em alguns locais do país, onde são trocadas mensagens não urgentes entre controlador e piloto como uma alternativa à comunicação de voz. Nesse caso os controladores possuem a capacidade de enviar comandos para alterações de altitude, de velocidade, alteração de frequência de rádio e outros.

1.1 Objetivo

O objetivo deste trabalho é o desenvolvimento de um sistema de reconhecimento automático de fala especializado no cenário de controle de tráfego aéreo. Este sistema utilizará aprendizado por transferência (vide seção 2.4) para aprimorar um modelo pré-treinado de reconhecimento de fala e, então, avaliar e comparar a performance do modelo nos cenários descritos.

1.2 Apresentação do Manuscrito

O Capítulo 2 introduz o leitor aos conceitos e algoritmos utilizados no trabalho e provê uma breve visão sobre o histórico de sistemas de reconhecimento de fala. No Capítulo 3

são abordados trabalhos relacionados que foram utilizados para inspirar este trabalho. O Capítulo 4 apresenta a definição do problema, o conjunto de dados, pré-processamento, as métricas e metodologia do trabalho. No Capítulo 5 são apresentados os resultados obtidos e a análise destes. E por fim, o Capítulo 6 realiza a conclusão do trabalho e cita trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta conceitos utilizados no trabalho, como a inteligência artificial de modo geral, modelos inteligentes, processamento de linguagem natural e reconhecimento automático de fala.

2.1 Inteligência Artificial

Inteligência artificial, ou IA, pode possuir várias definições diferentes com relação a diferentes campos do qual é estudado e pode ser analisada de diversos pontos de vista diferentes, como pela visão matemática, ou visão psicológica ou neurológica. Por isto, a IA também pode ser analisada de ângulos diferentes, em favor da comparação entre a inteligência humana e um programa de computador; pode ser analisada em termos de racionalidade, raciocínio ou comportamento inteligente.

Em 2004, o professor John Maccarty promoveu a seguinte definição para inteligência artificial em tradução literal: "*É a ciência e engenharia de tomar decisões inteligentes, especialmente, programas de computador inteligentes. É relacionado com a tarefa similar de usar computadores para entender a inteligência humana, mas IA não precisa se limitar a métodos que são biologicamente observáveis.*" [8]. De modo geral, fornecer uma definição precisa para IA é uma tarefa árdua, porém valiosa [9].

2.2 O Processamento de Linguagem Natural

O processamento de linguagem natural (NLP sigla em inglês para *natural language processing*), é uma subárea da inteligência artificial que se concentra em permitir que as máquinas compreendam, interpretem e gerem linguagem humana de forma eficiente. O objetivo principal do NLP é desenvolver tecnologias que possam compreender a linguagem

natural em todas as suas nuances e complexidades, incluindo contexto, intenção, sentimento, entre outros aspectos. É considerada um dos grandes desafios da IA. Considere a seguinte frase: "Eu não disse que eu bati nele". Essa é uma frase simples, porém, a entonação em diferentes palavras pode resultar em sentidos completamente diferentes, como abaixo:

- "**Eu** não disse que eu bati nele";
- "Eu **não** disse que eu bati nele";
- "Eu não **disse** que eu bati nele";
- "Eu não disse que **eu** bati nele";
- "Eu não disse que eu **bati** nele";
- "Eu não disse que eu bati **nele**".

No primeiro exemplo, podemos inferir que o comunicador nega ter sido ele ao dizer que bateu no sujeito, já no último exemplo, podemos inferir que o comunicador afirma ter dito que bateu em alguém, porém, não no sujeito em questão.

O ser humano utiliza a linguagem falada há milênios, e possui uma linguagem de comunicação complexa e diversa com inúmeras nuances. Mesmo para um mesmo idioma, podem existir diversos sotaques, dialetos e gírias diferentes. No exemplo acima, podemos começar a perceber o motivo do desafio do processamento de linguagem natural.

Ao longo do presente capítulo, abordaremos mais a fundo os conceitos de NLP.

2.3 Reconhecimento Automático de Fala

Reconhecimento automático de fala (ASR sigla em inglês para *automatic speech recognition*) é um subcampo da inteligência artificial e da NLP e lida com sinais de áudio e texto. Sua tarefa é mapear um sinal acústico contendo uma língua natural falada em uma sequência de palavras ditas por alguém.

De forma matemática, isso implica em dizer que a tarefa do ASR é criar uma função que calcula a maior probabilidade de sequência linguistica y para uma sequência acústica X:

$$f^*_{ASR}(X) = \arg_y \max P^*(y|X = X)$$

em que P^* é a distribuição condicional que relaciona a sequência acústica X para a sequência linguística y.

Por exemplo, na Figura 2.1 temos a gravação de áudio sm2_06_085 obtida da base de dados ATCOSIM [1], utilizada em nosso projeto. Para este caso, o objetivo do ASR seria converter o sinal de áudio da figura em seu correspondente fonético, transcrito como: "*thank you*" pela base de dados.

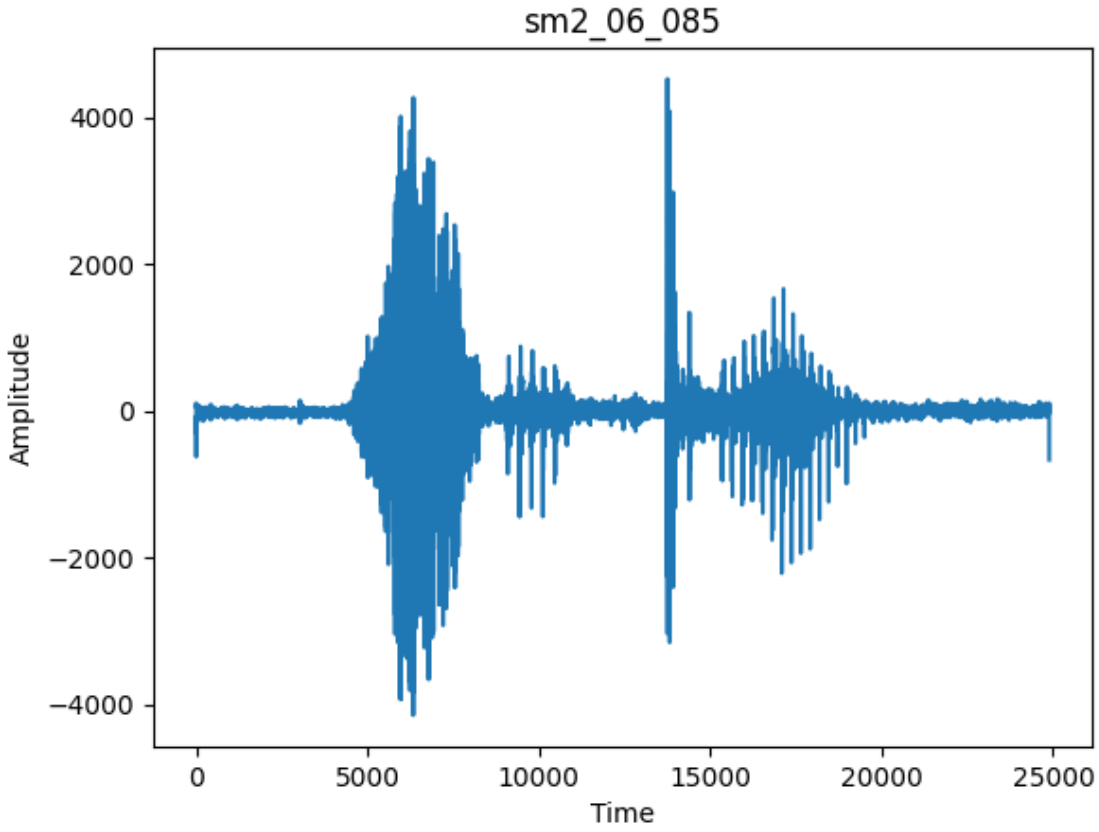

Figura 2.1: Amplitude × Tempo: sequência acústica.

Um sistema ASR se mostra especialmente difícil por sofrer influência de diversos fatores, como sotaques e dialetos. O português, por exemplo, possui dialetos originários da América do Sul, da Europa e da África, e com sotaques diferentes. Por esses motivos, o desenvolvimento de um ASR que possa identificar uma fala, apesar dos dialetos e sotaques, mostra-se uma tarefa árdua.

2.3.1 Modelos de Mistura Gaussiana

O modelo de mistura Gaussiana (GMM, da sigla em inglês para Gaussian mixture models) é um modelo probabilístico representado como a soma ponderada de componentes Gaussianos [10]. O GMM é capaz de capturar a estrutura subjacente dos dados ao iden-

tificar várias distribuições Gaussianas e suas respectivas probabilidades de ocorrência; em outras palavras, é capaz de identificar sub-grupos dentro de um grupo maior.

No exemplo da Figura 2.2 podemos ver o resultado da aplicação da modelagem do GMM para um modelo de dados hipotético.

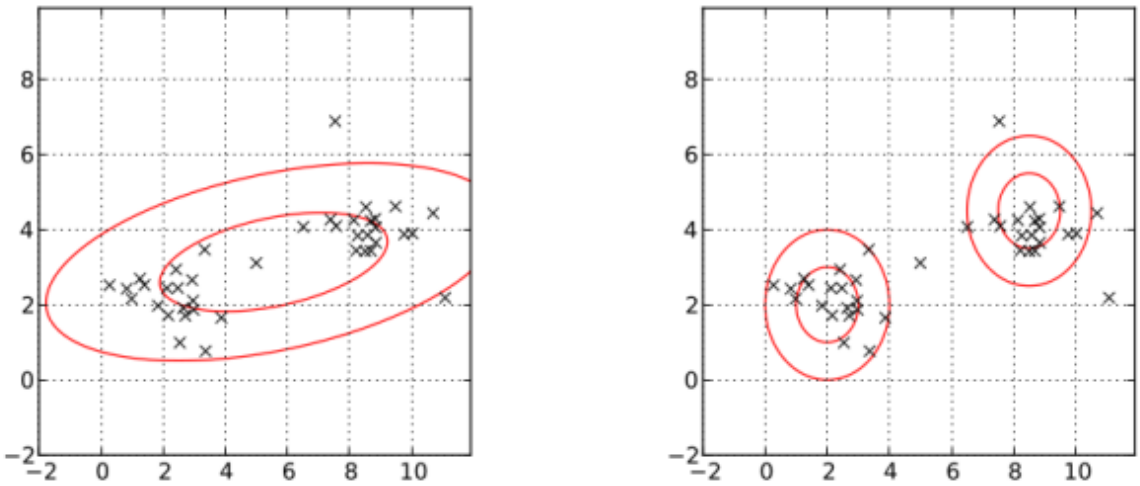

Figura 2.2: Distribuição Gaussiana (esquerda) e modelagem com GMM (direita) (Fonte: [11]).

2.3.2 Modelos Ocultos de Markov

Modelo oculto de Markov (HMM, da sigla em Inglês para *hidden Markov model*) é também um modelo probabilístico, onde este é um modelo finito que descreve a provabilidade de distribuição sobre um número finito de possíveis sequências [12]. Apesar de possuir diversos problemas, como a dificuldade de se obter robustês para bases de dados variadas, este modelo era indicado para tarefas de reconhecimento no século passado [13].

O modelo HMM ainda é amplamente utilizado para a comparação de sequências de DNA genômico [14]. Quando alimentado com uma determinada sequência de entradas, como palavras, o HMM processa seu funcionamento calculando as probabilidades de transição entre seus estados ocultos. Um HMM pode ser representado como um grafo que contém distribuições de probabilidade associadas aos seus nós, permitindo que o modelo capture a dinâmica do sistema subjacente e faça comparações precisas.

2.3.3 GMM-HMM

A junção dos modelos GMM e HMM foram o estado da arte no cenário do ASR por várias décadas [15] no século passado e por diversos anos deste século. Neste modelo de ASR,

que combina os modelos GMM e HMM, o modelo Gaussiano, modela a associação entre características acústicas e fonemas, enquanto o modelo de Markov gera uma sequência de fonemas e subfonemas.

No geral esse modelo de ASR possui alguns módulos, sendo os principais:

- *Feature extraction*: onde diversas características de uma sequência acústica são obtidas;
- *Acoustic model*: é o componente principal do ASR e é usado para ligar as features do sinal de áudio com a hipótese da sentença;
- *Lexical model*: componente para identificar a pronunciação da palavra, é particularmente útil no desenvolvimento de modelos para falantes não nativos de uma língua;
- *Recognition*: que deve realizar a identificação da sentença.

Pela dificuldade de se construir sistemas ASR e o esforço usado na construção de sistemas GMM-HMM, as pesquisas com redes neurais só avançaram após o final dos anos 2000 [15]. Com o aumento de datasets e da capacidade computacional para uso de modelos mais profundos, as redes neurais foram substituindo o modelo Gaussiano nos modelos tradicionais, o que trouxe uma melhoria de performance de cerca de 30% em poucos anos [15]. Isso gerou uma rápida guinada de sistemas ASR para o aprendizado profundo.

2.3.4 Aprendizado Profundo

O aprendizado profundo (do inglês *deep learning*) é amplamente usado no campo da inteligência artificial, especificamente em cenários de ASR. Miao et al. [16] por exemplo, utilizam o aprendizado profundo em um ASR robusto para sotaques.

O trabalho de Mohamed et al. [17] foi um dos pioneiros em obter um ASR no estado da arte usando aprendizado profundo. Isto ocorre devido ao fato de redes de aprendizado profundo necessitarem de grandes bases de dados e poder computacional para obterem uma boa performance, e Mohamed et al. [17] obtiveram isto em 2009 para o reconhecimento de números de telefone, e a redução de erros de fonemas de cerca de 26% para 20,7% na base de dados TIMIT.

2.3.5 ASR Ponta-a-Ponta

Modelos ASR ponta-a-ponta (do inglês *end-to-end*) são no geral, mais simples que os modelos da Seção 2.3.4. O GMM-HMM, é um modelo complexo composto de diversas etapas intermediárias, e com o avanço do aprendizado profundo, e com o crescente número

de bases de dados, o modelo ponta-a-ponta tem se tornado mais popular. Neste modelo, ao invés de existirem diversas etapas, uma rede neural profunda é treinada para traduzir diretamente uma sequência de áudio em uma sequência de texto [18]. Este modelo descarta todos os módulos existentes em sistemas de ASR tradicionais que foram usados por décadas.

Existem diversos modelos ponta-a-ponta no estado da arte. Como o *Attention based Encoder-Decoder* [19], *Recurrent Neural Network Transducer* [20] e modelos desenvolvidos por grandes empresas, como o Jasper [21], modelo desenvolvido pela Nvidia.

O trabalho de Graves et al. [22] foi também pioneiro em redes ponta-a-ponta, removendo completamente o modelo HMM de um sistema ASR. Graves et al. utilizando a mesma base de dados TIMIT usada por Mohamed et al. [17] obtiveram redução na taxa de erros e alcançou um novo estado da arte em 17,7%.

Um outro avanço para sistemas ASR ponta-a-ponta, é permitir que o sistema aprenda a alinhar informações acústicas com informações fonéticas [23].

2.3.6 Modelo Transformer

O modelo Transformer [24] tem sido usado com sucesso em muitas tarefas de transformação de *sequence-to-sequence* [25]. Este usa um mecanismo de auto-atenção que pode modelar longas distâncias de contexto sem uma dependência sequencial [26]. A arquitetura do Transformer, ilustrada na Figura 2.3, é composta por *encoders*, *decoders* e camadas de atenção. O modelo original do Transformer é uma pilha de 6 camadas. A saída de uma camada x é o input de uma camada $x + 1$ até o final da pilha. São 6 camadas de *encoders* e 6 camadas de *decoders*.

O *encoder*, módulo à esquerda da Figura 2.3, possui em cada camada duas sub-camadas, um mecanismo de auto atenção, e uma rede *feed-forward* . O papel do *encoder* pode ser simplificado a mapear uma sequência de entradas a uma sequência de representação contínua. Alguns modelos baseados apenas em *encoders* tem bom desempenho em tarefas que requerem o entendimento de uma sentença por completo. O BERT [27] é um exemplo de modelo baseado em *encoders*.

O *decoder*, módulo à direita da Figura 2.3, possui três sub-camadas, em que duas são as mesmas do encoder, porém com uma camada de atenção a mais sobre a saída da pilha de *encoders*. O pré-treino de um *decoder*, usualmente envolve prever a próxima palavra de uma sentença. Modelos baseados em *decoders* possuem grande performance em tarefas que envolvem geração de texto, como os modelos GPT [28].

O modelo de atenção executa o produto do vetor de palavras e determina os maiores relacionamentos entre todas as outras palavras, incluindo ela mesma. O mecanismo de atenção provê um relacionamento mais profundo entre as palavras, produzindo melhores

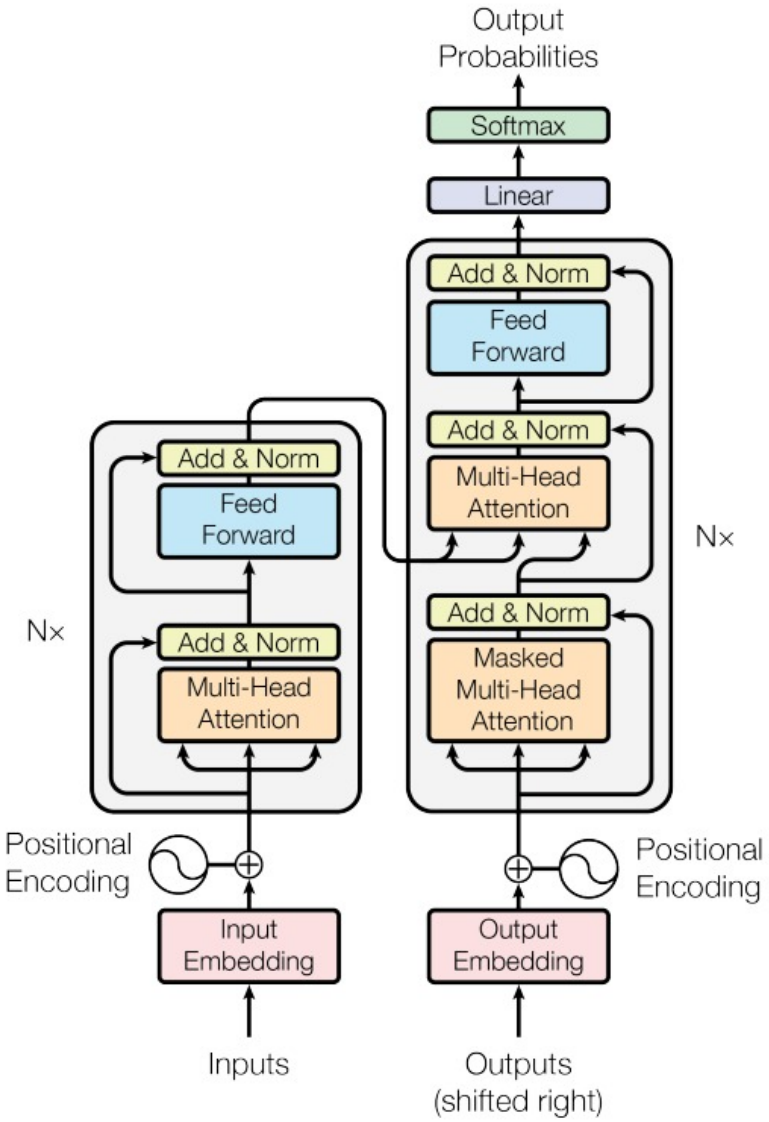

Figura 2.3: Arquitetura Transformer (Fonte: [24]).

resultados. No exemplo da Figura 2.4 , a partir da frase: "*The animal didn't cross the street because it was too tired*", o mecanismo de atenção relaciona as palavras mais provavéis de serem relacionadas com a palavra "*it*", que no caso são as palavras "*the*" e "*animal*".

2.3.7 Whisper

O Whisper [29], desenvolvido pela OpenAI, mesma criadora do recente ChatGPT, desenvolveu este modelo de ASR treinado em 680 mil horas de dados supervisionados multilinguais e multitarefa coletados da internet. Sua base de dados é diversa e leva a robustês a

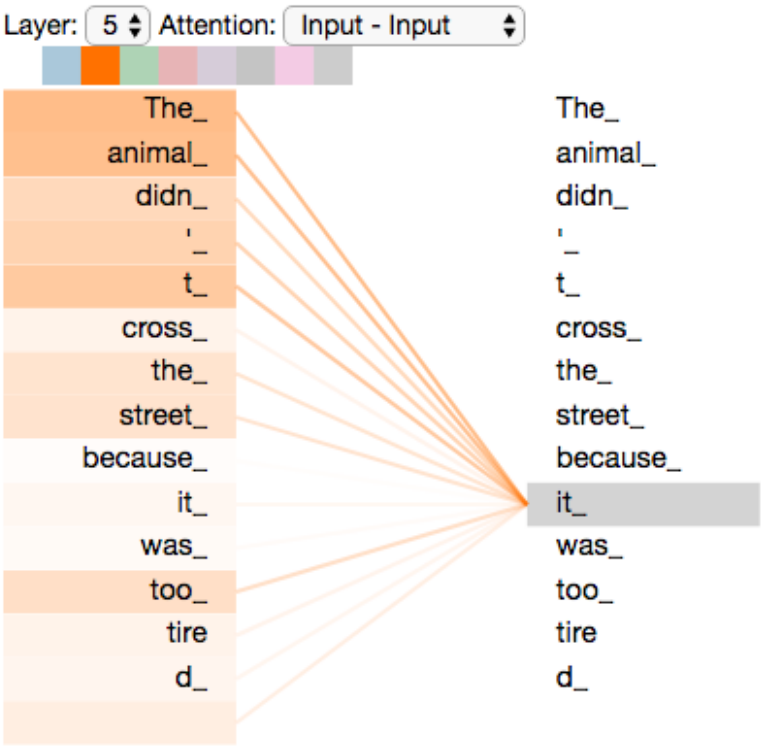

Figura 2.4: Módulo atenção Transformer (Fonte: Tensor2Tensor coolab notebook).

sotaques e barulho. Ele permite transcrição de áudio para texto em múltiplas línguas e tradução destas para o inglês.

O Whisper é um ASR ponta-a-ponta implementado como um *encoder-decoder* Transformer (Figura 2.5). O áudio é dividido em blocos de 30 segundos, convertido em espectograma no domínio da frequência e então disponibilizado à pilha de *encoders*. Este então, envia o resultado a pilha de *decoders* que prevê os *tokens* de texto, tendo conhecimento dos *tokens* recebidos e estados dos *encoders*.

O modelo *encoder-decoder* Transformer também é conhecido como modelo *sequence-to-sequence*, onde o *encoder* transforma a entrada de áudio em uma lista de estados, e extrai características importantes do áudio. O *decoder* realiza o papel de modelo de língua, ele processa os estados do *encoder* e gera o texto correspondente às transcrições.

O Whisper é pré-treinado e especializado utilizando o médoto de *cross-entropy*[31]. O modelo é treinado para classificar corretamente os *tokens* de texto a partir de um vocabulário pré-definido de *tokens* de texto.

Na Tabela 2.1 podemos ver os modelos pré-treinados disponíveis do Whisper e sua quantidade de parâmetros. Destes modelos, apenas o large não possui um modelo específico para inglês. Neste trabalho, usamos os modelos small, medium, medium.en e

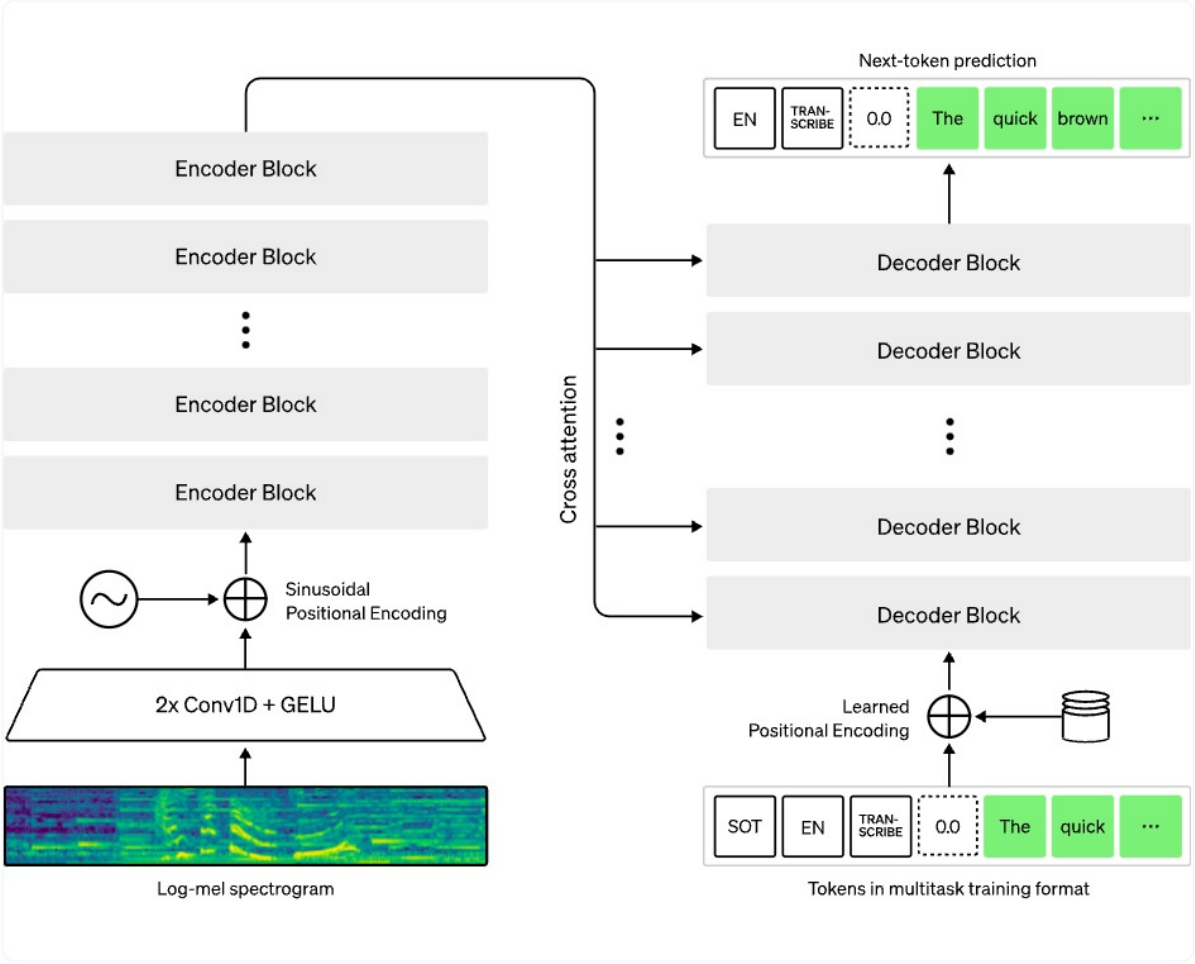

Figura 2.5: Arquitetura Whisper (Fonte: [30]).

large.

Na Figura 2.6 podemos observar a performance do modelo em diversas línguas, particularmente no inglês e português, o inglês é a língua do dataset utilizado no trabalho, porém, a WER (sigla em inglês para word error rate ou taxa de erro por palavra) no modelo, próxima a do português, apesar de não ser utilizado no trabalho, abre as portas para o uso de sistemas ASR eficientes em português, e em outros idiomas.

2.4 Aprendizado por Transferência

O aprendizado por transferência [32] (do inglês *transfer learning*) se refere ao treinamento de modelos de inteligência artificial de diferentes modelos. Esta prática é útil quando há dificuldade de se obter dados do domínio estudado. Neste caso, pode-se utilizar um

Modelo	Parâmetros	Apenas inglês
tiny	244M	tiny.en
base	244M	base.en
small	244M	small.en
medium	769M	medium.en
large	1550M	N/A

Tabela 2.1: Modelos pré-treinados por métricas.

modelo pré-treinado em outros domínios e realizar o aprendizado por transferência para o domínio específico.

Em outras palavras o aprendizado por transferência ocorre quando se usa um conhecimento que foi adquirido anteriormente resolvendo um problema, e então é aplicado para um novo (porém relacionado) problema.

2.4.1 Ajuste Fino

O ajuste fino (do inglês *fine-tuning*), é uma técnica para aprimorar o desempenho de um modelo já treinado em uma tarefa específica, utilizando um conjunto de dados adicional que é similar, mas diferente do conjunto de dados original.

O processo de ajuste fino envolve congelar todas ou a maioria das camadas do modelo pré-treinado e, em seguida, treinar as camadas restantes utilizando o novo conjunto de dados. Dessa forma, o modelo pode ajustar suas respostas ao novo conjunto de dados, sem precisar ser retreinado completamente. Em resumo, significa utilizar os pesos de um modelo já treinado, como valores iniciais no treinamento de uma nova rede.

No trabalho, a especialização de base foi feita por meio do processo de ajuste fino.

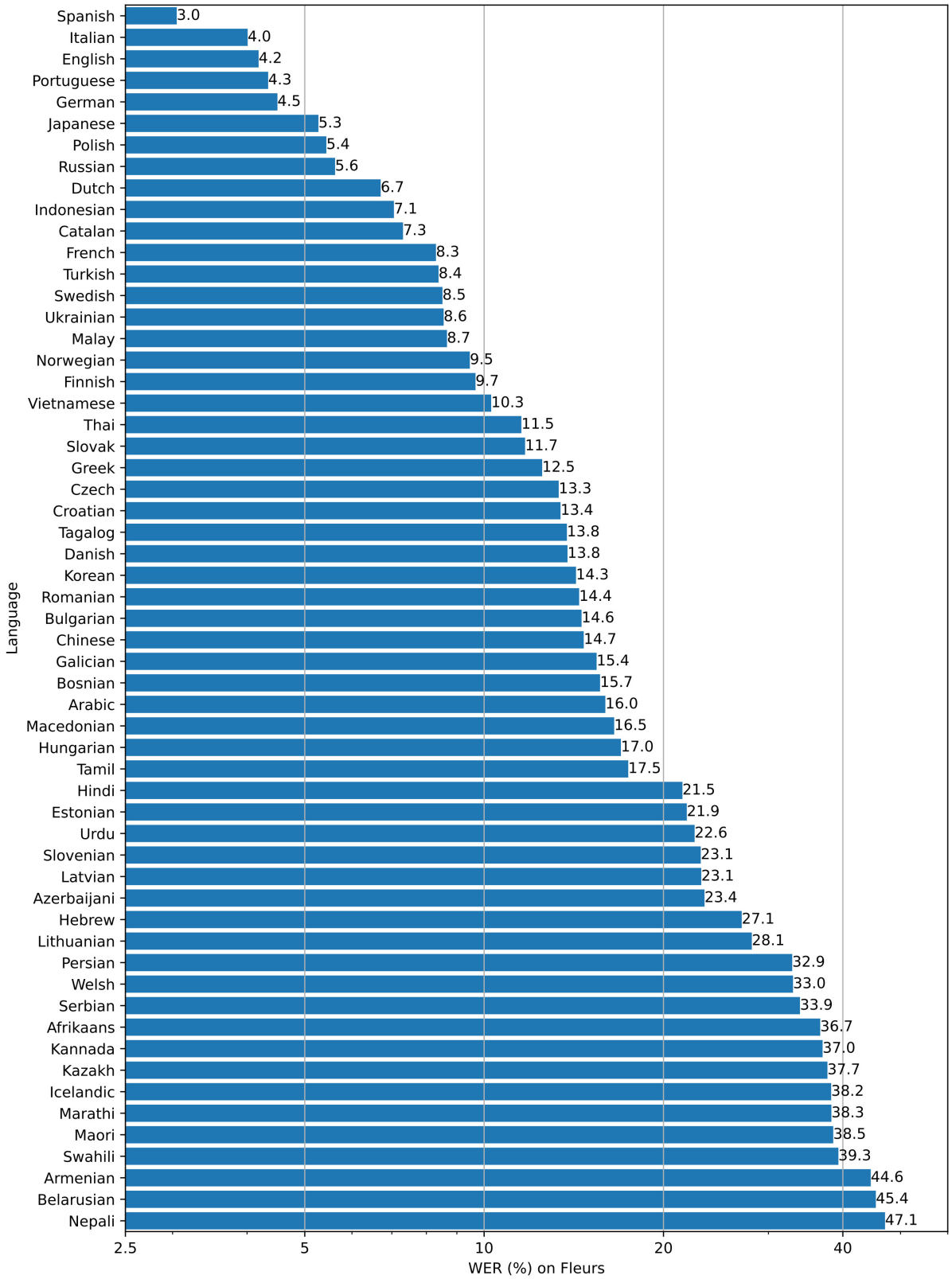

Figura 2.6: Taxa de erro por palavra do Whisper para diferentes línguas.

Capítulo 3

Revisão Bibliográfica

Este trabalho tem como campo de pesquisa o reconhecimento automático de fala (ASR) em tráfego aéreo. Assim, serão apresentados abaixo os trabalhos mais significativos neste campo, bem como os trabalhos mais recentes que apresentam resultados promissores.

3.1 Reconhecimento de Fala

Ghai et al. [33] revisam as técnicas, modelos, módulos, acurácia, velocidade, robustez e áreas de aplicação de um sistema ASR. Apesar de ser um artigo de 2012, e não envolver sistemas ASR ponta-a-ponta, traz informações relevantes ao estudo dos modelos de ASR. Os autores citam diversas áreas de aplicação de um sistema ASR como ligações automáticas, e sistemas de vendas e validações de cartões de crédito. Os autores também informam o grande potencial da união de redes neurais artificiais com o modelo HMM em cenários de grande vocabulário. Os autores chegam a conclusão que para o desenvolvimento de um sistema ASR os seguintes passos são necessários:

1. obtenção da base de dados;
2. parametrização do sinal utlizando técnicas de extração de sinal;
3. análise acústica, transformando os sinais em vetores de coeficientes;
4. definição dos modelos a serem utilizados;
5. treinamento dos modelos;
6. definição das tarefas a serem realizadas;
7. reconhecimento do sinal de entrada;

Li et al. [34] revisam os avanços de sistemas ponta-a-ponta no campo do reconhecimento automático de fala, e observam a tendência de mudança de sistemas tradicionais

para sistemas ponta-a-ponta. Os autores introduzem técnicas de ASR ponta-a-ponta populares e apresentam artigos recentes que obtiveram ótimos resultados no campo, como o Deep Speech 2 [35] e o w2v-BERT [36]. Por fim, os autores concluem que modelos ponta-a-ponta estão se tornando os modelos dominantes de ASR, mas que ainda possuem desafios pela frente, como a dificuldade de incluir conhecimento em um único modelo ponta-a-ponta.

Benzeghiba et. al. [37] apresentam as características e os avanços das principais variações do sinal de fala que dificultam a tarefa do reconhecimento automático de fala, tais como: sotaque estrangeiro e regional; fisiologia; estilo de fala e fala espontânea; ritmo da fala; fala infantil; e estado emocional. Dentre as principais metodologias que melhoram a robustez e a acurácia da modelagem e análise de ASR, destacam-se: compensação e invariância; sugestões adicionais e fluxos de recursos; redução de dimensionalidade e seleção de recursos; adaptação; modelagem múltipla; recursos acústicos auxiliares; técnicas de modelagem de pronúncia; e bases de treinamento maiores e diversificadas.

Huang et. al. [38] focam na perspectiva histórica do reconhecimento de fala, e nos avanços tecnológicos observados nas últimas quatro décadas, que levaram à solução de tarefas antes impossíveis. Em 1976, o poder computacional disponível era adequado apenas para realizar o reconhecimento de fala em tarefas altamente restritas com baixos fatores de ramificação (perplexidade). Hoje, somos capazes de lidar com vocabulários quase ilimitados com fatores de ramificação muito maiores. O aprendizado básico e os algoritmos de decodificação não mudaram substancialmente em 40 anos. Porém, muitas melhorias algorítmicas foram observadas, como usar algoritmos distribuídos para o aprendizado profundo. Lidar com palavras anteriormente desconhecidas continua a ser um problema para a maioria dos sistemas. A coleta de vocabulários muito grandes com base em perfis baseados na web torna provável que o usuário quase sempre use uma das palavras conhecidas. Os mecanismos de pesquisa da Web de hoje armazenam mais de 500 milhões de entradas de entidades, o que pode ser poderoso para aumentar o vocabulário que normalmente é muito menor para reconhecimento de fala.

Baevski et al. [39] desenvolveram um modelo de ASR ponta-a-ponta treinado apenas com dados não supervisionados. Os autores obtiveram uma melhoria no estado da arte na WER de 26,1% para 11,3% em modelos não supervisionados para a base de dados TIMIT, e uma WER de 5,9% no maior *benchmark* da base de dados Librispeech.

3.2 Inteligência Artificial na Aviação

Nesta seção analisaremos alguns artigos que abordam o mesmo tema trabalhado neste projeto.

Helmke et. al. [40] desenvolveram um sistema de assistência baseado no reconhecimento de fala para controladores de voo. O sistema, conforme Figura 3.1, inicia-se na fala de um controlador de voo. O sistema realiza o reconhecimento de fala, obtém a frase dita pelo controlador e realiza uma busca dos possíveis comandos disponíveis mais próximos à fala do controlador. Por fim, imprime a lista de prováveis comandos para o controlador, que irá selecionar o comando desejado. O autor informa que o sistema foi capaz de reduzir muito a carga em cima dos controladores de voo.

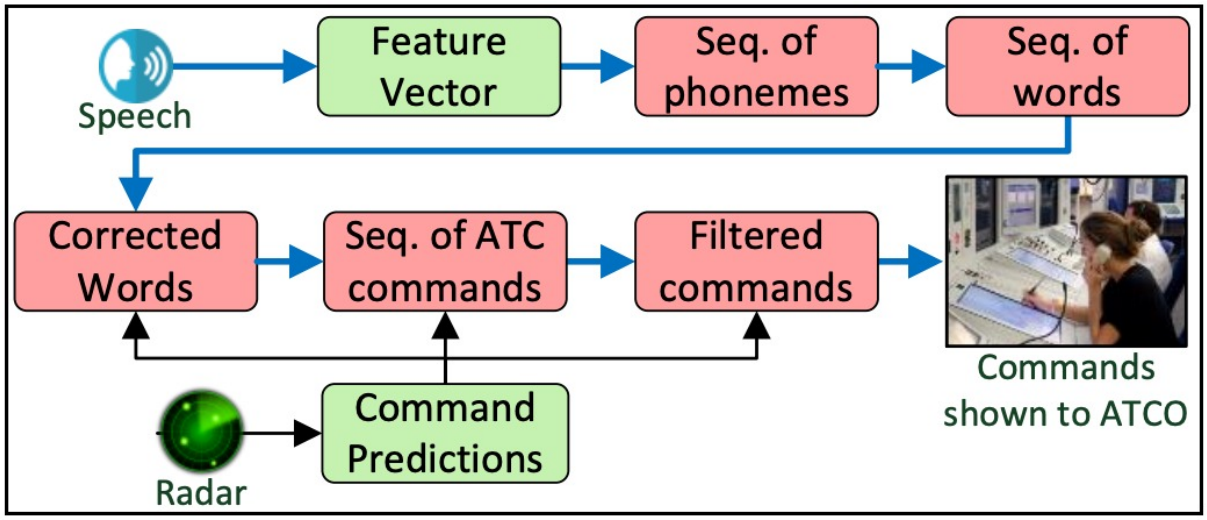

Figura 3.1: Transformação de entrada em saída para controlador (Fonte: [40]).

Zuluaga et al. [41] em seu artigo apresentam um *benchmark* de diversos modelos de ASR, baseados em redes neurais, aplicados para o domínio de ATC. Para tal fim, os autores utilizam de diversas bases de dados (Tabela 3.1) do domínio de tráfego aéreo, bases de domínio público como ATCOSIM e bases privadas como LDC ATCC, e de domínio comum, e utilizam combinações das bases de dados, conforme Tabela 3.2, com diversos modelos para descobrir os modelos mais performáticos no campo do tráfego aéreo. Os autores então diversas redes neurais com as bases de dados Librispeech e Commonvoice de inglês fora do domínio da aplicação, e então realizam operações de aprendizado por transferência para modelos derivados de seu modelo original. Com a modelagem os autores puderam obter uma WER de 5,0% para a base de dados ATCOSIM, mesma utilizada neste projeto, com o dataset de treino Tr1 + Tr2 que possui 176,4 horas de treino dentro do domínio de controle de tráfego aéreo. O trabalho destes autores serviu como uma base para as métricas de desempenho do trabalho corrente.

Srinivasamurthy et al. [43] utilizam aprendizado semi-supervisionado para um sistema ASR. Os autores utilizam 150 horas de dados fora do domínio do ATC com 5 horas de

Base de dados	Horas	Sotaque	Referência
MALORCA	13	Alemão, Tcheco	[42], [43]
LDC ATCC	72.5	Inglês Americano	[44]
HIWIRE	28.3	Francês, Grego, Italiano e Espanhol	[45]
ATCOSIM	10.6	Alemão, Alemão Suíço e Francês	[1]
UWB ATCC	20.6	Tcheco	[46]
AIRBUS	45	Francês	[47]
Librispeech	960	Inglês Diverso	[48]
Commonvoice	500	Inglês	[49]

Tabela 3.1: Gravações por língua nativa.

Nome	Horas	Bases
Train1	38.7	Atcosim + Malorca + UWB ATCC
Train2	137.7	Airbus + ATCC USA + Hiwire
Tr1 + Tr2	176.4	Train1 + Train2
OOD	1500	Librispeech + Commonvoice

Tabela 3.2: Datasets de Treino.

dados do domínio do ATC transcritos. Com este método os autores foram capazes de obter a WER de 9,4% usando datasets obtidos de Vienna ATC.

Pamplona et al. [50] utiliza redes neurais para prever o atraso aeronaves em seus voos. O atraso de aeronaves pode custar em custos tanto para a companhia aérea quanto para os passageitos. Nesse cenários os autores conseguiram predizer com acima de 90% de certeza os voos entre os aeroportos de Congonhas em São paulo e Santos Dumont no Rio de Janeiro.

O trabalho de Cruciol et al. [51] busca utilizar funções de recompensa para investigar o impacto dessas funções no processo do gerenciamento de tráfego aéreo, tanto em cenários de simulações quanto em cenários reais. Os autores propões duas funções de recompensa que podem ser auxiliares aos controoladores de tráfego em questões de segurança e justiça no tráfego aéreo.

Monteiro et al. [52] desenvolveram um framework de detecção e resolução de conflitos de aeronaves em 4 dimensões utilizando as bases de dados NoSQL Cassandra e MongoDB. O objetivo do trabalho de Monteiro et al. era a construção de uma solução que incorporasse a predição de trajetórias de aeronaves, poda de árvores de decisão, e bases de dados construídas especialmente para grandes quantidades de dados. Utilizando dados de 58% do território brasileiro, os autores obtiveram performance ótimas para a quantidade de dados utilizada. O trabalho realizado pelos autores apresenta ser robusto, e de rápida execução para que possa ser utilizado para dar assistência a controladores de voo.

Badrinath et al. [53] investigam o uso do reconhecimento de fala para o domínio de ATC. Seguindo uma abordagem de ASR ponta-a-ponta, e utilizando como base o modelo de rede neural *Deep Speech*, os autores tinham como objetivo, não só a transcrição do texto, como também a extração de características da comunicação, como os identificadores das aeronaves. Os autores utilizaram a base de dados AIRBUS-ATC e foram capazes de obter 17% de WER.

Capítulo 4

Métodos

Neste capítulo, abordaremos a definição do problema em conjunto com alguns acidentes marcantes e relevantes da aviação, também abordaremos o conjunto de dados utilizado, pré-processamento realizado, as métricas para avaliação do modelo de reconhecimento de fala utilizado, a proposta de solução e por fim os métodos experimentais para o desenvolvimento do trabalho.

4.1 Definição do Problema

A comunicação do piloto com o controle de tráfego aéreo, é um dos principais pilares da aviação moderna, é o permite milhares de aeronaves utilizando as vias aéreas, e levantando voo e pousando em aeroportos. É por meio da comunicação dos pilotos com os controladores de voo que são definidas as pistas de pouso e decolagem, o controle de altitude de aeronaves para o pouso e informação de emergências.

Na Figura 4.1 temos um exemplo do fluxo de aeronaves na costa leste dos EUA, um cenário quase caótico com tantas aeronaves em vôo ao mesmo tempo. É preciso, além de muita padronização em cima das manobras de voo, muito esforço dos controladores de tráfego para gerenciar individualmente essa quantidade de aeronaves, por isso a comunicação entre o controlador de voo e cada aeronave é indispensável para se evitar cenários catastróficos.

Apesar de ser um aspecto tão importante, nem sempre a comunicação é feita de modo que as duas pontas se entendam completamente. Abaixo citamos situações em que a comunicação falha foi um dos pontos críticos na aviação.

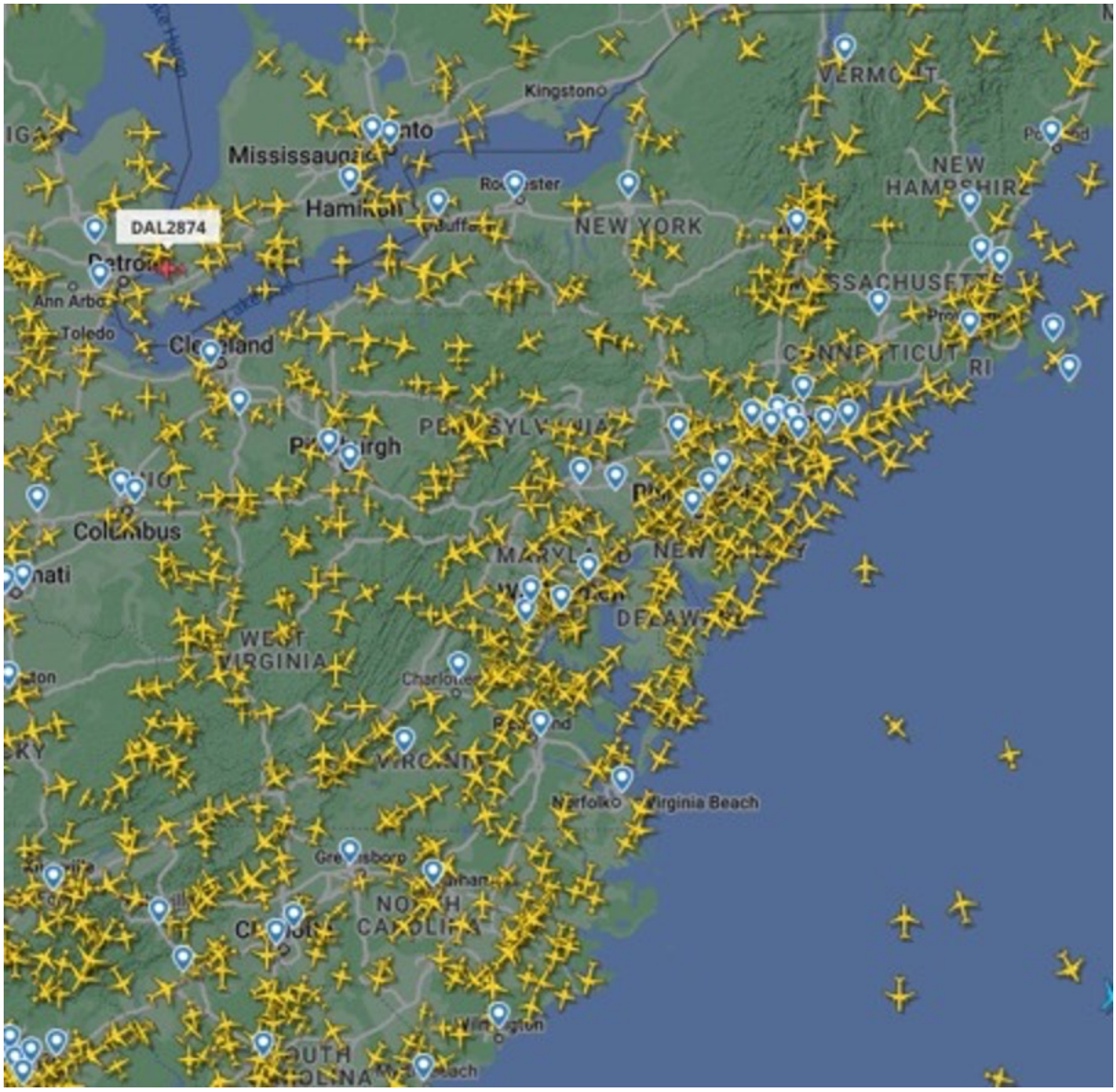

Figura 4.1: Quantidade de Aeronaves na costa leste dos EUA.

4.1.1 Março, 1977 - Desastre Aéreo de Tenerife

No dia 27 de março de 1977, um trágico acidente de aviação aconteceu no aeroporto de Tenerife [54], nas Ilhas Canárias, após um grupo de terroristas explodir uma bomba no saguão do aeroporto de Las Palmas. O aeroporto de Las Palmas foi fechado para pousos e decolagens, desviando todos os voos para Tenerife, um aeroporto menor e com menor capacidade de receber tantos aviões.

O voo da PanAm, com 396 pessoas a bordo, estava em uma rota de Los Angeles para Las Palmas e o voo da KLM, com 249 pessoas, já havia sido desviado para Tenerife. O aeroporto estava superlotado e, por isso, os aviões maiores estavam aguardando na pista

12, enquanto outras aeronaves menores estavam estacionadas na pista de taxeamento.

Enquanto aguardava a reabertura do aeroporto, a KLM solicitou um reabastecimento de 50 toneladas de combustível, o que foi considerado incomum para um voo de curta duração. Quando o aeroporto de Las Palmas reabriu, a PanAm solicitou autorização para decolar, mas a aeronave da KLM estava estacionada na frente, impedindo a decolagem da PanAm. Além disso, a tripulação da KLM havia permitido que os passageiros desembarcassem, o que atrasou ainda mais o taxiamento da aeronave da PanAm.

Assim que o aeroporto reabriu, a torre de controle autorizou a KLM a seguir para a pista de decolagem e, ao chegar no final, realizar uma volta de 180° e retornar para a decolagem. Enquanto a KLM seguia as instruções passadas pela torre, uma densa neblina cobriu a pista, reduzindo a visibilidade para 300 metros, menos da metade do requerido legalmente para a decolagem. Quando o KLM estava chegando ao final da pista, a torre autorizou a PanAm a iniciar o taxeamento, instruindo o piloto a ir até a terceira saída, que era a pista de taxeamento, para liberar a pista de decolagem para a KLM. A torre de controle pediu para que a PanAm seguisse para a terceira saída, que correspondia a uma curva de 150° para a esquerda, para liberar a pista para a KLM. Contudo, a tripulação da PanAm ficou confusa com a indicação, pois havia uma quarta saída que seria mais adequada para o tamanho do Boeing 747.

Infelizmente, enquanto a PanAm seguia para a terceira saída indicada, o capitão da KLM, Van Zanten, começou a avançar as manetes para decolar. O copiloto, Klaas Meurs, questionou a decisão do capitão, mas o avião já estava a 600 metros da outra aeronave e colidiu com o voo da PanAm, matando todos a bordo do Jumbo KLM e deixando apenas 61 pessoas vivas que estavam no Boeing 747. O acidente foi causado por vários fatores, incluindo falta de espaço no aeroporto, condições climáticas adversas e erros de comunicação.

Após o acidente de Tenerife, em 1977, houve uma série de mudanças significativas na comunicação na aviação. Algumas dessas mudanças incluem:

- **Melhorias na formação dos controladores de tráfego aéreo:** A formação dos controladores de tráfego aéreo foi aprimorada para incluir mais treinamento em comunicações claras e precisas, para garantir que não haja confusão ou mal-entendidos entre a torre de controle e as aeronaves.
- **Padronização das comunicações:** O uso de frases padronizadas foi amplamente implementado, incluindo as respostas aos comandos dos controladores de tráfego aéreo, para garantir que as mensagens sejam claras e compreendidas de forma consistente.

- **Introdução da tecnologia de comunicação a bordo:** A introdução de tecnologia de comunicação a bordo, como transponders e sistemas de comunicação de dados, permitiu uma melhor comunicação entre as aeronaves e a torre de controle, tornando a transmissão de informações mais precisa e segura.
- **Melhoria da infraestrutura de comunicação:** A infraestrutura de comunicação nas torres de controle e nas aeronaves também foi significativamente melhorada, para garantir uma comunicação clara e precisa em todas as condições meteorológicas e outras situações adversas.

Essas mudanças, entre outras, levaram a uma melhoria significativa na segurança da aviação, e ajudaram a prevenir acidentes futuros causados por problemas de comunicação.

4.1.2 Novembro, 1996 - Colisão Charkhi Dadri

O acidente aéreo de Charkhi Dadri [55] [56], também conhecido como desastre de Charkhi Dadri, ocorreu no dia 12 de novembro de 1996 e envolveu o voo 763 da Saudi Arabian Airlines, com 312 tripulantes, e o voo 1907 da Kazakhstan Airlines, com 37 tripulantes. Infelizmente, todas as 349 pessoas a bordo perderam a vida.

A colisão foi resultado de uma série de fatores, incluindo falhas no controle de tráfego aéreo e na comunicação entre o piloto do voo 1907, sua equipe e o controlador de voo.

No momento da decolagem do voo 763, o controlador de tráfego aéreo estabeleceu uma altitude de 10 mil pés, enquanto o voo 1907 voava a uma altitude de 23 mil pés. Devido à reserva do espaço aéreo de Nova Delhi para uso exclusivo da Força Aérea Indiana, as duas aeronaves acabaram voando na mesma via aérea, em direções opostas, o que representou o primeiro erro crítico.

A falta de comunicação eficiente e a falta de sincronização dos instrumentos de navegação dos dois aviões foram outros fatores que contribuíram para a tragédia. O conhecimento limitado de inglês dos pilotos do voo 1907 e a necessidade de tradução dos comandos da torre de controle para eles também representaram desafios.

A instrução para estabilizar a altitude de 14 mil pés até nova ordem foi dada ao voo 763 a apenas 1 minuto do impacto. Enquanto isso, o voo 1907 informou estar a 15 mil pés, quando na verdade estava a 16,3 mil pés e descia gradualmente, chegando a 14,1 mil pés em poucos segundos. A torre de controle avisou o voo 1907 sobre a presença do voo 763 a sua frente a poucos segundos do impacto, mas já era tarde demais.

A descida contínua do voo 1907 levou a uma colisão iminente quando os dois aviões estavam a 14,1 e 14 mil pés, respectivamente. A tragédia poderia ter sido evitada se, pelo menos, um dos aviões tivesse um sistema de evitação de colisões aéreas (TCAS). O TCAS (sistema de alerta de tráfego aéreo) é um sistema de alerta de colisão para aeronaves

comerciais que utiliza *transponders* para detectar outras aeronaves e alertar os pilotos sobre possíveis colisões. Nenhum dos dois aviões possuía este sistema.

Após o acidente de Charkhi Dadri, investigações foram realizadas para determinar as causas que levaram a esta tragédia. Com base nas conclusões, foi identificado como fator principal a decida não autorizada do voo 1907. Como resultado da investigação, algumas medidas de segurança foram recomendadas, incluindo:

- separação do tráfego aéreo descendente e ascendente através de corredores exclusivos;
- instalação de um sistema de radar secundário nos aeroportos da Índia, para incluir dados de altitude das aeronaves;
- instalação obrigatória de sistema TCAS II nos aviões comerciais para evitar colisões com outras aeronaves;
- exigência de melhor proficiência em inglês por parte de todos os operadores e do controle de tráfego aéreo.

O acidente de Charkhi Dadri é conhecido como a pior colisão aérea da história e as lições aprendidas têm sido incorporadas para aprimorar a segurança em voos. Embora a ampla instalação do sistema TCAS II tenha significativamente reduzido a chance de um evento semelhante, outros acidentes aéreos subsequentes mostraram que ainda existem riscos envolvidos mesmo com a presença desse sistema durante voos controlados.

4.2 Conjunto de Dados

Para o trabalho, a base de dados ATCOSIM [1] foi utilizada, tanto para o ajuste fino quanto para a validação dos resultados. O objetivo desta base é proporcionar acesso público a gravações de operações de controle de tráfego aéreo, tornando-se versátil para vários usos. A base de dados contém mais de 10 horas de comunicação simulada entre piloto e controlador de vôo, distribuídas em mais de 10.000 trechos de áudio. A Tabela 4.1 contém mais informações sobre a base de dados.

A base foi criada em cenários simulados no EUROCONTROL Experimental Centre (EEC), com controladores profissionais ativamente empregados no setor. Além disso, a base foi construída com a participação de cidadãos da Alemanha e Suíça, oferecendo uma variedade de sotaques e línguas maternais, como pode ser visto na Tabela 4.3. Essa tabela demonstra o valor da base de dados, já que contém uma ampla gama de pessoas falantes com sotaques variados.

Total de horas	51.4h
Total de horas de fala	10.4h
Controladores de vôo	10
total de gravações	10078
Total de palavras	10883

Tabela 4.1: Gravações por língua nativa (Fonte: [1]).

Apesar da qualidade da base, esta ainda possui erros e falhas, tanto na gravação, quanto na transcrição, como dito pelo autor [1], como:

- barulho humano como de respirações, tosse e risos;
- articulações não verbais como *ah*, *hm* e *nah*;
- gravações vazias onde o controlador apertou o botão de gravação e nada foi dito;
- conversas paralelas que não são direcionadas nem ao piloto nem ao controlador;
- palavras sem sentido;
- palavras desconhecidas.

A base de dados também conta com uma lista de palavras e expressões fora o inglês, todas expressões de boas vindas ou de despedida. A lista de expressões se encontra na Tabela 4.2.

4.2.1 Pré-processamento

A base de dados conta com caracteres e tags especiais, como o uso do caracter til antecedente a caracteres ditos pelo controlador, e uso o de tags como [NONSENSE] para indicar que o transcritor da base não conseguiu entender o trecho da comunicação.

Partindo deste ponto, o pré-processamento realizado retirou caracteres especiais, retirou as tags especiais e também transformou números para sua forma extensa de acordo com a frasologia correta.

Pela Tabela 4.3 pode ser observado o resultado do pré-processamento de algumas transcrições de referência da base de dados. No caso da primeira transcrição, sm2_01_204, as tags “<FL>” e “</FL>” foram removidas. No segundo caso, sm2_02_058, o caracter “~” foi removido. Já para o terceito caso, onde a transcrição era vazia, foi removida do trabalho.

Expressão	Significado em inglês	Língua materna
hallo	hello	Alemão
auf wiederhoren	goodbye	Alemão
gruss gott	hello	Alemão
servus	hi	Alemão
guten morgen	good morning	Alemão
guten tag	hello	Alemão
tschuss	goodbye	Alemão
tschu	goodbye	Alemão
danke	hello	Alemão
bonjour	hello	Francês
au revoir	goodbye	Francês
adieu	goodbye	Francês
merci	hello	Francês
hoi	hello	Holandês
dag	goodbye	Holandês
buongiorno	hello	Italiano
arrivederci	goodbye	Italiano
hejda	goodbye	Suíço
adios	goodbye	Espanhol

Tabela 4.2: Lista de expressões estrangeiras (Fonte: [1]).

Língua nativa	País de origem	Gravações
Alemão	Alemanha	4985
Francês da Suíça	Suíça	1000
Alemão da Suíça	Suíça	4093

Tabela 4.3: Gravações por língua nativa.

Transcrição	Original	Após pré-processamento
sm2_01_204	tunair four eight zero two contact rhein one two seven decimal three seven <FL> </FL>	tunair four eight zero two contact rhein one two seven decimal three seven
sm2_02_058	˜k ˜l ˜m three four six report your requested flight level	k l m three four six report your requested flight level
sm2_01_215	[EMPTY]	

Tabela 4.4: Resultado de transcrições após pré-processamento.

4.3 Métricas

Segundo McCowan et al. [57], uma métrica para ASR ideal deve ser:

- Direta: mensura um modelo ASR independente da aplicação;
- Objetiva: a medida deve ser calculada de maneira automática;
- Interpretável: o valor absoluto da medida deve dar uma ideia geral sobre a performance do modelo;
- Modular: a avaliação do modelo deve permitir uma análise completa da aplicação.

Tendo em vista os pontos acima, os seguintes modelos foram escolhidos para realizarmos as métricas do modelo desenvolvido.

4.3.1 Acertos, Substituições, Deleções e Inserções

Estes valores, são referentes à comparação entre a *string* de referência da base de dados, e a *string* hipótese fornecida pelo sistema de ASR. Considere as *strings* abaixo:

- Referência: swissair nine three five two climb flight level three five zero set course to gotil
- Hipótese: swissair nine three five two climb fl three five zero that is the course to gotthiel

Ao alinharmos as duas strings, podemos perceber que houveram:

- 3 substituições: *level -> fl*, *set -> that*, *gotil -> gotthiel*;
- 1 deleção: *flight*;
- 2 inserções: *is*, *the*.

Com o objetivo de quantificar as diferenças entre strings de referência e hipótese, foram utilizadas as métricas definidas nas subseções seguintes.

4.3.2 WER

Word error rate (WER) é uma das métricas mais usadas em modelos ASR [58], seu cálculo consiste na medição da porcentagem de palavras incorretas, e sua fórmula está abaixo:

$$\text{WER} = \frac{\text{Substituições} + \text{Deleções} + \text{Inserções}}{\text{Substituições} + \text{Deleções} + \text{Acertos}}$$

4.3.3 MER

Match error rate (MER) é a probabilidade de uma determinada sequência ser incorreta, ou o desempenho absoluto do modelo, e sua fórmula:

$$\text{MER} = \frac{\text{Substituições} + \text{Deleções} + \text{Inserções}}{\text{Substituições} + \text{Deleções} + \text{Acertos} + \text{Inserções}}$$

4.3.4 WIL

Word information lost (WIL) esta métrica mensura a porcentagem de palavras que foram transcritas incorretamente em comparação com a referência, ela pode ser calculada:

$$\text{WIL} = 1 - \frac{\text{Acertos}}{\text{Total palavras referência}} + \frac{\text{Acertos}}{\text{Total palavras predição}}$$

4.3.5 WIP

Word information preserved (WIP) esta métrica é o complementar da anterior, WIL. A métrica avalia a porcentagem de palavras transcrita corretamente, e pode ser calculada da seguinte forma:

$$\text{WIP} = \frac{\text{Acertos}}{\text{Total palavras referência}} + \frac{\text{Acertos}}{\text{Total palavras predição}}$$

4.4 Proposta de solução

A proposta deste projeto é aplicar o modelo Whisper, pré-treinado em bases de dados em inglês, ao problema de controle de tráfego aéreo. Em seguida, faremos uma especialização com aprendizado por transferência com a base de dados ATCOSIM, e avaliaremos o desempenho do modelo por meio de métricas relevantes. Finalmente, compararemos os resultados obtidos no trabalho.

4.5 Experimentos

O modelo Whisper foi pré-treinado em bases de dados de diversos tamanhos diferentes, descritos na Tabela 4.5. Cada tamanho oferece uma vantagem em desempenho e acurácia, sendo o modelo large, o com maior desempenho, porém, o que necessita de maiores recursos computacionais, e o modelo tiny sendo o modelo com menor desempenho, porém, o que precisa de menor poder computacional.

Tamanho	Parâmetros	Modelo inglês	VRAM necessária	Velocidade relativa
tiny	39M	tiny.en	˜1 GB	˜32×
base	74M	base.en	˜1 GB	˜16×
small	244M	small.en	˜2 GB	˜6×
medium	769M	medium.en	˜5 GB	˜2×
large	1550M	N/A	˜10 GB	˜1×

Tabela 4.5: Modelos pré-treinados do Whisper com tamanhos diferentes.

Modelo	Parâmetros	WER	MER	WIL	WIP
small	244M	0.32	0.30	0.45	0.55
medium	769M	0.25	0.24	0.37	0.63
medium.en	769M	0.26	0.25	0.38	0.62
large	1550M	0.23	0.22	0.34	0.66

Tabela 4.6: Modelos pré-treinados por métricas.

O modelo Whisper também conta com tamanhos especialistas em inglês, e conforme os autores do modelo, esses obtiveram melhor performance em cenários específicos para a língua inglesa.

Para a realização do ajuste fino e avaliação do modelo Whisper, a base de dados ATCOSIM foi subdividida em 3 (vide Tabela 5.3) e suas partições são descritas abaixo:

- Train: subdivisão utilizada para o treinamento de ajuste fino do modelo;
- Evaluation: subdivisão utilizada para se avaliar performance do modelo durante treinamento
- Test: Subdivisão utilizada para o teste do modelo após ajuste fino.

Para a avaliação do modelo Whisper no domínio de tráfego aéreo, foi definida a seguinte estratégia:

1. Testes do modelo Whisper pré-treinado em toda a base de dados ATCOSIM, para se ter conhecimento da performance bruta do modelo no domínio;
2. Estudo do desempenho do modelo pré-treinado para o domínio;
3. Ajuste fino do modelo Whisper com subdivisão de treino da base de dados ATCOSIM;
4. Estudo do desempenho do modelo após ajuste fino com subdivisão de teste da base de dados ATCOSIM;

Subdivisão	Porcentagem base inicial	Estimativa de horas
Train	60%	6h
Evaluation	20%	2h
Test	20%	2h

Tabela 4.7: Subdivisão base de dados.

5. Testes do modelo Whisper pré-treinado na subdivisão de treino da base de dados ATCOSIM;
6. Comparação dos resultados obtidos na subdivisão de treino da base de dados ATCOSIM.

Para o desenvolvimento do trabalho, os modelos small, medium, medium.en e large foram escolhidos para se avaliar a performance bruta do modelo, e os modelos small e medium, para serem feitos os ajustes finos. O modelo large não foi utilizado para o ajuste fino devido à falta de poder computacional necessária para tal.

Capítulo 5

Resultados

Neste capítulo apresentaremos os resultados obtidos do modelo Whisper em seu estado pré-treinado, e com o mesmo modelo após o ajuste fino, e então iremos comparar as métricas, apresentadas na seção 4.3.

5.1 Whisper Base Pré-Treinada

Como visto anteriormente, o modelo Whisper já possui diversos modelos pré-treinados. Nesta seção apresentaremos o resultado de alguns modelos pré-treinados, sem especialização de base, para a base inteira do ATCOSIM. Esta seção possui o propósito de entender o comportamento padrão do Whisper, onde se encontram seus erros, e compará-los com o modelo após a especialização de base.

Na Tabela 5.1 podemos observar a quantidade de acertos, substituições, deleções e inserções dos modelos, com um aumento do acerto do modelo small até o large. Podemos observar também que o modelo medium multilingual obteve maior quantidade de acertos se comparado ao modelo medium especializado na língua inglesa, apesar do artigo referência do Whisper [29] obter resultados melhores para a língua inglesa nos modelos específicos da língua inglesa. Isso pode se dar ao fato da base de dados utilizada no projeto se tratar de falantes não nativos da língua inglesa, com sotaques específicos de sua região, mas se deve principalmente pelo uso de expressões estrangeiras conforme explicitado na Tabela 4.2. Apesar deste fator, a diferença entre os modelos medium e medium.en não foi tão significativa como pode ser visto na Tabela 5.2, cujas métricas obtiveram resultados aproximados.

Com os resultados obtidos das métricas, podemos definir que o melhor modelo, como esperado, foi o modelo large, porém por uma margem abaixo do esperado, visto que este teve um treinamento muito mais elevado se comparado ao modelo medium.

Modelo	Acertos	Substituições	Deleções	Inserções
small	78846	19895	9171	5466
medium	84779	16430	6703	3684
medium.en	83786	17159	6967	3894
large	86540	14816	6556	3346

Tabela 5.1: Estatísticas dos modelos pré-treinados avaliados para toda base de dados ATCOSIM.

Modelo	WER	MER	WIL	WIP
small	0.32	0.30	0.45	0.55
medium	0.25	0.24	0.37	0.63
medium.en	0.26	0.25	0.38	0.62
large	0.23	0.22	0.34	0.66

Tabela 5.2: Métricas dos modelos pré-treinados avaliados para toda base de dados ATCOSIM.

Na Figura 5.1 é possível observar comparativamente a desempenho dos modelos pré-treinados, com pequenas melhoras partindo do modelo small até o modelo large. Na Figura 5.2 é possível conferir de forma visual o resultado das métricas calculadas.

A fim de se aprofundar nos possíveis erros do modelo ASR, gráficos da distribuição dos erros também são mostrados. Na Figura 5.3 podemos ver a distribuição do modelo small, na Figura 5.4 a distribuição do modelo medium, na Figura 5.5 a distribuição do modelo medium.en e por fim, na Figura 5.6 a distribuição do modelo large. Os gráficos ajudam a evidenciar a melhor desempenho do modelo large, se comparado aos outros modelos adjacentes.

Podemos então perceber que em todos os modelos, a maior porcentagem de erro se encontra nas substituições, o que pode ser um forte indício de que o modelo se beneficiaria do processo de ajuste fino.

5.2 Whisper Ajuste Fino

Para a correta realização do ajuste fino, é necessário identificar o limite de treinamento. Isso se deve afim de evitar o uso de um modelo viciado nos dados, pois o objetivo é obter um modelo que se adeque à diversas situações. Para isto, foram analisadas as curvas de aprendizado dos modelos small e medium. Com este objetivo, os modelos foram treinados pela técnica do *early stop* [15].

O modelo small obteve a curvas de aprendizado Figura 5.7 para o treinamento e a curva Figura 5.8 para a validação. Para este caso, o modelo começa a se establizar a

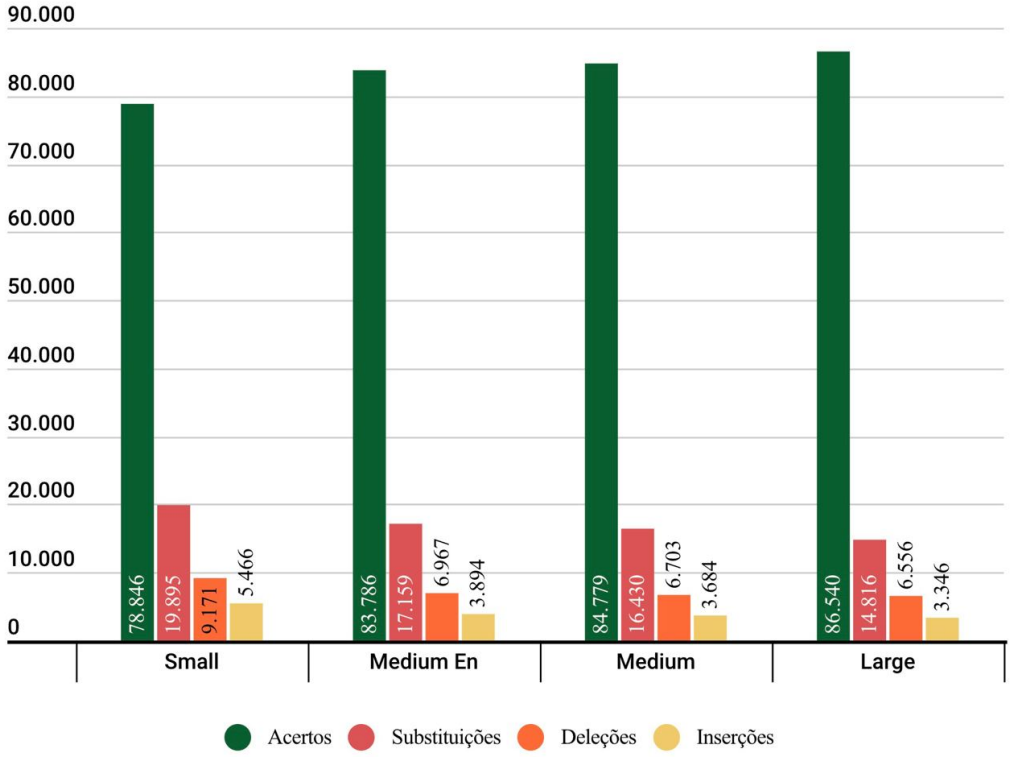

Figura 5.1: Gráficos comparativos das estatísticas dos modelos pré-treinados avaliados para toda a base de dados ATCOSIM.

Subdivisão	Porcentagem base inicial	Estimativa de horas
Train	60%	6h
Evaluation	20%	2h
Test	20%	2h

Tabela 5.3: Subdivisão base de dados.

partir do step 42, ou seja, o modelo começa a aprender comporamentos específicos da base de dados, portanto, o modelo small foi treinado até o step 42.

O modelo medium obteve a curvas de aprendizado Figura 5.9 para o treinamento e a curva Figura 5.10 para a validação. Para o modelo medium, esse já inicia a establização a partir do *step* 22. Portanto, o modelo medium foi treinado até o *step* 22.

Nesta seção analisaremos o comportamento do modelo com execução do ajuste fino nas bases de dados small e medium, denominadas de small-atcosim e medium-atcosim. Analisaremos se houveram melhoras com relação ao patamar inicial, e também iremos analisar as curvas de aprendizado.

Na Tabela 5.4 são apresentados os dados de acertos, substituições deleções e inserções dos modelos small, small-atcosim, medium e medium-atcosim.

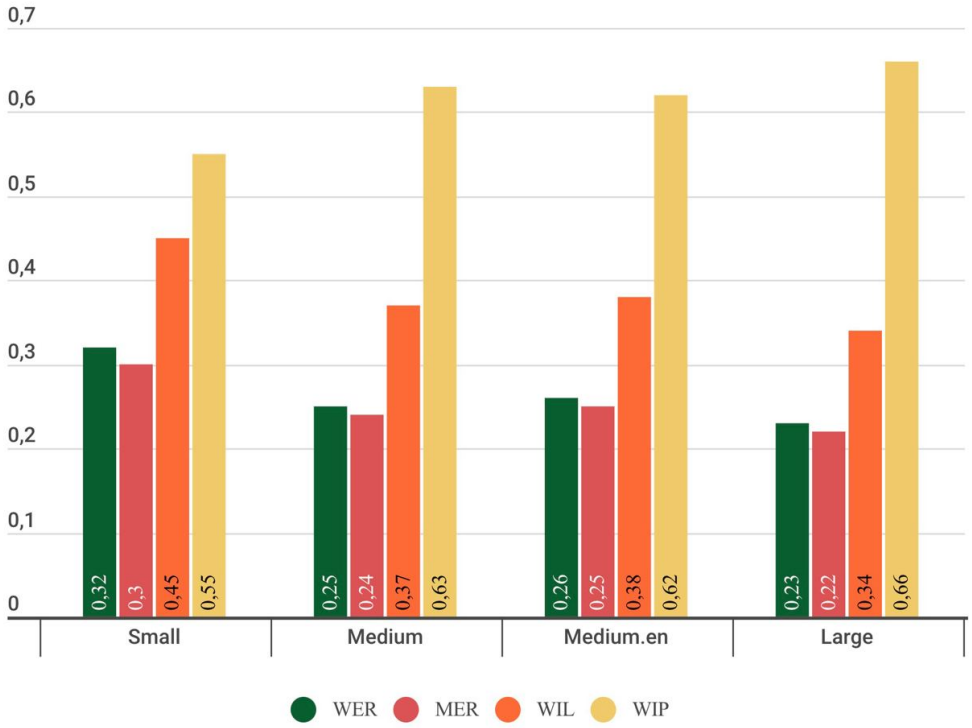

Figura 5.2: Gráficos comparativos das métricas dos modelos pré-treinados avaliados para toda a base de dados ATCOSIM.

Modelo	Acertos	Substituições	Deleções	Inserções
small	15630	4108	1760	1145
small-atcosim	20553	762	183	1900
medium	16823	3372	1303	784
medium-atcosim	19924	1237	337	699

Tabela 5.4: Modelos por estatísticas.

Observando os dados das Tabela 5.5 e dos gráfico Figura 5.11 e Figura 5.12 podemos concluir que a desempenho dos modelos small-atcosim e medium-atcosim sofreu significativa melhora com o ajuste fino em comparação com seu modelo pré-treinado, visto que todas as métricas sugeriram melhores resultados.

5.3 Análise dos Resultados

Entre os modelos pré-treinados do Whisper, aquele que obteve melhor desempenho foi o large, com uma WER de 0,23 como mostrado na Tabela 5.2. Com estes dados e as outras métricas, é possível perceber que o Whisper não possui um desempenho razoável para o

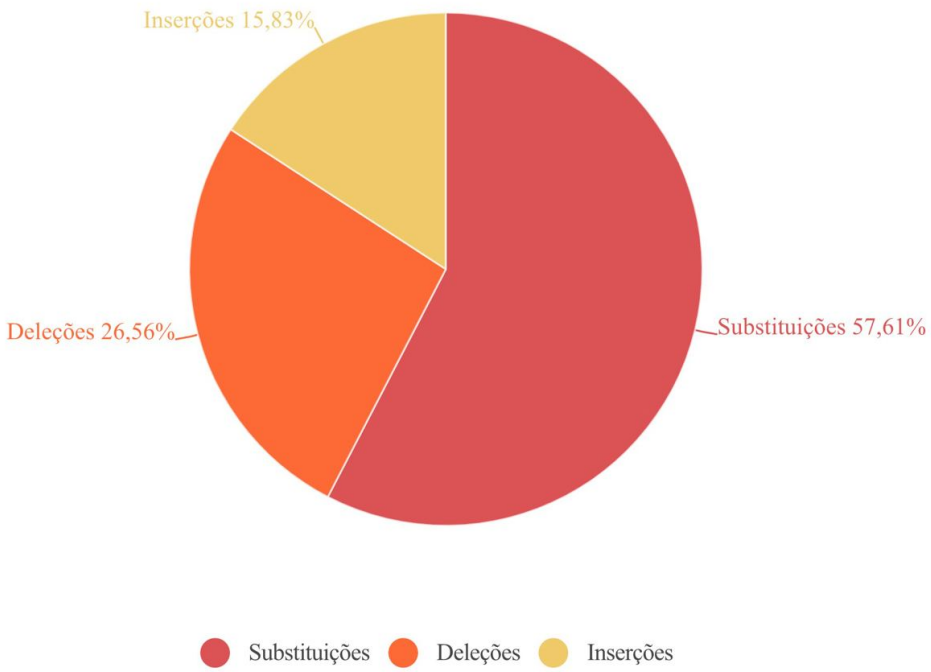

Figura 5.3: Gráficos comparativos dos erros obtidos pelo modelo small pré-treinado e avaliado para toda a base de dados ATCOSIM.

Modelo	WER	MER	WIL	WIP
small	0,33	0,31	0,46	0,54
small-atcosim	0,13	0,12	0,15	0,85
medium	0,25	0,24	0,37	0,63
medium-atcosim	0,10	0,10	0,15	0,85

Tabela 5.5: Modelos por métricas.

cenário de controle de tráfego aéreo de forma pré-treinada, apesar de possuir uma WER de 4,2% para cenários fora do domínio de ATC [29].

Observando a Figura 5.11, podemos observar que o modelo small-atcosim obteve ainda mais acertos se comparado com o modelo medium-atcosim, porém o segundo errou muito menos.

Em comparação com a quantidade de substiuições dos modelos pré-treinados para os modelos após ajuste fino, a porcentagem de substituições em comparação com a porcentagem total de erros diminuiu significativamente, partindo de 58.6% do total de erros para 26,8% nos modelos small.

Um outro ponto positivo do modelo, é sua característica multilingual. Radford et. al. [29] apresentam WER de 4,3% para o português, em comparação a 4,2% de WER para o

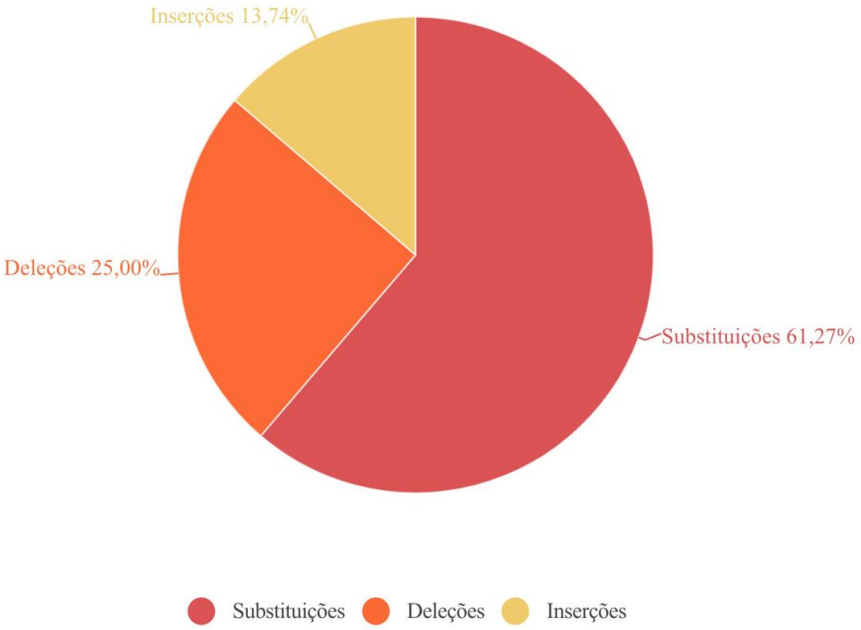

Figura 5.4: Gráficos comparativos dos erros obtidos pelo modelo medium pré-treinado e avaliado para toda a base de dados ATCOSIM.

inglês. Apesar de não ter sido tratado neste trabalho, o ajuste fino do modelo para bases de dados em português de ATC pode trazer melhorias do estado da arte nesse âmbito.

Apesar do baixo desempenho bruta do modelo para o cenário de testes, após o ajuste fino, podemos perceber que os testes obtiveram melhoria significativa de desempenho com poucas horas treino, para todas as métricas. Este comportamento demonstra a capacidade do modelo de aprender de forma rápida e em cenários de pouca quantidade de dados. Em termos de comparação, Zuluaga et. al [41] obtiveram em seu melhor modelo uma WER de 5%, porém com 176.4 horas de treino no domínio de ATC, enquanto o melhor modelo deste trabalho obteve 10% de WER com 6 horas de treino.

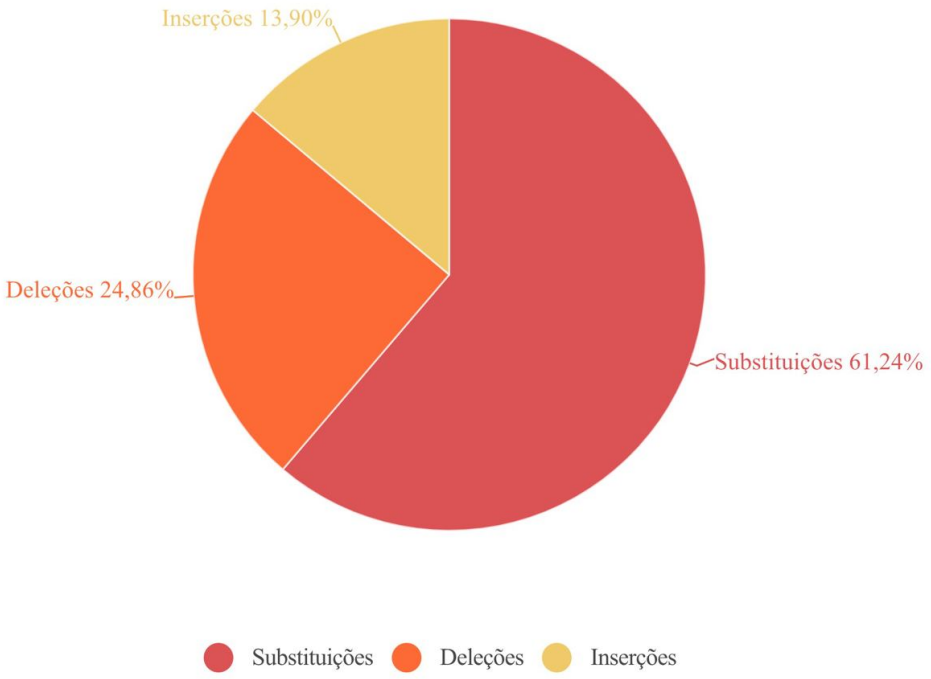

Figura 5.5: Gráficos comparativos dos erros obtidos pelo modelo medium.en pré-treinado e avaliado para toda a base de dados ATCOSIM.

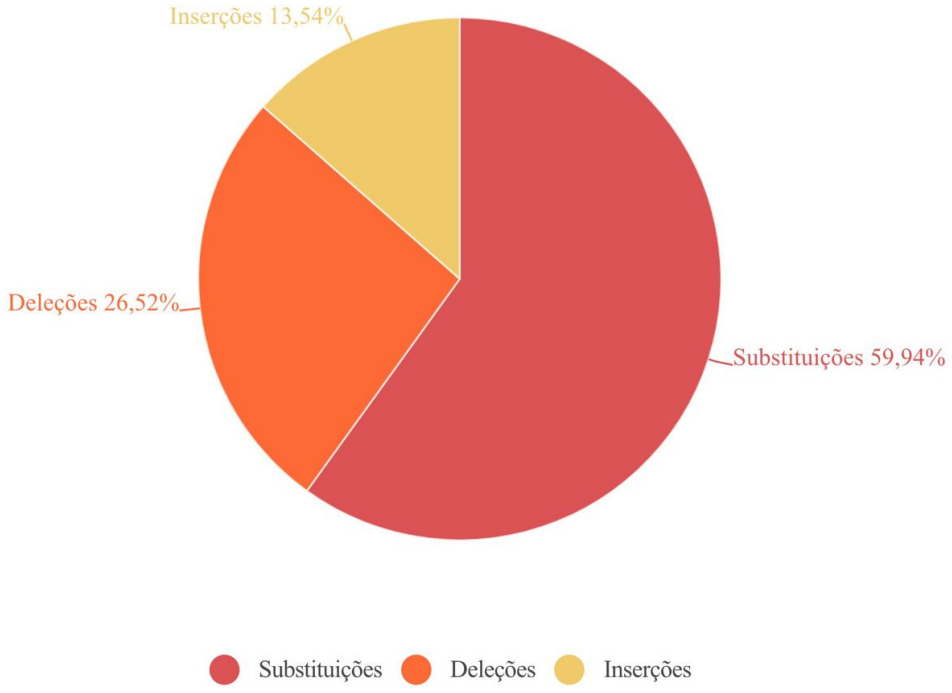

Figura 5.6: Gráficos comparativos dos erros obtidos pelo modelo large pré-treinado e avaliado para toda a base de dados ATCOSIM.

Figura 5.7: Curva de aprendizado: treinamento, modelo small.

Figura 5.8: Curva de aprendizado: avaliação, modelo small.

Figura 5.9: Curva de aprendizado: treinamento, modelo medium.

Figura 5.10: Curva de aprendizado: avaliação, modelo medium.

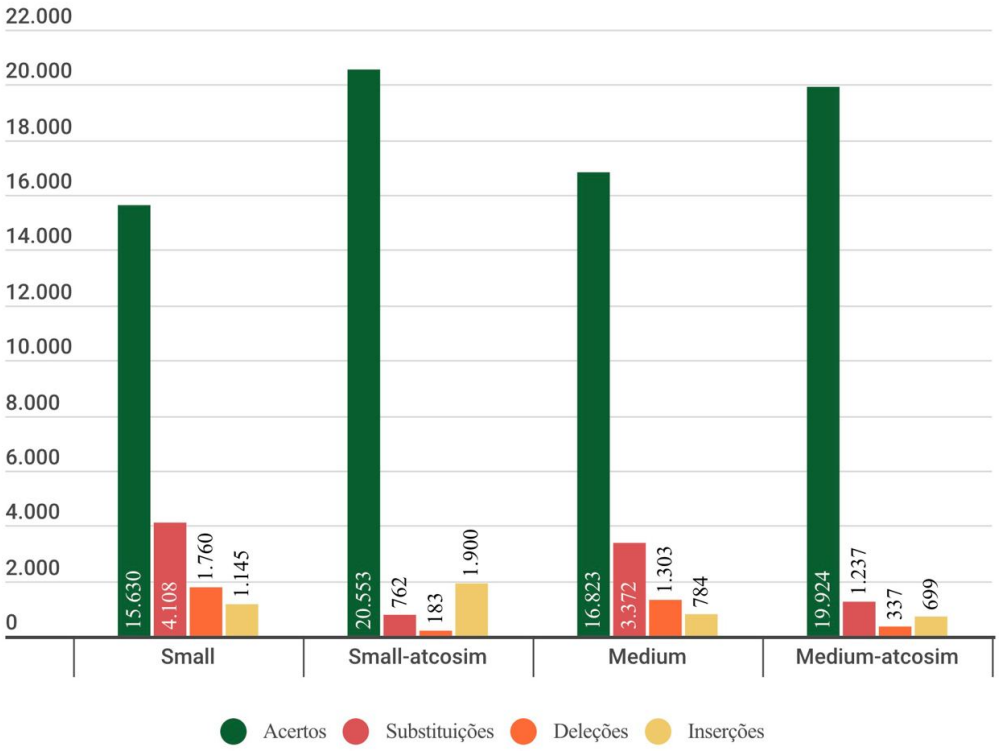

Figura 5.11: Gráficos comparativos das estatísticas dos modelos avaliados após ajuste fino.

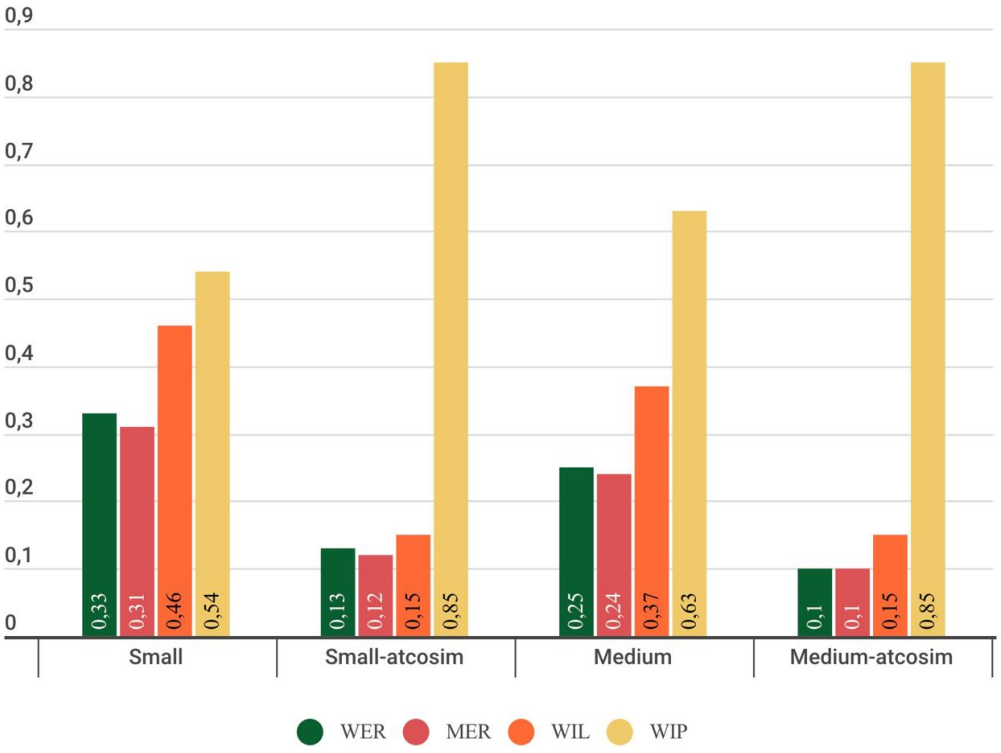

Figura 5.12: Gráficos comparativos das métricas dos modelos avaliados após ajuste fino.

Capítulo 6

Conclusão

Neste projeto, comparamos e avaliamos o desempenho do modelo Whisper no domínio de controle de tráfego aéreo. Analisamos a desempenho do modelo em diferentes estados pré-treinados e, em seguida, comparamos sua desempenho após o aprendizado por transferência por meio da base de dados ATCOSIM. Com base nas análises realizadas, concluímos sobre a efetividade do modelo Whisper no domínio de ATC e sua capacidade de se ajustar ao contexto do problema.

A seção 6.1 descreve as considerações finais do estudo, juntamente com a validação das hipóteses e dos objetivos previamente propostos. A seção 6.2 indica projeções futuras relacionadas ao desenvolvimento do trabalho.

6.1 Considerações Finais

Neste trabalho, analisamos a desempenho do modelo Whisper em diversos estados pré-treinados, e comparamos o seu desempenho com os mesmos modelos após a realização de especialização de base no domínio de ATC.

Com os resultados obtidos, podemos concluir que, apesar do modelo Whisper não possuir um ótimo desempenho para o domínio de ATC pela base de dados ATCOSIM em seu estado bruto, e apesar do modelo não possuir um processo de ajuste fino proposto pelos desenvolvedores, mas pela comunidade, o modelo apresentou melhorias significativas de taxas de erro após a especialização de base no domínio, reduzindo em até 25% a WER, apesar da baixa quantidade de horas de treino.

Este trabalho demonstra o potencial do modelo para cenários onde existem poucas bases de dados em um domínio específico, e especialmente mostra seu potencial no domínio de ATC.

6.2 Trabalhos Futuros

Durante a realização deste trabalho, foram observadas diversas possíveis melhorias e adições que podem ser feitas no futuro. A realização do ajuste fino do modelo large, a utilização de mais bases de dados de ATC para a especialização de base do modelo Whisper, e a especialização de base de ATC do modelo Whisper para a língua portuguesa. Estas melhorias podem trazer grande impacto na comunidade acadêmica e da aviação.

Referências

[1] Hofbauer, Konrad, Stefan Petrik e Horst Hering: *The ATCOSIM corpus of non-prompted clean air traffic control speech.* Em *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, maio 2008. European Language Resources Association (ELRA). `http://www.lrec-conf.org/proceedings/lrec2008/pdf/545_paper.pdf`. x, 6, 18, 24, 25, 26

[2] Juang, Biing Hwang e Lawrence R Rabiner: *Automatic speech recognition–a brief history of the technology development.* Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara, 1:67, 2005. 1

[3] Yu, Dong e Li Deng: *Automatic speech recognition*, volume 1. Springer, 2016. 1

[4] Wang, Dong, Xiaodong Wang e Shaohe Lv: *An overview of end-to-end automatic speech recognition.* Symmetry, 11(8):1018, 2019. 1

[5] Kamath, Uday, John Liu e James Whitaker: *Deep learning for NLP and speech recognition*, volume 84. Springer, 2019. 2

[6] Lin, Yi, Dongyue Guo, Jianwei Zhang, Zhengmao Chen e Bo Yang: *A unified framework for multilingual speech recognition in air traffic control systems.* IEEE Transactions on Neural Networks and Learning Systems, 32(8):3608–3620, 2020. 2

[7] Bolczak, Richard, John C Gonda III, William J Saumsiegle e Ronald A Tornese: *Controller-pilot data link communications (cpdlc) build 1 value-added services.* Em *The 23rd Digital Avionics Systems Conference (IEEE Cat. No. 04CH37576)*, volume 1, páginas 2–D. IEEE, 2004. 2

[8] McCarthy, John: *What is artificial intelligence?* 2004. 4

[9] Weigang, Li, Liriam Michi Enamoto, Denise Leyi Li e Geraldo Pereira Rocha Filho: *New directions for artificial intelligence: human, machine, biological, and quantum intelligence.* Frontiers of Information Technology & Electronic Engineering, 23(6):984–990, 2022. 4

[10] Reynolds, Douglas A *et al.*: *Gaussian mixture models.* Encyclopedia of biometrics, 741(659-663), 2009. 6

[11] *Gaussian mixture model.* `https://brilliant.org/wiki/gaussian-mixture-model`. Accessed: 2022-01-01. 7

[12] Eddy, Sean R: *Hidden markov models.* Current opinion in structural biology, 6(3):361–365, 1996. 7

[13] Rabiner, L. e B. Juang: *An introduction to hidden markov models.* IEEE ASSP Magazine, 3(1):4–16, 1986. 7

[14] Koehl, Patrice: *Protein structure classification.* Reviews in computational chemistry, 22:1–55, 2006. 7

[15] Goodfellow, Ian J., Yoshua Bengio e Aaron Courville: *Deep Learning.* MIT Press, Cambridge, MA, USA, 2016. `http://www.deeplearningbook.org`. 7, 8, 32

[16] Miao, Felix, Yang Sun, Junho Park, Daniel Willett e Puming Zhan: *Deep learning based mandarin accent identification for accent robust asr.* Em *INTERSPEECH*, páginas 510–514, 2019. 8

[17] Mohamed, Abdel rahman, George Dahl, Geoffrey Hinton *et al.*: *Deep belief networks for phone recognition.* Em *Nips workshop on deep learning for speech recognition and related applications*, volume 1, página 39, 2009. 8, 9

[18] Li, Jinyu *et al.*: *Recent advances in end-to-end automatic speech recognition.* APSIPA Transactions on Signal and Information Processing, 11(1), 2022. 9

[19] Zeineldeen, Mohammad, Aleksandr Glushko, Wilfried Michel, Albert Zeyer, Ralf Schlüter e Hermann Ney: *Investigating methods to improve language model integration for attention-based encoder-decoder asr models.* arXiv preprint arXiv:2104.05544, 2021. 9

[20] Makino, Takaki, Hank Liao, Yannis Assael, Brendan Shillingford, Basilio Garcia, Otavio Braga e Olivier Siohan: *Recurrent neural network transducer for audio-visual speech recognition.* Em *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*, páginas 905–912. IEEE, 2019. 9

[21] Li, Jason, Vitaly Lavrukhin, Boris Ginsburg, Ryan Leary, Oleksii Kuchaiev, Jonathan M. Cohen, Huyen Nguyen e Ravi Teja Gadde: *Jasper: An end-to-end convolutional neural acoustic model*, 2019. `https://arxiv.org/abs/1904.03288`. 9

[22] Graves, Alex, Abdel rahman Mohamed e Geoffrey Hinton: *Speech recognition with deep recurrent neural networks.* Em *2013 IEEE international conference on acoustics, speech and signal processing*, páginas 6645–6649. Ieee, 2013. 9

[23] Chorowski, Jan K, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho e Yoshua Bengio: *Attention-based models for speech recognition.* Advances in neural information processing systems, 28, 2015. 9

[24] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser e Illia Polosukhin: *Attention is all you need.* Advances in neural information processing systems, 30, 2017. 9, 10

[25] Nakatani, Tomohiro: *Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration*. Em *Proc. Interspeech*, volume 2019, 2019. 9

[26] Russell, Stuart e Peter Norvig: *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3ª edição, 2010. 9

[27] Devlin, Jacob, Ming Wei Chang, Kenton Lee e Kristina Toutanova: *Bert: Pre-training of deep bidirectional transformers for language understanding*. arXiv preprint arXiv:1810.04805, 2018. 9

[28] Floridi, Luciano e Massimo Chiriatti: *Gpt-3: Its nature, scope, limits, and consequences*. Minds and Machines, 30:681–694, 2020. 9

[29] Radford, Alec, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey e Ilya Sutskever: *Robust speech recognition via large-scale weak supervision*. arXiv preprint arXiv:2212.04356, 2022. 10, 31, 35

[30] *"introducing whisper*. `https://openai.com/blog/whisper/`. 12

[31] De Boer, Pieter Tjerk, Dirk P Kroese, Shie Mannor e Reuven Y Rubinstein: *A tutorial on the cross-entropy method*. Annals of operations research, 134:19–67, 2005. 11

[32] Weiss, Karl, Taghi M Khoshgoftaar e DingDing Wang: *A survey of transfer learning*. Journal of Big data, 3(1):1–40, 2016. 12

[33] Ghai, Wiqas e Navdeep Singh: *Literature review on automatic speech recognition*. International Journal of Computer Applications, 41(8), 2012. 15

[34] Li, Jinyu *et al.*: *Recent advances in end-to-end automatic speech recognition*. APSIPA Transactions on Signal and Information Processing, 11(1), 2022. 15

[35] Amodei, Dario, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen *et al.*: *Deep speech 2: End-to-end speech recognition in english and mandarin*. Em *International conference on machine learning*, páginas 173–182. PMLR, 2016. 16

[36] Collobert, Ronan, Awni Hannun e Gabriel Synnaeve: *Word-level speech recognition with a letter to word encoder*. Em *International Conference on Machine Learning*, páginas 2100–2110. PMLR, 2020. 16

[37] Benzeghiba, M., R. De Mori, O. Deroo, S. Dupont, T. Erbes, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris, R. Rose, V. Tyagi e C. Wellekens: *Automatic speech recognition and speech variability: A review*. Speech Communication, 49(10):763–786, 2007, ISSN 0167-6393. `https://www.sciencedirect.com/science/article/pii/S0167639307000404`, Intrinsic Speech Variations. 16

[38] Huang, Xuedong, James Baker e Raj Reddy: *A historical perspective of speech recognition*. Commun. ACM, 57(1):94–103, jan 2014, ISSN 0001-0782. `https://doi.org/10.1145/2500887`. 16

[39] Baevski, Alexei, Wei Ning Hsu, Alexis Conneau e Michael Auli: *Unsupervised speech recognition*. Advances in Neural Information Processing Systems, 34:27826–27839, 2021. 16

[40] Helmke, Hartmut, Matthias Kleinert, Jürgen Rataj, Petr Motlicek, Dietrich Klakow, Christian Kern e Petr Hlousek: *Cost reductions enabled by machine learning in atm how can automatic speech recognition enrich human operators performance?* 2019. 17

[41] Zuluaga-Gomez, Juan, Petr Motlicek, Qingran Zhan, Karel Vesely e Rudolf Braun: *Automatic speech recognition benchmark for air-traffic communications*. arXiv preprint arXiv:2006.10304, 2020. 17, 36

[42] Kleinert, Matthias, Hartmut Helmke, Gerald Siol, Heiko Ehr, Aneta Cerna, Christian Kern, Dietrich Klakow, Petr Motlicek, Youssef Oualil, Mittul Singh *et al.*: *Semi-supervised adaptation of assistant based speech recognition models for different approach areas*. Em *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, páginas 1–10. IEEE, 2018. 18

[43] Srinivasamurthy, Ajay, Petr Motlicek, Ivan Himawan, Gyorgy Szaszak, Youssef Oualil e Hartmut Helmke: *Semi-supervised learning with semantic knowledge extraction for improved speech recognition in air traffic control*. Relatório Técnico, 2017. 17, 18

[44] *"the air traffic control corpus (atc0) - ldc94s14a*. `https://catalog.ldc.upenn.edu/LDC94S14A`. 18

[45] J. Segura, T. Ehrette, A. Potamianos D. Fohr I. Illina P. Breton V. Clot R. Gemello M. Matassoni e P. Maragos: *"the hiwire database, a noisy and non-native english speech corpus for cockpit communication*. 18

[46] Šmídl, Luboš, Jan Švec, Daniel Tihelka, Jindřich Matoušek, Jan Romportl e Pavel Ircing: *Air traffic control communication (atcc) speech corpora and their use for asr and tts development*. Language Resources and Evaluation, 53:449–464, 2019. 18

[47] Delpech, Estelle, Marion Laignelet, Christophe Pimm, Céline Raynal, Michal Trzos, Alexandre Arnold e Dominique Pronto: *A real-life, french-accented corpus of air traffic control communications*. Em *Language Resources and Evaluation Conference (LREC)*, 2018. 18

[48] Panayotov, Vassil, Guoguo Chen, Daniel Povey e Sanjeev Khudanpur: *Librispeech: an asr corpus based on public domain audio books*. Em *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, páginas 5206–5210. IEEE, 2015. 18

[49] Ardila, Rosana, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers e Gregor Weber: *Common voice: A massively-multilingual speech corpus*. arXiv preprint arXiv:1912.06670, 2019. 18

[50] Pamplona, Daniel Alberto, Li Weigang, Alexandre Gomes de Barros, Elcio Hideiti Shiguemori e Claudio Jorge Pinto Alves: *Supervised neural network with multilevel input layers for predicting of air traffic delays*. Em *2018 International Joint Conference on Neural Networks (IJCNN)*, páginas 1–6. IEEE, 2018. 18

[51] Cruciol, Leonardo LBV, Antonio C de Arruda Jr, Li Weigang, Leihong Li e Antonio MF Crespo: *Reward functions for learning to control in air traffic flow management*. Transportation Research Part C: Emerging Technologies, 35:141–155, 2013. 18

[52] Monteiro, Lucas Borges, Vitor Filincowsky Ribeiro, Cristiano Perez Garcia, Geraldo Pereira Rocha Filho e Li Weigang: *4d trajectory conflict detection and resolution using decision tree pruning method*. IEEE Latin America Transactions, 21(2):277–287, 2023. 18

[53] Badrinath, Sandeep e Hamsa Balakrishnan: *Automatic speech recognition for air traffic control communications*. Transportation research record, 2676(1):798–810, 2022. 19

[54] *Tenerife report*. `https://www.project-tenerife.com/engels/rapporten.htm`. Accessed: 2022-01-01. 21

[55] *Charki-dadri report*. `https://www.dgca.gov.in/digigov-portal/Upload?flag=iframeAttachView&attachId=130614975&mainAccidentReports`. Accessed: 2022-01-01. 23

[56] *Charki-dadri case study*. `https://web.archive.org/web/20160223114256/http://www.cdmhipa.in/images/Case%20Study/casestudy.php`. Accessed: 2022-01-01. 23

[57] McCowan, Iain A, Darren Moore, John Dines, Daniel Gatica-Perez, Mike Flynn, Pierre Wellner e Hervé Bourlard: *On the use of information retrieval measures for speech recognition evaluation*. Relatório Técnico, IDIAP, 2004. 27

[58] Errattahi, Rahhal, Asmaa El Hannani e Hassan Ouahmane: *Automatic speech recognition errors detection and correction: A review*. Procedia Computer Science, 128:32–37, 2018. 27