AS-ASR

A Lightweight and Real-Time Aphasia-Specific Auto Speech Recognition System

Chen Bao

Delft University of Technology

TU Delft

AS-ASR

A Lightweight and Real-Time Aphasia-Specific Auto Speech Recognition System

by

Chen Bao

to obtain the degree of Master of Science at the Delft University of Technology,
to be defended publicly on Monday July 21, 2025 at 10:00 AM.

Student number: 5957370
Project duration: September 1, 2024 – July 20, 2025
Thesis committee:

Dr. Sijun Du	TU Delft, Chair
Dr. Chang Gao	TU Delft, Supervisor
Dr. Qinwen Fan	TU Delft, Core Member

This thesis is confidential and cannot be made public until July 21, 2027.

Cover:
Style: TU Delft Report Style, with modifications by Daan Zwaneveld

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Abstract

Aphasia is a language disorder caused by brain damage. It often behaves in the form of disfluent speech, fragmented grammar, and irregular word usage. These characteristics make it difficult for current automatic speech recognition (ASR) systems to produce accurate transcriptions. While models like Whisper perform well on fluent speech, they often struggle with aphasic speech, especially in low-resource clinical settings. This thesis presents AS-ASR, an aphasia-specific ASR system built on the lightweight Whisper-tiny model and optimized for real-time use on edge devices. To improve transcription accuracy, we created a mixed dataset of fluent and aphasic speech and used the GPT-4 large language model (LLM) to clean and refine transcripts from disfluent recordings. We tested different ratios of aphasic to typical data during training to find a balance that supports both accuracy and generalization. In addition, we applied mixed-precision quantization and a simple energy-based voice activity detection method to reduce model size and inference time. The fine-tuned model achieved up to 40% lower word error rates on aphasic speech and also kept accurate performance on fluent speech. Our findings suggest that targeted fine-tuning and lightweight deployment strategies can make ASR systems more accessible and effective for clinical and assistive use.

Acknowledgement

I would like to express our sincere appreciation to the AphasiaBank project, supported in part by the National Institutes of Health – National Institute on Deafness and Other Communication Disorders (NIH-NIDCD) under Grant R01-DC008524 (2022–2027), for granting me access to the aphasic speech dataset, which played a key role in supporting this research. In particular, I gratefully acknowledge the use of the following subcorpora: the ACWT Corpus, Adler Corpus, APROCSA Corpus, Boston University Corpus, and University of Kansas Corpus.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
ASR	Automatic Speech Recognition
CTC	Connectionist Temporal Classification
GELU	Gaussian Error Linear Unit
LTH	Lottery Ticket Hypothesis
ONNX	Open Neural Network Exchange
PCA	Principal Component Analysis
PEFT	Parameter-Efficient Fine-Tuning
PPA	Primary Progressive Aphasia
PPAOS	Primary Progressive Aphasia of Speech
ReLU	Rectified Linear Unit
SSD	Speech Sound Disorde
VTLP	Vocal Tract Length Perturbation
VAD	Voice Activity Detection
WER	Word Error Rate

1 Introduction

1.1. Motivation

Aphasia, often caused by stroke or brain injury, presents substantial challenges in spoken communication due to disfluent, fragmented, and atypical speech patterns. While mainstream automatic speech recognition (ASR) models such as Whisper have achieved impressive performance on fluent speech, they struggle significantly with pathological speech like aphasia. This mismatch limits the practical application of ASR technologies in clinical and assistive scenarios.

Moreover, the increasing demand for speech recognition in healthcare settings underscores the importance of lightweight, efficient, and privacy-preserving ASR systems that can operate offline. These constraints highlight the need for customized ASR frameworks that not only accommodate disordered speech but also support real-time deployment under resource-constrained conditions.

1.2. Problem Statement

Existing ASR models are predominantly trained on clean and fluent speech, resulting in poor generalization to aphasic speech. This leads to high word error rates, loss of speaker intent, and ultimately hinders their usability in clinical or assistive applications. Additionally, current ASR systems are often too resource-intensive to be deployed on embedded edge devices without significant compromise in speed, accuracy, or memory footprint.

Therefore, there exists a twofold challenge:

- Enhancing ASR model robustness to effectively recognize disordered speech, especially aphasic utterances.
- Optimizing the model architecture and inference pipeline for deployment on resource-constrained platforms, ensuring low latency and efficient computation.

1.3. Research objectives

This thesis aims to address the above challenges through the following objectives:

- **Develop** an aphasia-specific speech recognition framework based on Whisper-tiny, capable of handling disfluent and atypical speech patterns through targeted fine-tuning.
- **Investigate** the impact of hybrid data composition (aphasic and typical speech) on model generalization and performance under different mixing ratios.
- **Convert** the labels from AphasiaBank by using GPT-4-based transcript refinement with carefully designed prompts to provide better training supervision.
- **Deploy** the fine-tuned model in low-resource conditions, evaluating its performance in terms of inference speed, memory usage, and model size under real-world constraints.

1.4. Thesis Organization

This thesis is organized into seven chapters. Chapter 1 introduces the motivation, problem statement, research objectives, and an overview of the thesis structure. Chapter 2 provides background knowledge essential to the work, including the fundamentals of ASR, characteristics of aphasic speech, and model optimization techniques for resource-constrained deployment.

Chapter 3 reviews related work in the fields of disordered speech ASR, Whisper-based approaches for pathological speech, and efficient ASR systems. Chapter 4 explains the proposed methodology, covering the system overview, model selection, hybrid data fine-tuning strategy, and aphasia-centric dataset construction.

Chapter 5 presents the software implementation pipeline, including data preprocessing, model fine-tuning, evaluation setup, and error analysis. Chapter 6 focuses on model optimization and deployment strategies, such as mixed-precision quantization, handling of long audio using the max_new_tokens parameter, and VAD integration for real-time applications.

Finally, Chapter 7 summarizes the contributions of this work and outlines directions for future research toward more accessible and efficient ASR systems for individuals with speech disorders.

2

Background

2.1. Automatic Speech Recognition (ASR)

Automatic Speech Recognition (ASR) refers to the process of converting a spoken utterance into a sequence of words. It plays a vital role in various applications such as virtual assistants, real-time captioning, transcription services, and accessibility tools for individuals with speech and hearing impairments.

Historically, ASR systems followed a modular architecture comprising several independent components. These included a feature extractor (typically computing Mel-frequency cepstral coefficients), an acoustic model (often based on Hidden Markov Models, HMMs [1]), a pronunciation lexicon, a statistical language model (such as n-grams [2]), and a decoder that integrates all components to find the most probable word sequence. While effective, such systems required complex engineering, extensive domain knowledge, and handcrafted features, and they were often brittle in noisy or disfluent environments.

Recent advances in deep learning have led to a paradigm shift toward end-to-end ASR systems. These approaches integrate all components into a single trainable neural network, significantly simplifying the pipeline. End-to-end ASR models, such as those based on Connectionist Temporal Classification (CTC) [3], attention-based encoder-decoder structures, and Transformer architectures [4], have demonstrated superior performance and robustness, particularly in low-resource and multilingual settings.

Whisper, developed by OpenAI, is a state-of-the-art end-to-end ASR model based on the Transformer encoder-decoder architecture. It takes log-Mel spectrograms as input and directly generates text tokens as output, eliminating the need for separate acoustic or language models. Unlike conventional systems, Whisper is trained on a large-scale, multilingual, and multitask dataset, enabling it to generalize well across a variety of speech types, accents, and domains [5].

Whisper's encoder processes the input audio into high-level representations, while its decoder generates the transcribed text, optionally conditioned on task-specific tokens (e.g., language ID or transcription/translation mode). Its robustness in zero-shot and low-resource scenarios makes it particularly attractive for real-world applications involving disfluent or impaired speech.

In this thesis, Whisper serves as the foundation for investigating ASR performance on aphasic speech, especially in edge computing environments. Its compact variants (e.g., Whisper-tiny or Whisper-small) are particularly suited for deployment on resource-constrained platforms.

2.2. Aphasic Speech

Aphasia is a language disorder that affects a person's ability to speak, understand, read, and write, while leaving their intelligence intact [6]. As it is shown in Figure 2.1, it typically results from damage to key language centers in the left hemisphere of the brain, most notably Broca's area (associated with speech production) and Wernicke's area (associated with language comprehension) [7]. It often occurs

suddenly following events such as stroke, traumatic brain injury, brain tumors, or infections, causing the individual's everyday language ability to feel as if a "familiar language suddenly becomes foreign."

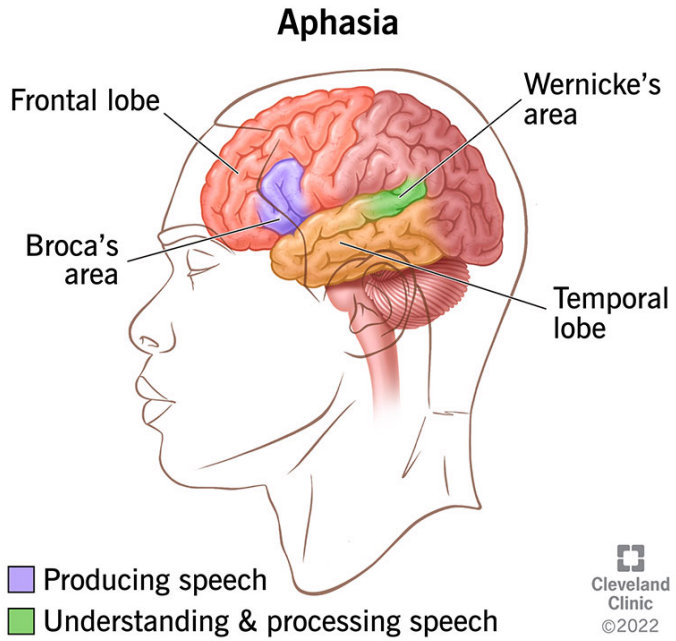

Figure 2.1: Key Brain Regions Involved in Aphasia: Broca's and Wernicke's Areas from Cleveland Clinic [7]

Table 2.1 summarizes typical aphasic speech examples across different types of aphasia, along with their original intended meanings based on the Cleveland Clinic and linguistic standards [7]. From this overview, it is evident that aphasia can manifest through various language impairments. First, speech production difficulties are common. This includes non-fluent aphasia (e.g., Broca's aphasia), where speech is short, effortful, and grammatically simplified, though comprehension remains relatively intact; and fluent aphasia (e.g., Wernicke's aphasia), where speech remains fluent but content is often incoherent, and comprehension is poor. Second, word-finding difficulties (anomic aphasia) are observed, in which individuals know what they want to say but cannot retrieve the precise word. Third, phonological or semantic errors may occur, such as phonemic paraphasia (for example, saying "tat" instead of "cat") or semantic paraphasia (for example, saying "banana" instead of "apple"). These may even include made-up words known as neologisms. Fourth, comprehension and response issues can arise, making it difficult for the patients to understand others or repeat sentences. Fifth, reading and writing impairments may also be present, including spelling errors, incomplete sentence structure, and difficulties in understanding or expressing numbers, time, and related concepts. These symptoms vary across different types of aphasia and often require professional diagnosis and therapeutic intervention.

Aphasic speech presents significant challenges to ASR systems due to its highly irregular and variable nature. Patients often produce fragmented, grammatically incomplete utterances, exhibit phonemic and semantic substitutions, or even create neologisms, all of which deviate from the standard speech patterns that ASR models are typically trained on. Additionally, individual differences in speech impairment, coupled with frequent articulation errors and limited publicly available aphasic speech datasets, hinder effective model training and generalization. These challenges necessitate specialized strategies such as fine-tuning on aphasic datasets, incorporating language model correction, speaker-adaptive techniques, data augmentation, and the use of more context-aware evaluation criteria.

Table 2.1: Typical Types of Aphasia and Their Speech Characteristics from Cleveland Clinic [7]

Type of Aphasia	Aphasic Speech Example	Original Meaning
Broca (Non-fluent)	"Walk dog... park..." (missing grammar)	"I'm going to walk the dog in the park."
Wernicke (Fluent but confused)	"The snoover grolks in the tree-top." (nonsense words)	"The bird sings in the treetop."
Anomic (Word-finding difficulty)	"Can you pass me the... the... uh... thing for writing?"	"Can you pass me the pen?"
Conduction (Repetition difficulty)	Clinician: "The window is open." Patient: "The... winder is open." (phonemic error)	Correct repetition: "The window is open."
Global (Severe impairment)	"Uhh... mmm..." (minimal output)	"I want some water."
Mixed Transcortical (Expression and comprehension impaired)	"No... I... book..." (fragmented, poor understanding)	"I don't have the book with me."

2.3. Model Optimization Techniques for Resource-Constrained Deployment

2.3.1. Motivation for Model Optimization

Deploying deep learning models in resource-constrained environments poses several challenges due to limited computational capacity, restricted memory, and strict power budgets. While high-performance ASR models such as Whisper achieve excellent accuracy, they typically rely on large parameter counts and demand substantial compute power during inference. This makes them impractical for direct use in low-resource scenarios. To address these limitations, a range of model optimization techniques has been developed to reduce model size, improve inference speed, and lower energy consumption, thereby enabling lightweight ASR deployment under constrained conditions.

2.3.2. Common Optimization Techniques

Several well-established techniques are widely used to optimize deep learning models for edge inference. These methods aim to balance the trade-offs between model performance, latency, and hardware efficiency.

Quantization

Quantization reduces the numerical precision of model weights and activations from 32-bit floating point (FP32) to lower-bit representations such as 8-bit integers (INT8) or 16-bit floats (FP16). This significantly reduces model size and accelerates computation, especially on hardware that supports low-precision arithmetic. Quantization can be applied statically (with calibration) or dynamically (without retraining). It is one of the most commonly used methods for edge deployment.

Pruning

Pruning involves removing redundant or less important weights, neurons, or channels from the model. Structured pruning techniques eliminate entire filters or layers, leading to a reduction in both memory usage and computational complexity. Although aggressive pruning may degrade accuracy, mild or gradual pruning can preserve performance while improving efficiency.

Knowledge Distillation

In this technique, a smaller "student" model is trained to mimic the behavior of a larger, pre-trained "teacher" model. The student model learns from the soft labels or intermediate representations of the teacher, enabling it to achieve comparable accuracy with fewer parameters. This method is particularly effective when designing custom lightweight models for edge inference.

Graph Optimization and Operator Fusion

Frameworks such as ONNX Runtime and TensorRT apply graph-level optimizations that fuse multiple operations, eliminate redundant computation, and re-order execution for better hardware compatibility. These improvements reduce inference time without modifying the model architecture.

2.3.3. Relevance to This Thesis

Given the limited memory and compute resources available in low-resource environments, model optimization is essential for deploying Whisper-based ASR systems efficiently. Among the available techniques, dynamic quantization offers a practical and low-overhead solution, particularly due to its compatibility with Transformer architectures and its ability to be applied without extensive retraining.

In this thesis, we adopt a mixed-precision dynamic quantization approach, where selected linear and attention layers of the Whisper model are quantized to INT8 while other components remain in higher precision formats (e.g., FP16 or FP32) to preserve model stability and recognition accuracy. This strategy balances performance and efficiency, significantly reducing memory footprint and inference latency while maintaining acceptable ASR accuracy.

By applying mixed-precision dynamic quantization, we enable real-time, low-power ASR execution under resource-constrained conditions. This deployment strategy is especially valuable for speech recognition systems tailored for individuals with aphasia, as it ensures both responsiveness and data privacy in local, offline environments.

3

Related Works

3.1. Disordered Speech ASR: Early Challenges and Progress

Early efforts in ASR for disordered speech focused primarily on demonstrating feasibility and addressing the fundamental mismatch between typical ASR systems and pathological speech characteristics. Fraser et al. [8] were among the pioneers in this field, employing ASR-derived lexical and acoustic features to classify subtypes of primary progressive aphasia (PPA) using the AphasiaBank dataset. Despite high WERs, their work underscored that even noisy transcriptions could yield clinically informative patterns, thus establishing a foundation for ASR-based diagnosis and screening.

Subsequent studies began to explore the technical limitations and opportunities for improving disordered speech recognition. Kim et al. [9] investigated the clinical validity of an ASR system trained on a small child-specific speech sound disorder (SSD) corpus. Their model which was based on wav2vec 2.0, showed strong correlation with clinician assessments but also revealed that phoneme-level mismatches remained a barrier for clinical reliability, especially in articulatorily ambiguous cases. In parallel, Tetzloff et al. [10] evaluated ASR transcription accuracy for primary progressive apraxia of speech (PPAOS), finding that although ASR output correlated with human-rated severity, it failed to distinguish between phonetic and prosodic subtypes, exposing the limitations of current systems in fine-grained clinical differentiation.

The scarcity of training data and the acoustic variability in pathological speech pose persistent challenges. Addressing this, Geng et al. [11] conducted a systematic comparison of data augmentation techniques, such as vocal tract length perturbation (VTLP), tempo adjustment, and speed modification. Their results demonstrated that speaker-adaptive training combined with speed perturbation achieved a significant reduction in WER of up to 2.92% compared to the baseline on the UASpeech dataset. Expanding upon this, Jin et al. [12] introduced a GAN-based adversarial augmentation method that transforms normal speech into disordered-like speech by learning spectrotemporal differences. This technique yielded up to 3.05% absolute WER improvement over traditional augmentation methods.

Beyond augmentation, domain adaptation and personalized modeling have also emerged as effective strategies. A series of studies by Tobin et al. [13] explored these dimensions from both modeling and data perspectives. In one study, they compared ASR performance on read versus conversational speech from individuals with disordered speech, revealing a substantial performance gap due to hesitations, informal grammar, and atypical prosody characteristic of conversational settings. This work emphasized the necessity of including conversational speech in training data to improve model robustness. In a complementary project, the same research group investigated the feasibility of training a single, speaker-independent ASR model by incorporating approximately 1,000 hours of disordered speech into the fine-tuning process [14]. Remarkably, this model achieved up to 33% improvement in recognition accuracy for disordered speech without compromising performance on typical speech, demonstrating the potential of inclusive training to reduce recognition disparities and improve accessibility.

Meanwhile, large-scale foundation model adaptation has gained traction. Sanguedolce et al. [15] fine-tuned the Whisper model on their SONIVA dataset containing speech from 600 stroke patients. Not only did their model outperform disorder-specific baselines, but it also generalized well to other corpora, including AphasiaBank and DementiaBank, demonstrating the feasibility of cross-pathology transfer and motivating future work in universal disordered speech recognition.

Finally, Qian et al. [16] provided a comprehensive survey of technologies in dysarthric speech recognition. Their review synthesized the evolution of acoustic and multimodal fusion approaches and highlighted that while audiovisual models improve performance, their practical deployment is hindered by the lack of large-scale, multimodal pathological speech datasets. The authors called for more open-source databases and better domain adaptation strategies to bridge the gap between ASR technology and clinical speech applications.

Together, these works trace a progression from early exploratory research and baseline adaptation to more sophisticated techniques such as adversarial augmentation, personalized training, and foundation model fine-tuning. Despite considerable progress, the field continues to face challenges related to data sparsity, generalizability, and evaluation metrics that are sensitive to the unique characteristics of disordered speech.

3.2. Whisper for Pathological and Aphasic Speech

Recent advances in foundation models such as OpenAI's Whisper have catalyzed research into their application for pathological speech recognition. A particularly influential line of work by Sanguedolce et al. has systematically explored Whisper's potential across a range of speech disorders through a series of studies.

Their foundational effort, Universal Speech Disorder Recognition, proposed the use of Whisper-medium as a base model fine-tuned on the newly constructed SONIVA corpus (a 15-hour dataset comprising over 600 stroke patients with varying aphasia subtypes). The fine-tuned model demonstrated strong cross-pathology generalization on external datasets, including AphasiaBank [17], DementiaBank [18], and TORGO [19], confirming Whisper's capacity to act as a unified pathological speech recognizer. Notably, the model performed well on long-form narrative speech but showed degraded performance on isolated word-level speech, suggesting architectural biases favoring continuous decoding [15].

Building on this foundation, the team extended their work in When Whisper Listens to Aphasia (2024), focusing specifically on Whisper's adaptability to aphasic speech. Using Whisper-large-v2, they conducted dedicated fine-tuning on SONIVA and AphasiaBank, finding consistent outperformance over baseline models such as Wav2Vec2 [20]. This study also highlighted Whisper's sensitivity to syntactic complexity, with recognition accuracy dropping for grammatically intricate utterances, which underscores the need for syntax-aware modeling in ASR systems designed for aphasic patients [21].

While Sanguedolce's team focused on aphasia and general cross-pathology generalization, further extension to dementia speech was pursued by Akinrintoyo, Abdelhalim, and Salomons in WhisperD: Dementia Speech Recognition and Filler Word Detection with Whisper (2025). This study introduced a variant of Whisper-medium that integrates filler-word tokens (e.g., <filler>) into the tokenizer and is fine-tuned on DementiaBank and the CONNECT corpus. The model achieved state-of-the-art performance in terms of WER and also accurately detected disfluencies such as hesitations and interruptions. This work highlights Whisper's flexibility beyond speech recognition, extending to the modeling of fluency impairments, which are a key characteristic of neurodegenerative speech disorders [22].

However, Whisper's cross-lingual generalization remains an open challenge. Mühlhausen et al. [23] tested Whisper-large-en on the German Aphasia Corpus, revealing that zero-shot transfer from English not only failed to improve performance but degraded WERs due to the combined effects of language mismatch and speech pathology. This emphasizes the limitations of naive multilingual transfer and calls for language-specific fine-tuning in clinical ASR. In contrast, Chatzoudis et al. (2022) [24] proposed an alternative approach using Whisper in a zero-shot aphasia detection pipeline. The model extracts language-agnostic transcripts for Greek and French, which are then classified via a BERT-based detector. This study highlights Whisper's potential as a front-end universal transcriber in low-resource settings when paired with robust classifiers.

For fine-grained clinical insights, Wagner et al. [25] proposed Careful Whisper, targeting aphasia subtype classification. Using Whisper-tiny's disfluency-sensitive outputs as input to a Transformer classifier, the authors achieved accurate classification into Broca, Wernicke, and other subtypes. A novel "cautious decoding" strategy was introduced to maintain temporal and semantic coherence, pointing toward Whisper's utility in diagnostic assistance tools.

To address language diversity and accent adaptation, Jiang et al. [26] proposed Perceiver-Prompt, combining Whisper-large with Perceiver IO prompts for Chinese disordered speech. Their framework outperformed traditional fine-tuning on TORGO-CN, suggesting that prompt-based methods can serve as effective low-resource personalization tools in Whisper pipelines.

From a benchmarking perspective, Tang et al. [27] introduced AphasiaBench, a multi-task dataset supporting recognition, alignment, and severity estimation. While Whisper excelled in ASR tasks, it underperformed compared to E-Branchformer in severity classification. This underscores the need to integrate linguistic and acoustic cues beyond transcription for comprehensive clinical support.

Finally, Liu et al. [28] explored parameter-efficient fine-tuning (PEFT) strategies such as LoRA and adapters on Whisper for small-scale disordered speech. Their results showed that lightweight adaptations retained 98% of full-model performance while reducing parameter overhead by 90%, advocating Whisper's deployment on edge devices and bedside systems in real-time low-resource clinical settings.

Together, these studies chart the evolution of Whisper from a general-purpose ASR model to a specialized, efficient, and adaptable backbone for pathological speech recognition, highlighting both its strengths in long-form multilingual transcription and its challenges in cross-lingual transfer, disfluency modeling, and clinical diagnostics.

3.3. Efficient ASR Systems

In recent years, the development of lightweight and efficient ASR systems has become a key driver for practical deployment. Researchers have extensively explored two major directions: model architecture optimization and model compression.

From the perspective of architecture design, the Conformer model combines convolutional networks with the Transformer structure, enabling efficient modeling of both local and global features. It has significantly improved recognition performance on multiple benchmark datasets and has become one of the mainstream ASR model architectures [29]. Building upon this, Squeezeformer further optimizes both the macro and micro architecture of Conformer by introducing a Temporal U-Net structure and efficient downsampling strategies, achieving better performance under fixed computational resources, especially for real-time speech recognition tasks [30].

To address the inference bottlenecks of Transformer architectures with long input sequences, LiteASR focuses on compressing the encoder component of models like Whisper. It introduces a low-rank approximation strategy based on PCA to decompose attention and feed-forward modules, significantly reducing computation while maintaining transcription accuracy, thus achieving a better balance between efficiency and performance [31].

In terms of model compression, DQ-Whisper proposes a joint distillation and quantization framework. By applying dynamic layer matching for distillation and quantization-aware training, it achieves more than a 5× compression ratio for small Whisper models while maintaining recognition performance [32]. Similarly, LightHuBERT leverages the Once-for-All approach to design a supernet with multiple searchable subnetworks. Combined with a two-stage distillation strategy, it reduces the number of parameters by up to 29% while preserving performance across various downstream tasks [33].

To further enhance compression efficiency and avoid accuracy degradation, Xu et al. propose a mixed-precision quantization method that automatically assigns different bit-widths to different modules and integrates quantization with precision-aware training. This method achieves an $8.6\times$ compression ratio without loss in performance, significantly outperforming traditional two-stage approaches [34].

In addition, the Audio Lottery method introduces the Lottery Ticket Hypothesis (LTH) into the ASR field for the first time. It identifies highly sparse subnetworks ("winning tickets") that maintain or even improve

performance across mainstream architectures, demonstrating excellent transferability and robustness to noise [35].

Meanwhile, some studies focus on the efficient modeling of downstream ASR modules. For instance, You et al. propose a CNN-BiLSTM structure for punctuation and word casing prediction. With a model size only 1/40 of a Transformer and 2.5× faster inference, it is well-suited for on-device streaming ASR systems [36].

To provide a comprehensive understanding of ASR efficiency optimization, Ahlawat et al. conduct a systematic review of recent deep learning approaches applied to ASR. Their survey covers key topics such as Conformer, Transformer, self-supervised pretraining, and multilingual modeling, and highlights remaining challenges in low-resource languages and latency optimization [37].

Finally, Fujita et al. propose replacing self-attention in Transformers with lightweight and dynamic convolutions to reduce the quadratic time complexity. This method achieves performance close to Transformer models in various ASR tasks while significantly lowering computational costs [38].

In summary, research on efficient ASR systems follows two main directions: one focuses on designing inherently lightweight architectures, such as Conformer, Squeezeformer, and convolution-based models; the other targets the compression and optimization of existing large models like Whisper and HuBERT using methods such as distillation, pruning, low-rank approximation, and mixed-precision quantization. These efforts lay a solid foundation for deploying ASR systems in embedded and real-time applications.

4

Methodology

4.1. System Overview

To address the inherent challenges in recognizing aphasic speech, this section introduces **AS-ASR**, a lightweight and aphasia-specific Automatic Speech Recognition framework based on the Whisper model, optimized specifically for edge-device deployment and disordered speech recognition. As illustrated in Fig. 4.1, AS-ASR effectively interprets fragmented and disorganized aphasic utterances (e.g., “Cuh... I... brown..water”) as coherent intentions such as “I want a coffee,” contrasting sharply with conventional ASR systems like Wav2Vec, which often produce incomplete or misleading transcriptions. This example underscores the necessity of domain-specific adaptation and intent recovery capabilities when addressing aphasia-affected speech.

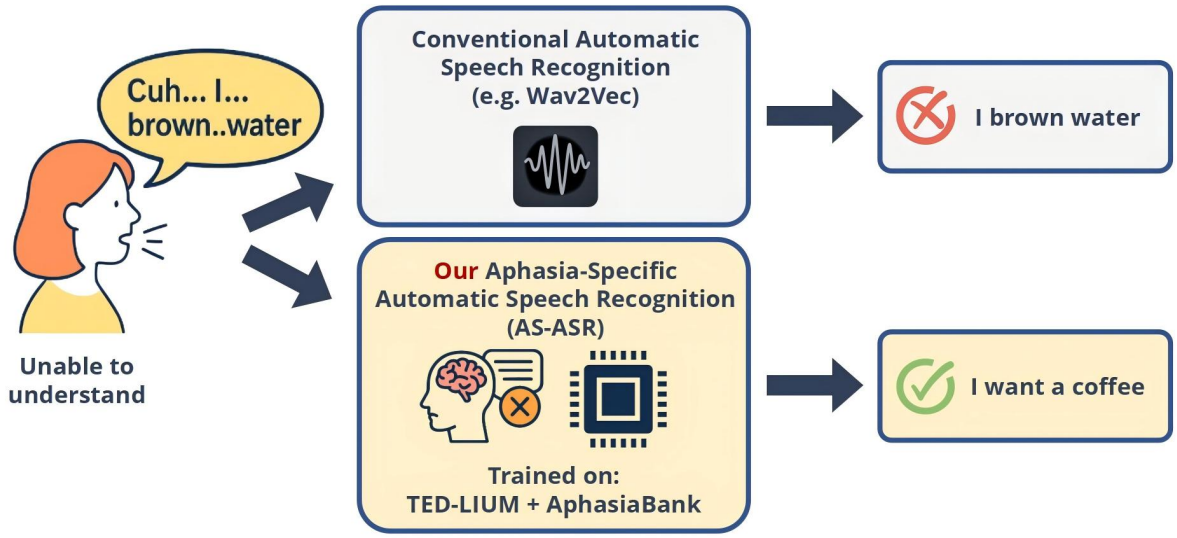

Figure 4.1: Comparison between conventional and aphasia-specific ASR systems on aphasia speech.

The main contributions of our approach are summarized as follows:

1. The introduction of **AS-ASR**, a compact and aphasia-oriented ASR framework leveraging the Whisper architecture, explicitly optimized for efficient deployment on resource-limited edge platforms.
2. The creation of a comprehensive hybrid dataset composed of pathological and normative speech data with adjustable mixing ratios, enabling a thorough evaluation of how training data composition influences ASR performance.

3. The development of a GPT-4-based reference enhancement technique designed to improve the quality and coherence of noisy aphasic speech transcripts, thereby enhancing training effectiveness and boosting model generalization.
4. The implementation of several model optimization strategies, including mixed-precision quantization for memory compression, latency-aware decoding through `max_new_tokens` tuning, and a lightweight energy-based VAD module for improved responsiveness in real-time transcription.

4.2. Efficient Model Selection

This project employs Whisper-tiny-en, the smallest and most lightweight variant in the Whisper model family, containing approximately 39 million parameters. As summarized in Table 4.1, Whisper-tiny-en is a monolingual model specifically trained for English-only tasks, in contrast to the multilingual versions found in larger models like Whisper-base or Whisper-large. This specialization enables more efficient utilization of model capacity, yielding improved accuracy in English-centric applications without the overhead of multilingual modeling.

Table 4.1: Comparison of Whisper model sizes in terms of parameter count, memory requirements, and relative inference speed.

Model Size	Parameters	Required VRAM	Relative Speed
Tiny	39M	$\sim$1 GB	$\sim$10$\times$
Base	74M	$\sim$1 GB	$\sim$7$\times$
Small	244M	$\sim$2 GB	$\sim$4$\times$
Medium	769M	$\sim$5 GB	$\sim$2$\times$
Large	1550M	$\sim$10 GB	1$\times$

The model requires only around 1GB of VRAM, making it highly suitable for deployment on edge computing platforms such as the NVIDIA Jetson Nano, Raspberry Pi with an accelerator, or other low-power AI hardware. In terms of performance, Whisper-tiny-en achieves inference speeds up to 10 times faster than Whisper-large, making it ideal for real-time speech recognition scenarios where both low latency and low power consumption are critical.

Despite its compact size and reduced parameter count, Whisper-tiny-en retains much of the core architecture and design principles of the full Whisper model, including a transformer-based encoder-decoder structure and pretraining on a large corpus of diverse audio-text pairs. This enables it to preserve key capabilities such as handling noisy inputs and producing robust transcriptions under various acoustic conditions.

In this project, we leverage fine-tuning techniques to adapt Whisper-tiny-en specifically to the challenges of aphasic speech recognition, a task characterized by irregular prosody, disfluencies, and atypical phonetic patterns. By tailoring the model to the acoustic and linguistic characteristics of disordered speech, we enable it to outperform general-purpose ASR systems in recognizing pathological speech with both higher accuracy and lower latency. As such, Whisper-tiny-en emerges as a cost-effective and deployable solution for specialized ASR applications in clinical, rehabilitation, and assistive contexts.

4.3. Hybrid Data Fine-tuning Strategy

To enhance the model's adaptability to aphasic speech, we conducted targeted fine-tuning of the pretrained Whisper-tiny-en model using a hybrid training dataset that combines both disordered and fluent speech samples. Specifically, we curated a mixed dataset comprising utterances from individuals with aphasia, sourced from AphasiaBank, and typical speech recordings from the TED-LIUM v2 corpus. This dual-source approach enables the model to simultaneously learn pathological acoustic variations—such as atypical prosody, hesitations, phonemic errors, and articulatory distortions—as well as standard linguistic patterns present in fluent speech.

The motivation for this hybrid training strategy stems from the observation that models trained solely on typical speech data often struggle to generalize to disfluent or impaired speech, especially when

confronted with irregularities in timing, pronunciation, and syntax that deviate from normative patterns. By introducing speech variability directly into the training process, we aim to increase the model's robustness, allowing it to better handle a wider range of speaker profiles and real-world conditions, including noisy environments or spontaneous dialogue.

To further investigate the impact of data composition, we systematically varied the ratio of aphasic to normal speech within the training corpus across multiple experimental configurations. This analysis allowed us to assess how different levels of exposure to disordered speech influence the model's recognition accuracy, especially on unseen aphasic speech samples. Results from these experiments offer insights into how data balancing strategies affect model performance, providing practical guidance for future training pipelines that target low-resource or pathological speech domains.

4.4. Enhanced Aphasia-Centric Dataset Construction

For the aphasic speech corpus, we utilize speech samples drawn from five sub-corpora within the AphasiaBank database, namely the ACWT Corpus[39], Adler Corpus[40], APROCSA Corpus[41], Boston University Corpus[42], and University of Kansas Corpus [43]. These corpora consist of structured clinical interviews and narrative speech tasks conducted with individuals diagnosed with post-stroke aphasia, collected from a range of clinical and research institutions across the United States. Each sub-corpus contains both patient and clinician utterances, carefully transcribed and time-aligned, thus offering a rich linguistic resource for training and evaluating speech recognition models tailored to disordered speech.

To ensure fair and interpretable comparisons, the overall corpus size is strategically aligned with TED-LIUM v2, resulting in a balanced training dataset containing approximately 14,000 segments from each domain. This controlled data balancing facilitates cross-corpus evaluations and allows us to systematically examine how training on pathological versus typical speech affects model generalization and robustness.

A key challenge in using aphasia data lies in the noisy, incomplete, or semantically irregular nature of patient utterances, which may include false starts, neologisms, word substitutions, or unintelligible fragments. Standard transcriptions from AphasiaBank are often annotated using CHAT conventions, embedding symbols that denote nonverbal behaviors, vocal gestures, or annotation notes, which are not directly usable as clean reference texts for supervised learning. To address this, we introduce a GPT-4–based reference enhancement pipeline that transforms raw aphasic utterances into grammatically coherent, semantically faithful textual references, better aligned with the acoustic input while maintaining the speaker's communicative intent.

Concretely, we start by parsing the raw .cha transcription files and extracting dialogue turns between the patient and the interviewer, along with their corresponding time-aligned audio segments. In the next step, we clean the patient's transcripts by removing CHAT-specific markup and non-linguistic annotations (e.g., &=laugh, +..., xxx, etc.), resulting in a clearer representation of the spoken words. These cleaned utterances are then passed into GPT-4, with carefully designed prompts instructing the model to rewrite each utterance in fluent and standard English while preserving the core meaning. The prompt explicitly avoids hallucination or paraphrasing beyond what is implied by the speaker's intent.

The resulting GPT-enhanced references serve as high-quality targets for fine-tuning the Whisper-tiny-en model. This enhancement step acts as a semantic denoising layer, reducing the mismatch between pathological input speech and the textual supervision during training. It not only improves transcription quality but also boosts the model's ability to handle lexical ambiguity, grammatical disfluency, and partial utterances, which are common in aphasic speech. This contributes significantly to the final model's intelligibility, robustness, and clinical usability in real-world applications.

5

Software Implementation: Fine-tuning, Evaluation and Optimization

5.1. Experimental Setup

5.1.1. System Configuration

Fine-tuning was performed on a single NVIDIA RTX A6000 GPU (48GB GDDR6 each) on the Quechua1 high-performance server at the Department of Microelectronics of TU Delft. The Whisper-tiny model was trained for 10 epochs using a per-device batch size of 16. Mixed-precision (fp16) training and gradient checkpointing were enabled to reduce memory footprint and accelerate throughput.

The training pipeline was implemented with the HuggingFace `Trainer` framework, using the AdamW optimizer (learning rate = 5×10^{-5}) and a linear learning rate scheduler. Model evaluation was conducted after each epoch, with checkpoints saved accordingly.

The optimization after fine-tuning and the real-time demo were conducted on a local machine equipped with an AMD Ryzen 9 6900HX CPU and an NVIDIA GeForce RTX 3070 Ti Laptop GPU (8 GB VRAM), running Windows 11 with 32 GB of RAM. The latency measurement is performed entirely on the CPU.

5.1.2. Data Preparation and Preprocessing

AphasiaBank corpus overview

AphasiaBank is a password-protected branch of the TalkBank project that provides multimedia recordings and transcriptions of discourse produced by people with aphasia (PWAs) and age-matched neurologically healthy adults [17]. The collection has grown steadily since its launch in 2007 and currently contains ≈ 290 PWAs and ≈ 190 control participants, drawn from more than ten North American and international clinical sites. All sessions are recorded in both audio and video and then transcribed in CHAT format, which is compatible with CLAN analysis tools.

Figure 5.1 presents a fabricated CHAT snippet that mimics the structure of an AphasiaBank transcript without revealing any protected material. Each interview is transcribed in four aligned tiers: (i) the main tier records orthographic words and inline time-stamps; (ii) the **%mor** tier assigns stem-level part-of-speech tags; (iii) the **%gra** tier encodes dependency relations; and (iv) the **%wor** tier links every token to its precise acoustic span. This layered annotation enables fine-grained analyses ranging from pause distribution to syntactic complexity and is therefore ideally suited for downstream ASR experimentation.

Parsing and Cleaning Raw `.cha` Files

AphasiaBank transcripts are stored in the **CHAT** format (`.cha`). Each file embeds speaker labels, inline time stamps, discourse markers, and revision traces. Before the corpus can be used for modelling,

```
@UTF8
@Begin
@Languages: eng
@Participants:  INV Investigator, PAR Participant
@Media: video
@Date:  10-OCT-2012
@G: Speech
*INV:   I'm going to be asking you to do some talking . NAK6970_9331NAK
%mor:   pro:sub|I~aux|be&1S part|go-PRESP inf|to aux|be part|ask-PRESP pro:per|you inf|to v|do pro:indef|some part|talk-PRESP .
%gra:   1|3|SUBJ 2|3|AUX 3|0|ROOT 4|6|INF 5|6|AUX 6|3|COMP 7|6|OBJ 8|9|INF 9|6|COMP 10|9|OBJ 11|9|XJCT 12|3|PUNCT
%wor:   I'm NAK6970_7090NAK going NAK7210_7470NAK to NAK7530_7610NAK be NAK7650_7710NAK asking NAK7870_8231NAK you NAK8251_8311NAK to NAK8331_8391NAK
        do NAK8451_8531NAK some NAK8611_8791NAK talking NAK8831_9331NAK .
*PAR:   yeah well &=laughs +... NAK9972_11072NAK
%mor:   co|yeah co|well +...
%gra:   1|2|COM 2|0|INCROOT 3|2|PUNCT
%wor:   yeah NAK9972_10212NAK well NAK10572_11072NAK +...
```

Figure 5.1: Fabricated AphasiaBank-style CHAT example

these richly annotated files must be converted into structured *utterance–transcript* pairs [17]. The procedure is summarised below.

1. **Line-by-line parsing and field extraction**
   - *Speaker tag.* A regular expression `^\*([A-Za-z]+):` captures the code at the start of each main tier (e. g. `PAR` for the participant, `INV` for the investigator).
   - *Time stamps.* Inline codes of the form `start_end` are extracted with `(\d+)_(\d+)`, yielding `start_time` and `end_time` in milliseconds. Lines without valid stamps are treated as meta-comments and discarded.
2. **Cleaning of CHAT mark-up**
   - Remove embedded time stamps (e. g. `123_456`), pragmatic tokens such as `&-uh` or `&=laughs`, and revision prefixes beginning with `+`.
   - Delete explanatory text enclosed in square (`[ ... ]`) or round (`( ... )`) brackets.
   - Collapse multiple spaces and trim leading/trailing whitespace to obtain a clean `cleaned_text` string.
3. **Filtering rules**
   - Discard any line for which `start_time` $=$ `end_time` $= 0$ or `cleaned_text` is empty.
   - Retain only lines that map to an actual speech segment, producing a four-field record: `speaker` | `start_time` | `end_time` | `text`.

All of the above steps are implemented in the Python function, which ensures consistent formatting and full reproducibility of the preprocessing pipeline.

LLM-based Generation of Enhanced Reference Texts

Spontaneous speech from people with aphasia (i.e. the `PAR` tiers) is characterised by grammatical disruption, lexical omissions, and phonological errors; using such strings verbatim would enlarge label entropy during fine-tuning.

To regularise the targets, we call **GPT-4** on every patient utterance. The *system prompt* instructs the model to “retain the intended meaning, rewrite concisely, and return `(incomprehensible)` if the utterance cannot be interpreted”, while the *user prompt* takes the form `The speaker said: {raw_text}`. We set the sampling temperature to 0.7 (balancing creativity and stability) and `max_tokens` to 100. Whenever the returned string contains the literal substring `(incomprehensible)`, the sample is discarded; otherwise, the GPT-4 rewrite replaces the original transcript. The resulting quadruple: `speaker`, `start_time`, `end_time`, and `final_text`, is written back to disk. All non-patient turns (e.g. `INV`) keep their original wording so that dialogic coherence is preserved.

Post-processing and Reproducible Data Split

After enhancement, we retain only `PAR` rows whose text does not contain `Incomprehensible`, yielding the directory `Reference_Text_PAR_only_origin`. With a fixed random seed of 42, the utterances are

shuffled and divided into training, development, and test partitions in an 80 / 10 / 10 ratio. Each entry occupies one line of the TSV files `train.tsv`, `dev.tsv` and `test.tsv` and records the fields `file | speaker | start_ms | end_ms | text`. The TSV files and their corresponding audio fragments are placed in a well-defined folder hierarchy, which simplifies batch loading and version control.

Data Partitioning

For every targeted proportion of pathological to normal speech, including 10 : 90, 30: 70, 50 : 50, 70 : 30, and 100 : 0, we impose the same absolute dataset sizes to keep the downstream experiments strictly comparable. Concretely, each configuration contains 14,000 utterances in total, partitioned into 11,200 segments for training (80%), 1,400 for validation (10%), and 1,400 for held-out testing (10%). Within a given split, the pathological (AphasiaBank) and normal (TED-LIUM) portions follow the prescribed ratio while summing to the fixed quota.

Table 5.1 enumerates the exact counts. As an example, the 10 % : 90 % condition comprises 1,120 aphasic utterances and 10,080 normal utterances in the training set, plus 140 / 1,260 and 140 / 1,260 aphasic/normal utterances in the validation and test sets, respectively.

By holding the overall cardinality constant and varying only the class mixture, we can attribute any performance differences solely to the relative abundance of pathological versus normative speech, rather than to dataset size effects.

Table 5.1: Data distribution under different Aphasia:TED-LIUM mixing ratios. Each configuration totals 14,000 samples, partitioned into training (80%), development (10%), and test (10%) sets.

Ratio	Aphasia			TED-LIUM		
	Train	Dev	Test	Train	Dev	Test
10%:90%	1120	140	140	10080	1260	1260
30%:70%	3360	420	420	7840	980	980
50%:50%	5600	700	700	5600	700	700
70%:30%	7840	980	980	3360	420	420
90%:10%	10080	1260	1260	1120	140	140

Audio Alignment and Unified Format

Accurate one-to-one correspondence between audio and transcript is established by cutting each interview waveform into single-utterance clips with `ffmpeg` (or `pydub`), using the millisecond boundaries stored in the TSV. Each clip is named `<id>_<start>_<end>.wav`. All files are then resampled to 16 kHz, quantised at 16-bit PCM, and converted to mono.

During training and inference, the loader uses the file name as a key, ensuring that the acoustic signal and the textual label are always retrieved in tandem. By enforcing identical acoustic characteristics and temporal alignment across AphasiaBank and TED-LIUM, we remove confounding factors due to format discrepancies and provide Whisper fine-tuning with a reliable input stream.

5.1.3. Evaluation Metrics

We adopt the standard **Word Error Rate (WER)** as the primary metric for evaluating ASR performance. WER quantifies the minimum number of word-level edit operations-substitutions (S), deletions (D), and insertions (I)-required to align a model hypothesis with the corresponding reference transcript of length N:

$$\text{WER} = \frac{S + D + I}{N} \times 100\% \tag{5.1}$$

All evaluations are conducted on the same speaker-disjoint test sets to ensure fair and consistent comparisons across systems. Lower WER values indicate more accurate transcriptions.

5.1.4. Transcript Normalisation

To avoid inflating WER due to orthographic or stylistic mismatches, all reference and hypothesis texts are passed through a normalisation function prior to scoring. This process includes:

- converting all text to lowercase;
- replacing hyphens with spaces;
- removing bracketed content (e.g., annotations in `[...]`, `(...)`), filler words (e.g., `uh`, `um`, `hmm`);
- fixing spacing issues around apostrophes;
- expanding contractions such as `can't` → `cannot`, `you're` → `you are`;
- removing punctuation and commas in numbers (e.g., `1,000` → `1000`);
- converting numerals (including decimals) into English words using a number-to-words utility;
- standardising British to American spellings (e.g., `colour` → `color`).

The final output is stripped of redundant whitespace and punctuation, ensuring that surface-form mismatches do not artificially penalise models during WER computation. The normalisation logic is implemented as part of a preprocessing function, which is applied uniformly to both reference and predicted transcripts before alignment.

5.2. Baseline System

To establish meaningful benchmarks for our proposed methods, we define two baseline systems that capture different aspects of the Whisper model's performance under varying conditions and configurations:

Baseline-1 assesses the zero-shot inference capability of the pre-trained Whisper-`Tiny.en` model across three distinct evaluation scenarios: (i) the TED-LIUM test set, representing typical and fluent speech from public talks; (ii) the Aphasia test set, consisting of spontaneous speech from individuals with post-stroke aphasia; and (iii) a merged test set combining samples from both TED-LIUM and Aphasia corpora. This configuration is designed to evaluate the model's inherent generalization ability to both in-domain (fluent) and out-of-domain (pathological) speech without any task-specific fine-tuning. The results from this baseline serve as a reference point to measure how much performance gain can be achieved through adaptation strategies.

Baseline-2 focuses exclusively on the recognition of pathological speech and reports the average WER obtained from a suite of Whisper model variants-including Tiny, Base, Small, and Medium-when evaluated solely on the Aphasia test set. This baseline provides insights into how model scale influences recognition performance in the presence of disordered linguistic patterns, acoustic irregularities, and reduced intelligibility common in aphasic speech. By comparing across different model sizes, we can observe trends in robustness, offering valuable guidance for selecting models under resource constraints or deployment scenarios.

5.3. Fine-tuning and Results

5.3.1. Baseline Performance

Table 5.2 presents the average WER achieved by five pre-trained Whisper model variants when evaluated on the pure aphasic speech test set (Baseline-2). These results offer an initial view of how model size and architecture affect recognition performance in the context of disordered, non-normative speech patterns.

Among all evaluated configurations, the `Whisper-Large` model achieved the best performance with an average WER of 0.622. This confirms its superiority in handling challenging input, likely owing to its significantly larger parameter count and multilingual training. The next two best-performing models were `Medium.en` (WER = 0.677) and `Small.en` (WER = 0.686), both of which demonstrate a strong trade-off between accuracy and model size, though their relative improvement over each other is marginal.

Interestingly, the `Base.en` model yielded the highest WER of 0.886, performing even worse than the much smaller `Tiny.en` model (WER = 0.787). This counterintuitive result suggests that certain model

configurations may be particularly brittle when applied to pathological speech, potentially due to overfitting to normative prosody or phonotactic patterns during pretraining. It also underscores that larger model sizes do not linearly translate into better robustness for atypical input distributions such as aphasia.

Considering both accuracy and computational efficiency, the `Tiny.en` model presents a compelling baseline. Despite having only 39 million parameters and requiring less than 1 GB of GPU memory, it achieves a moderate WER that is competitive within the mid-tier models. This makes it a strong candidate for real-time deployment on edge devices with limited compute and energy budgets, such as embedded medical hardware or low-power mobile applications.

Table 5.2: Baseline-2 WER Results on Aphasia Datasets by different Whisper models.

Model Size	Average WER
`Tiny.en`	0.787
`Base.en`	0.886
`Small.en`	0.686
`Medium.en`	0.677
`Large`	0.622

5.3.2. Fine-tuning Results

Following the baseline evaluation of multiple Whisper variants (Table 5.2), we selected `Whisper-Tiny.en` as the base model for further fine-tuning, owing to its favourable trade-off between recognition accuracy and computational efficiency. The model was trained on a hybrid dataset composed of TED-LIUM and AphasiaBank utterances in a 1:1 ratio and evaluated under three distinct test conditions: TED-LIUM only, Aphasia only, and a merged test set combining both domains. The results before and after fine-tuning are presented in Table 5.3.

In the zero-shot setting (Baseline-1), `Whisper-Tiny.en` achieves strong performance on clean, fluent speech, with WERs of 0.119 and 0.117 on the TED-LIUM development and test sets, respectively. However, its performance deteriorates significantly when evaluated on pathological speech, where WERs increase to 0.808 (dev) and 0.755 (test). This drop reflects the model's limited generalisation ability to disordered speech patterns, which deviate substantially from the distributions observed during pretraining. On the merged test set, which simulates real-world acoustic conditions involving a mix of fluent and aphasic speech, the baseline model struggles as well, with WERs of 0.515 (dev) and 0.498 (test), highlighting its lack of domain adaptability.

After fine-tuning on the mixed-domain training data, the model exhibits marked improvements. On TED-LIUM speech, WER slightly improves to 0.116 on both dev and test sets, indicating that the additional exposure to aphasic speech does not degrade its performance on fluent input. This stability is crucial for maintaining performance in clinical scenarios where speech characteristics can vary considerably within and across speakers.

More importantly, the fine-tuned model shows substantial gains on the AphasiaBank test set. The WER is reduced from 0.808 to 0.430 on the dev set (a relative improvement of 46.8%) and from 0.755 to 0.454 on the test set (a 39.9% improvement). These reductions represent meaningful improvements in transcription quality for aphasic speech, which is typically characterised by fragmented, grammatically irregular, and phonologically distorted utterances.

The largest relative improvement is observed on the merged evaluation set. Fine-tuning reduces the WER from 0.515 to 0.162 on the dev set (a 68.5% drop) and from 0.498 to 0.170 on the test set (a 65.9% drop). This indicates that the model not only adapts effectively to disordered speech but also becomes more robust to acoustic and linguistic domain shifts, a key requirement for real-world deployment in clinical or assistive ASR applications.

Overall, these results demonstrate that task-specific fine-tuning significantly enhances the model's ability to transcribe aphasic speech while preserving its accuracy on fluent input. The fine-tuned

`Whisper-Tiny.en` model thus emerges as a practical candidate for real-time deployment in clinical settings, combining compact architecture, low latency, and robust multi-domain transcription capability.

Table 5.3: WER of Baseline-1 and Fine-tuned `Whisper-Tiny.en` on TED-LIUM and Aphasic Speech

Model	Evaluate Dataset	Dev WER	Test WER
Whisper-`Tiny.en` (Baseline-1)	TED-LIUM only	0.119	0.117
	Aphasia only	0.808	0.755
	Merged	0.515	0.498
Fine-tuned Whisper-`Tiny.en`	TED-LIUM only	0.116	0.116
	Aphasia only	0.430	0.454
	Merged	0.162	0.170

5.3.3. Impact of Data Ratios

We further investigated how varying the mixing ratios of aphasic and normal speech data (Aphasia:TED-LIUM = 10:90, 30:70, 50:50, 70:30, and 90:10) affects the performance of the fine-tuned Whisper-`Tiny.en` model. As shown in Fig. 5.2, we evaluated the WER on both aphasic and TED-LIUM development and test sets under each training composition.

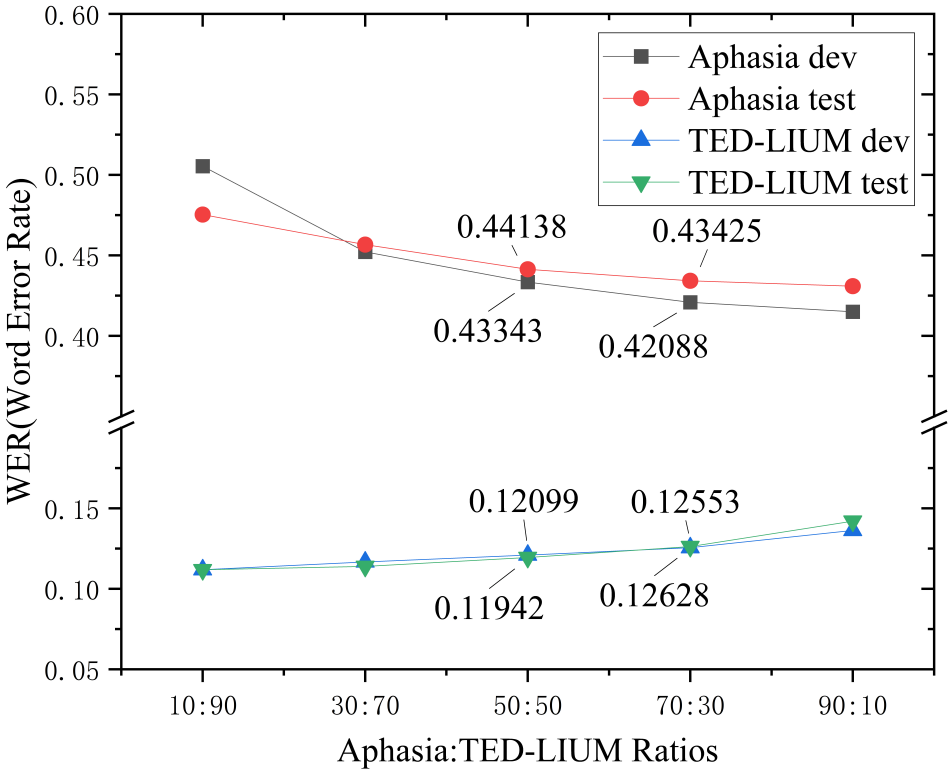

Figure 5.2: Impact of Aphasia-to-TED-LIUM ratios on WER performance across dev/test sets.

The results demonstrate a clear trade-off between recognition performance on pathological and fluent speech domains as the data ratio shifts. Specifically, as the proportion of aphasic speech in the training

data increases, WER on the Aphasia test set decreases consistently (from 0.474 at 10:90 to 0.434 at 90:10). A similar downward trend is observed on the Aphasia development set, with WER improving from 0.503 at 10:90 to 0.421 at 90:10. These findings confirm that exposing the model to more disordered speech improves its ability to transcribe such utterances, enhancing its robustness in clinical contexts.

Conversely, the model's performance on clean TED-LIUM speech shows mild degradation as aphasic data dominates the training mix. The WER on TED-LIUM test data rises from 0.120 at 10:90 to 0.126 at 90:10, while the dev set similarly increases from 0.110 to 0.126. Although the deterioration is relatively modest compared to the gains in aphasic recognition, it suggests a domain specialization effect, in which training biased toward one speech type slightly compromises the model's ability to generalize to another.

Interestingly, a 50:50 or 70:30 ratio appears to strike a favorable balance, with WER on Aphasia test data reduced to 0.433 and 0.434 respectively, while TED-LIUM test WERs remain within 0.125. This indicates that balanced or moderately aphasia-skewed compositions are optimal for building domain-robust models that can generalize well across both disordered and fluent speech without severe trade-offs.

Overall, these results underscore the sensitivity of multi-domain ASR systems to training data distribution. Designing training sets with appropriate domain diversity is essential for achieving robust and equitable performance across heterogeneous user populations.

5.3.4. Error Pattern Analysis in Aphasic Transcriptions

To better understand the performance limitations of the fine-tuned Whisper-tiny model on aphasic speech, we conducted a detailed error pattern analysis focusing on samples with relatively high WER. After fine-tuning, we analyzed the transcription outputs of the model on both the development and test sets to identify representative examples with high WER. These transcription mismatches reveal meaningful insights into the model's behavior under challenging speech conditions.

Table 5.4 organizes representative examples into five main categories: substitution/insertion/deletion errors, syntactic and semantic disruption, confusion of negation and function words, phonetic approximations, and the truncation of pronouns or proper nouns. Each category highlights a different aspect of the model's difficulty in processing disfluent or irregular language.

For instance, fragmented sentence structures often lead to repeated words or omitted content, while atypical prosody can confuse the model into selecting phonetically similar but incorrect terms. In some cases, critical functional elements such as negation are misrecognized, resulting in reversals of intended meaning. Moreover, utterances that trail off or include ellipses frequently cause named entities or key arguments to be dropped entirely from the output. By examining these patterns, we gain a clearer picture of where the model fails to generalize, reinforcing the need for domain-specific fine-tuning and structural robustness in ASR systems designed for disordered speech.

The analysis of high-WER cases reveals several recurring issues in the model's transcription of aphasic speech. The system often has difficulty handling disfluencies, repeated words, and fragmented grammar. These features frequently result in omissions, insertions, or incorrect substitutions. The model also tends to misinterpret sentences that include negation or inverted word order, which can significantly change the intended meaning. For longer or more complex utterances, the output sometimes becomes oversimplified or contains invented content, indicating that the model struggles with extended speech segments.

To improve performance, future work could involve training on synthetic speech that mimics aphasic patterns in a controlled and consistent manner. Including prosodic features such as pause duration, pitch, and speech rhythm may help the model better understand the flow of disordered speech. It is also worth exploring simple rule-based post-processing methods to correct recurring errors, especially those involving negation, without adding significant computational cost.

These findings highlight the importance of tailoring speech recognition systems to the characteristics of neurologically impaired speech. Adapting both the training data and the model design can lead to more reliable and practical ASR solutions in clinical and assistive settings.

Table 5.4: Error Examples by Category in Aphasic Speech Transcription

Error Type	Reference	Prediction	Notes
Substitution, Insertion, Deletion	*i had a bit of difficulty articulating many main points*	*i had little little of trouble withulating things strokes strokes like*	Multiple substitutions, insertions, and phonemic confusion
	i got divorced many years ago i don't know exactly maybe twenty or thirty years or possibly even more than that	*i have divorced many many ago i know know for what twenty thirty thirty or or so more that that*	Repetitions suggest model mimics perseveration
Syntax and Semantic Disruption	*somehow either a genie a fairy godmother or someone else managed to place a house there put a pressed dress there and then had the chance to attend the ball*	*somehow a a genie or fairy godmother or someone else could to go the dress there and a dress dress there and then go to ball to go the ball*	Severe semantic drift and fragmentation
	he simply lowered his head because he understood the situation	*just just staring his face a he knows what exact*	Loss of meaning due to syntactic breakdown
Negation and Function Word Confusion	*you don't have to do it*	*you will have to do it*	Missing negation causes semantic reversal
	i cannot do everything	*i cannot do it*	Loss of scope of limitation
Phonetic and Lexical Approximation	*peanut butter*	*wellanut butter*	Phonetic similarity without lexical accuracy
	paintbrush	*pain the*	Fragmented phrase suggests acoustic confusion
Pronoun and Proper Noun Truncation	*they have to call the firefighters to get him out of the tree*	*they have to call to fire to get the to of the tree*	Entity confusion and omission
	cinderella went with the people	*cinderella went with the*	Truncated named entity and object

5.4. Model Quantization

To reduce resource consumption while maintaining transcription quality, we applied mixed-precision quantization to the fine-tuned Whisper-Tiny model. This section details the quantization strategy, the design rationale, and the observed performance improvements and trade-offs. All evaluations were conducted using a 33-second English audio sample as a representative test case.

5.4.1. Mixed-Precision Quantization Strategy

The mixed-precision approach is designed to balance memory efficiency and inference accuracy by combining INT8 quantization with FP16 half-precision representation. The quantization process consists of the following three key steps:

INT8 Weight Quantization

All linear layer weights in the model were quantized to 8-bit integers using PyTorch's built-in dynamic quantization API, *torch.quantization.quantize_dynamic*. This method performs post-training quantization, which does not require any re-training or calibration dataset and is especially suited for models where most of the computation occurs in *nn.Linear* layers, as in the case of Whisper's Transformer-based architecture.

Dynamic quantization works by replacing specified floating-point modules (typically *nn.Linear*, and optionally *nn.LSTM* or *nn.GRU*) with quantized counterparts. During inference, the quantized model dynamically quantizes the input activations to INT8 on-the-fly, while the model weights are already pre-quantized to INT8. The core multiply-accumulate operations are then performed using integer arithmetic, which is significantly more efficient in terms of memory bandwidth and compute on supported hardware (e.g., AVX2 or AVX512 instructions on CPUs, and dequantization pipelines on GPUs).

In our implementation, we extended this idea by manually replacing each linear layer with a custom quantized module that stores weights in INT8 and biases in FP16, followed by computation in FP16. This hybrid format allows better GPU acceleration and aligns with the GPU's native support for half-precision matrix operations. Additionally, converting the quantized weights back to FP16 during inference enables us to preserve a balance between numerical stability and compute efficiency.

By using dynamic quantization, the original floating-point weights are not stored in memory at runtime, leading to a substantial reduction in model size and memory footprint, without requiring access to the training data or re-finetuning the model.

FP16 Biases and Computation

While weights are stored in INT8, biases and activations were retained in FP16 format. During inference, INT8 weights are dequantized to FP16 for multiply-accumulate operations, allowing the GPU to leverage its native half-precision acceleration (e.g., Tensor Cores on NVIDIA GPUs).

Simplified Activation Functions and FP16 Conversion

To further optimize the inference efficiency, all GELU (Gaussian Error Linear Unit) activation functions in the Whisper-Tiny model were replaced with ReLU (Rectified Linear Unit). This substitution was motivated by the observation that GELU, although offering smoother activation and better gradient flow in large-scale training, is computationally more expensive during inference due to its complex formulation.

The GELU function is defined as:

$$\mathsf{GELU}(x) = x \cdot \Phi(x) = x \cdot \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution, and $\operatorname{erf}(\cdot)$ is the Gaussian error function. This expression involves exponential and integral approximations, which are computationally intensive.

In contrast, the ReLU function is defined as:

$$\mathsf{ReLU}(x) = \max(0, x)$$

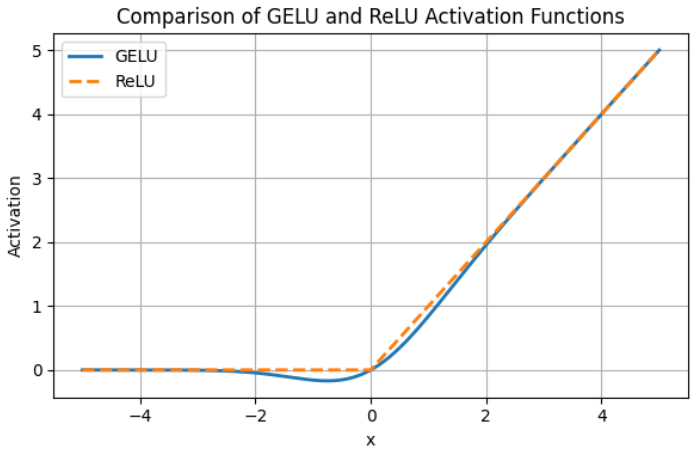

Figure 5.3: Comparison between GELU and ReLU activation functions.

which requires only a simple comparison operation and can be efficiently executed on modern hardware.

As we can see from Fig. 5.3, the function curves of GELU and ReLU differ primarily in their smoothness around $x = 0$. While GELU produces a soft activation and allows small negative values to pass through-beneficial for gradient flow during training-ReLU introduces sparsity by zeroing out all negative inputs. This sparsity reduces computation and memory bandwidth usage during inference.

By replacing GELU with ReLU in the inference pipeline, we eliminate the overhead of computing the error function and gain considerable speed-ups, particularly on edge devices and GPUs where branchless or fused ReLU operations are hardware-accelerated. Additionally, all remaining parts of the model-such as LayerNorm operations, residual additions, and attention projections-were converted to FP16 using the `.half()` casting method in PyTorch. This ensures that all tensor operations are performed in a consistent low-precision format, allowing the GPU to fully utilize its tensor cores for high-throughput inference. The combination of ReLU simplification and FP16 conversion contributes significantly to reducing overall latency and memory consumption.

5.4.2. Performance Evaluation

A comparative evaluation was performed between the original FP32 model and the quantized mixed-precision model. As shown in Table 5.5, the mixed-precision version achieves substantial reductions in both memory usage and model size, while incurring only a moderate increase in inference latency.

Table 5.5: Comparison between FP32 and Mixed-Precision Quantized Whisper-Tiny Models.

Quantization Type	Latency	Memory Usage	Model Size
`FP32`	616.5 ms	191.29 MB	144.11 MB
`Mixed-Precision`	769.1 ms	93.95 MB	40.49 MB

These results demonstrate that the proposed quantization strategy effectively reduces the Whisper-Tiny model's resource footprint, making it more suitable for real-time or embedded deployment. Specifically, the model size is reduced from 144.11 MB to 40.49 MB, representing a compression of approximately 71.9%, which is particularly beneficial for storage-constrained environments such as mobile devices or embedded platforms.

In addition to storage savings, the memory usage during inference is cut nearly in half, decreasing from 191.29 MB to 93.95 MB. This reduction significantly eases memory bandwidth and cache pressure, potentially allowing the model to run on devices with limited RAM or enabling parallel inference on shared-memory systems. While the inference latency increased modestly from 616.5 ms to 769.1

ms (a relative increase of 24.7%), this trade-off is generally acceptable, especially considering the significant gains in model compactness and memory efficiency. The slight increase in latency can be attributed to the overhead of runtime dequantization and precision conversion (e.g., INT8 to FP16), as well as changes in activation and computation patterns following the GELU-to-ReLU replacement.

Furthermore, this quantization approach maintains the model's architectural compatibility and can be modularly extended. For instance, integrating ONNX Runtime with hardware-specific inference backends (such as NVIDIA TensorRT or Intel OpenVINO) can help recover or even exceed the original latency performance while preserving the quantization-induced memory and size benefits.

Overall, the mixed-precision quantized Whisper-Tiny model provides a compelling trade-off: it retains practical inference capability while dramatically lowering deployment costs. This makes it a viable candidate for speech recognition applications in edge computing, wearable devices, and low-power scenarios where full-precision models are impractical.

5.5. Exploration of parameter max_new_tokens in Handling Long Audio

Whisper and similar generative models operate in an auto-regressive manner, meaning they generate one token at a time and repeatedly invoke the decoder to predict the next token based on the current context. The parameter *max_new_tokens* determines the upper bound on how many tokens can be generated in a single decoding segment.

A smaller value of *max_new_tokens* results in shorter generation bursts, requiring the decoder to be called more frequently, while a larger value allows the model to run longer uninterrupted, potentially increasing efficiency. However, each decoding step still requires computing attention, generating probability distributions, and sampling or selecting the next token. As such, the decoding loop becomes heavier and more time-consuming when *max_new_tokens* is large, especially in a streaming setup.

To quantify the latency impact of different *max_new_tokens* settings, we conducted experiments using a single long-form input: a 20-minute and 42-second English audio clip from the TED-LIUM corpus. The Whisper-Tiny model was configured to transcribe this audio using four different values of *max_new_tokens*: 64, 128, 256, and 448. All other decoding parameters were held constant.

Table 5.6 summarizes the results. Whisper processes long audio by segmenting it into overlapping 30-second windows. In this experiment, the 20min42s audio was split into 42 segments.

Table 5.6: Impact of *max_new_tokens* on transcription results and latency for a 20min42s TED-LIUM audio clip.

Max_new_tokens	Total Segments	Total Tokens	Total Latency
64	42	2606	15973.72 ms
128	42	3567	21248.32 ms
256	42	3567	21442.64 ms
448	42	3567	21540.56 ms

As shown in the table, when *max_new_tokens* is set to 64, the total number of generated tokens is significantly lower than in the other settings. This is because some 30-second audio segments naturally contain more than 64 tokens. When the token generation hits the ceiling of 64, decoding halts prematurely, resulting in truncated transcription. This leads to both **incomplete output** and lower total token counts.

In contrast, the settings of 128, 256, and 448 yield identical token outputs (3567 tokens) because nearly all 30-second audio segments in this dataset require fewer than 128 tokens for full transcription. These values are therefore **sufficient** to ensure that each segment is transcribed completely without early termination. However, latency increases as the value of *max_new_tokens* increases. This is expected, as higher values require longer attention computations and slower output sampling loops per decoding cycle.

As a result, if *max_new_tokens* is set too small, the output may be truncated, particularly for longer or denser speech segments. On the other hand, setting it too large introduces inefficiencies: it may waste computational resources, increase latency, and risk repeated outputs or even infinite loops in edge cases.

For **normal English speech**, the average number of tokens generated per second typically falls between **2.5 and 3.5**. For speakers with speech impairments, such as aphasia patients, this rate is generally slower. In the context of real-time streaming transcription, where the input refreshes every 3 seconds, setting *max_new_tokens* to 12 is sufficient to cover the full range of token demands. This conservative setting enables both efficient and responsive decoding without introducing truncation or excess latency.

5.6. VAD Testing

In real-time speech transcription systems, it is essential to detect whether an incoming audio segment contains speech or not. This process, known as Voice Activity Detection (VAD), helps avoid unnecessary computation when the input is silent and prevents the model from generating irrelevant or misleading transcriptions. VAD is particularly important in edge or low-latency applications where responsiveness and resource efficiency are critical.

Without VAD, the system attempts to transcribe every incoming audio buffer, including silent or noisy segments. This can lead to:

- Wasted computational resources.
- Generation of meaningless outputs (e.g., “I don’t know” or silence tokens).
- Reduced overall system efficiency and increased latency.

Therefore, incorporating a lightweight and effective VAD mechanism is a practical step toward building a more intelligent and responsive transcription pipeline.

5.6.1. Common VAD Methods

Traditional VAD algorithms can be broadly categorized into:

- **Energy-based VAD:** Simple statistical methods that compare the energy (or amplitude) of the signal against a predefined threshold.
- **Spectral-based VAD:** These methods analyze spectral features such as zero-crossing rate, spectral entropy, or band-specific energy.
- **Machine Learning-based VAD:** Advanced models trained on large speech corpora, including WebRTC VAD and neural network-based classifiers.

While machine learning-based methods generally achieve higher robustness, they also introduce additional computational overhead, which may not be acceptable in real-time or resource-constrained settings.

5.6.2. Proposed Implementation

In this project, we implement a simple yet effective energy-based VAD strategy to support real-time transcription with the fine-tuned Whisper-Tiny model. For each incoming audio segment, the system computes the mean absolute amplitude and compares it against a predefined threshold. Segments with energy above the threshold are classified as containing speech and are passed to the transcription model, while silent segments are ignored.

This method offers an efficient trade-off between complexity and responsiveness. It is lightweight enough to be integrated into streaming pipelines with minimal delay, and it eliminates silent inputs that would otherwise trigger unnecessary decoding operations. Although not suitable for noisy or low-SNR environments, this approach is sufficient for most clean speech scenarios and can serve as a baseline for future integration of more robust VAD techniques.

5.7. Real-time Transcription Demo

To demonstrate the effectiveness of the proposed VAD mechanism in practical scenarios, a real-time transcription demo was implemented using the fine-tuned Whisper-Tiny model. The system continuously captures live microphone input and processes it in 3-second audio segments, simulating a realistic streaming setup where the transcription pipeline refreshes every few seconds.

Each 3-second segment is first evaluated by the energy-based VAD module. Only if the signal energy exceeds the predefined threshold is the segment forwarded to the Whisper model for decoding. This setup ensures that silent intervals do not trigger unnecessary computation or produce irrelevant transcriptions.

This lightweight demo illustrates the feasibility of deploying VAD-enhanced ASR on local machines with constrained resources. It also highlights the importance of segment-wise voice detection in reducing latency and improving the user experience during continuous speech monitoring or dialogue systems.

To emulate realistic streaming input, the test used a 3-second English speech clip sampled from the LibriSpeech test-clean dataset. The inference latency for each segment was measured to be approximately **182.1 ms**. In a typical ASR system design, latency below **200 ms** is generally considered suitable for real-time interaction, as it allows for near-immediate feedback without perceptible delay for users.

5.8. Conclusion

This chapter presented the complete software pipeline for adapting and optimizing the Whisper-tiny model for aphasic speech recognition, with a primary focus on fine-tuning strategies. At the core of this work is the recognition that general-purpose ASR models struggle to handle the irregular, fragmented, and often unpredictable nature of aphasic speech. To address this, we conducted targeted fine-tuning using a carefully prepared hybrid dataset combining normative speech from TED-LIUM and disfluent utterances from AphasiaBank.

Significant effort was devoted to preprocessing the aphasic data, including transcript alignment, cleaning, and refinement using GPT-4-assisted corrections. These steps ensured high-quality supervision signals even when the original speech was heavily disrupted. Through a series of experiments, we evaluated the effect of different aphasia-to-typical speech ratios and confirmed that an aphasia-enriched training setup substantially improved transcription accuracy on disordered speech, without compromising performance on fluent utterances.

To support deployment in real-time and resource-constrained settings, we further introduced several runtime optimizations. Mixed-precision quantization, combining INT8 weights with FP16 operations, helped reduce model size and memory usage while maintaining acceptable latency. We also explored the impact of decoding parameters such as `max_new_tokens` and proposed a lightweight VAD mechanism to reduce unnecessary computation during silent periods.

In summary, fine-tuning the Whisper-tiny model on domain-specific aphasic data significantly enhanced its ability to handle disfluent and neurologically impaired speech. The additional optimizations made it suitable for edge deployment, reinforcing the model's practical value in assistive and clinical contexts. This work demonstrates the critical role of task-specific adaptation in building inclusive ASR systems and provides a strong foundation for further research into robust, low-resource speech technologies.

6

Conclusion and Future Work

6.1. Conclusion

This thesis presented **AS-ASR**, a lightweight and aphasia-specific automatic speech recognition (ASR) system built upon the Whisper-tiny architecture, with the goal of enabling real-time and offline speech transcription for individuals with disordered speech in resource-constrained environments.

Through targeted fine-tuning on a balanced hybrid dataset composed of aphasic and fluent speech, the system achieved a substantial reduction in WER on aphasic test data from 0.755 to 0.454, which confirms the effectiveness of domain-specific adaptation. Importantly, this performance gain was achieved with only minimal degradation in recognition accuracy on fluent TED-LIUM speech, thereby demonstrating the model's ability to generalize effectively across heterogeneous speech domains.

To support lightweight deployment, we applied mixed-precision quantization (INT8 weights and FP16 biases) to reduce model size and memory footprint. In addition, we adjusted the `max_new_tokens` parameter to lower inference latency for long-form audio inputs. An energy-based VAD module was integrated into the real-time transcription demo, enabling the system to skip silent segments and improve overall efficiency in streaming scenarios.

6.2. Future Work

While AS-ASR demonstrates significant progress in aphasia-specific transcription, several avenues remain open for further exploration. One important direction is the development of lightweight error correction and post-processing techniques to address common transcription problems such as negation errors, syntactic disfluencies, and semantic inconsistencies. In addition, incorporating speaker-specific adaptation through methods like on-device fine-tuning or embedding personalization modules may help improve recognition accuracy for users with more severe or atypical speech patterns.

Beyond improving model performance, it is also important to evaluate the system in real-world conditions. Future work could explore the integration of prosodic or visual cues to enhance robustness in low-articulation scenarios. Deploying the system on edge devices such as Jetson Nano or Raspberry Pi and conducting clinical evaluations with feedback from therapists and patients will be essential for assessing usability, latency, and privacy. Finally, expanding the framework to support other languages or dialects through multilingual fine-tuning or prompt-based transfer would increase accessibility for a broader population.

In summary, this work demonstrates that compact, domain-adapted ASR models such as AS-ASR can meaningfully improve speech accessibility for people with aphasia, while remaining efficient enough for real-time use on constrained hardware. The framework lays a strong foundation for future research at the intersection of speech technology, clinical support, and inclusive AI design.

References

[1] Dayne Freitag and Andrew McCallum. “Information extraction with HMMs and shrinkage”. In: *Proceedings of the AAAI-99 workshop on machine learning for information extraction*. Orlando, Florida. 1999, pp. 31–36.

[2] William B Cavnar, John M Trenkle, et al. “N-gram-based text categorization”. In: *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*. Vol. 161175. Las Vegas, NV. 1994, p. 14.

[3] Alex Graves et al. “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks”. In: *Proceedings of the 23rd international conference on Machine learning*. 2006, pp. 369–376.

[4] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[5] Alec Radford et al. “Robust speech recognition via large-scale weak supervision”. In: *International conference on machine learning*. PMLR. 2023, pp. 28492–28518.

[6] A. R. Damasio. “Aphasia”. In: *New England Journal of Medicine* 326.8 (1992), pp. 531–539.

[7] Cleveland Clinic. *Aphasia: Causes, Symptoms & Treatment*. 2022. URL: `https://my.clevelandclinic.org/health/diseases/5502-aphasia` (visited on 06/09/2025).

[8] Kathleen C Fraser et al. “Automated classification of primary progressive aphasia subtypes from narrative speech transcripts”. In: *cortex* 55 (2014), pp. 43–60.

[9] Do Hyung Kim et al. “Usefulness of Automatic Speech Recognition Assessment of Children With Speech Sound Disorders: Validation Study”. In: *Journal of Medical Internet Research* 27 (2025), e60520.

[10] Katerina A Tetzloff et al. “Automatic Speech Recognition in Primary Progressive Apraxia of Speech”. In: *Journal of Speech, Language, and Hearing Research* 67.9 (2024), pp. 2964–2976.

[11] Mengzhe Geng et al. “Investigation of data augmentation techniques for disordered speech recognition”. In: *arXiv preprint arXiv:2201.05562* (2022).

[12] Zengrui Jin et al. “Adversarial data augmentation for disordered speech recognition”. In: *arXiv preprint arXiv:2108.00899* (2021).

[13] Jimmy Tobin et al. “Automatic speech recognition of conversational speech in individuals with disordered speech”. In: *Journal of Speech, Language, and Hearing Research* 67.11 (2024), pp. 4176–4185.

[14] Jimmy Tobin, Katrin Tomanek, and Subhashini Venugopalan. “Towards a single asr model that generalizes to disordered speech”. In: *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2025, pp. 1–5.

[15] Giulia Sanguedolce et al. “Universal Speech Disorder Recognition: Towards a Foundation Model for Cross-Pathology Generalisation”. In: *Advancements In Medical Foundation Models: Explainability, Robustness, Security, and Beyond*.

[16] Zhaopeng Qian, Kejing Xiao, and Chongchong Yu. “A survey of technologies for automatic Dysarthric speech recognition”. In: *EURASIP Journal on Audio, Speech, and Music Processing* 2023.1 (2023), p. 48.

[17] Brian MacWhinney et al. “AphasiaBank: Methods for studying discourse”. In: *Aphasiology* 25.11 (2011), pp. 1286–1307.

[18] Alyssa M Lanzi et al. “DementiaBank: Theoretical rationale, protocol, and illustrative analyses”. In: *American Journal of Speech-Language Pathology* 32.2 (2023), pp. 426–438.

[19] Frank Rudzicz, Aravind Kumar Namasivayam, and Talya Wolff. “The TORGO database of acoustic and articulatory speech from speakers with dysarthria”. In: *Language resources and evaluation* 46.4 (2012), pp. 523–541.

[20] Leonardo Pepino, Pablo Riera, and Luciana Ferrer. “Emotion recognition from speech using wav2vec 2.0 embeddings”. In: *arXiv preprint arXiv:2104.03502* (2021).

[21] Giulia Sanguedolce et al. “When whisper listens to aphasia: Advancing robust post-stroke speech recognition”. In: *Proc. of Interspeech*. 2024, pp. 1995–1999.

[22] Emmanuel Akinrintoyo, Nadine Abdelhalim, and Nicole Salomons. “WhisperD: Dementia Speech Recognition and Filler Word Detection with Whisper”. In: *arXiv preprint arXiv:2505.21551* (2025).

[23] Sara Mühlhausen et al. “Cross lingual transfer learning does not improve aphasic speech recognition”. In: *Elektronische Sprachsignalverarbeitung 2025: Tagungsband der 36. Konferenz Halle/Saale, 05.–07. MÄRZ 2025*. TUDpress. 2025.

[24] Gerasimos Chatzoudis et al. “Zero-shot cross-lingual aphasia detection using automatic speech recognition”. In: *arXiv preprint arXiv:2204.00448* (2022).

[25] Laurin Wagner, Mario Zusag, and Theresa Bloder. “Careful Whisper–leveraging advances in automatic speech recognition for robust and interpretable aphasia subtype classification”. In: *arXiv preprint arXiv:2308.01327* (2023).

[26] Yicong Jiang et al. “Perceiver-prompt: Flexible speaker adaptation in Whisper for chinese disordered speech recognition”. In: *arXiv preprint arXiv:2406.09873* (2024).

[27] Jiyang Tang et al. “A new benchmark of aphasia speech recognition and detection based on e-branchformer and multi-task learning”. In: *arXiv preprint arXiv:2305.13331* (2023).

[28] Yunpeng Liu, Xukui Yang, and Dan Qu. “Exploration of Whisper fine-tuning strategies for low-resource ASR”. In: *EURASIP Journal on Audio, Speech, and Music Processing* 2024.1 (2024), p. 29.

[29] Anmol Gulati et al. “Conformer: Convolution-augmented transformer for speech recognition”. In: *arXiv preprint arXiv:2005.08100* (2020).

[30] Sehoon Kim et al. “Squeezeformer: An efficient transformer for automatic speech recognition”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 9361–9373.

[31] Keisuke Kamahori et al. “LiteASR: Efficient Automatic Speech Recognition with Low-Rank Approximation”. In: *arXiv preprint arXiv:2502.20583* (2025).

[32] Hang Shao et al. “DQ-Whisper: Joint Distillation and Quantization for Efficient Multilingual Speech Recognition”. In: *2024 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2024, pp. 240–246.

[33] Rui Wang et al. “Lighthubert: Lightweight and configurable speech representation learning with once-for-all hidden-unit bert”. In: *arXiv preprint arXiv:2203.15610* (2022).

[34] Haoning Xu et al. “Effective and Efficient Mixed Precision Quantization of Speech Foundation Models”. In: *arXiv preprint arXiv:2501.03643* (2025).

[35] S Ding, T Chen, and Z Wang. “Audio lottery: Speech recognition made ultra-lightweight, transferable, and noise-robust”. In: *International Conference on Learning Representations (ICLR)*. 2022.

[36] Jian You and Xiangfeng Li. “A light-weight and efficient punctuation and word casing prediction model for on-device streaming ASR”. In: *arXiv preprint arXiv:2407.13142* (2024).

[37] Harsh Ahlawat, Naveen Aggarwal, and Deepti Gupta. “Automatic Speech Recognition: A survey of deep learning techniques and approaches”. In: *International Journal of Cognitive Computing in Engineering* (2025).

[38] Yuya Fujita et al. “Attention-based asr with lightweight and dynamic convolutions”. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 7034–7038.

[39] Kitty Binek and Kathryn Shelley. *ACWT Corpus*. Aphasia Center of West Texas. doi:10.21415/X4RC-R061.

[40] Gretchen Szabo. *Adler Corpus*. Adler Aphasia Center. doi:10.21415/PM0P-5E52.

[41] Z. Ezzes et al. “An open dataset of connected speech in aphasia with consensus ratings of auditory-perceptual features”. In: *Data* 7.11 (2019), p. 148. DOI: 10.21415/KT40-EA41.

[42] Elizabeth Hoover. *Boston University Corpus*. Aphasia Resource Center, Boston University. doi:10.21415/TSJZ-S629.

[43] Susan Jackson. *University of Kansas Corpus*. University of Kansas Medical Center. doi:10.21415/N5XE-Q619.

A

Appendix: Submitted Paper

AS-ASR: A Lightweight Framework for Aphasia-Specific Automatic Speech Recognition

Chen Bao, Chuanbing Huo, Qinyu Chen, *Member, IEEE*, Chang Gao, *Member, IEEE*

***Abstract*—This paper proposes AS-ASR, a lightweight aphasia-specific speech recognition framework based on Whisper-tiny, tailored for low-resource deployment on edge devices. Our approach introduces a hybrid training strategy that systematically combines standard and aphasic speech at varying ratios, enabling robust generalization, and a GPT–4–based reference enhancement method that refines noisy aphasic transcripts, improving supervision quality. We conduct extensive experiments across multiple data mixing configurations and evaluation settings. Results show that our fine-tuned model significantly outperforms the zero-shot baseline, reducing WER on aphasic speech by over 30% while preserving performance on standard speech. The proposed framework offers a scalable, efficient solution for real-world disordered speech recognition.**

***Index Terms*—Aphasic Speech Recognition, AIoT, Healthcare, Whisper, Fine-Tuning**

I. Introduction

Aphasia is a language disorder caused by brain damage due to stroke, traumatic injury, or neurodegenerative disease. Individuals with aphasia experience impairments in both language comprehension and production [1]. Their speech is often marked by distorted articulation, hesitations, repetitions, and disrupted fluency. These features deviate significantly from typical speech patterns [2]. Such irregularities pose serious challenges for speech-based technologies and highlight the need for automatic speech recognition (ASR) systems that can assist clinicians in assessing, monitoring, and supporting individuals with disordered speech.

Despite recent progress in ASR, mainstream models remain predominantly trained on fluent, well-structured data and struggle to generalize to disordered speech. One such model is Whisper, a versatile encoder decoder ASR framework introduced by OpenAI, trained on 680,000 hours of multilingual and multitask audio data collected from the web [3]. While Whisper demonstrates robust performance across a wide range of domains including noisy and accented speech, it, like other standard ASR systems, shows limitations when applied to disordered speech such as aphasia [4]. These systems often produce incomplete, incorrect, or misleading transcriptions due to their lack of exposure to disfluent or atypical linguistic patterns during training [5].

Chen Bao and Chang Gao are with the Department of Microelectronics, Delft University of Technology, The Netherlands.

Qinyu Chen is with the Leiden Institute of Advanced Computer Science (LIACS), Leiden University, The Netherlands.

Chuanbing Huo is with Sanford Health, Minnesota, United States.

Corresponding authors: Chang Gao (chang.gao@tudelft.nl) and Qinyu Chen (q.chen@liacs.leidenuniv.nl).

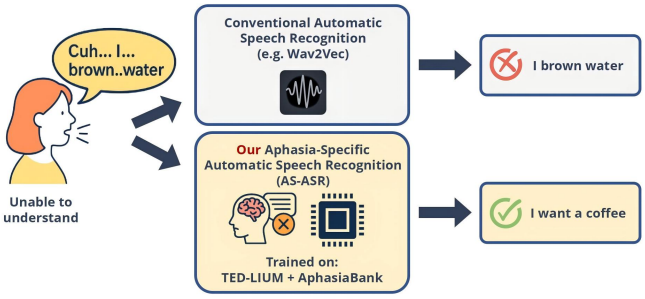

Fig. 1: Comparison between conventional and aphasia-specific ASR systems on aphasia speech.

In recent years, edge devices have been increasingly integrated into rehabilitation technologies and wearable systems [6]. This trend underscores a growing demand for real-time, offline speech recognition capabilities that can operate independently of cloud infrastructure. Edge devices offer several advantages, including low power consumption, portability, cost efficiency, and improved data privacy [7]. However, these benefits come at the cost of limited computational resources and memory capacity. As a result, deploying ASR models on such platforms requires careful design choices. Model architecture, size, and inference speed must be optimized to ensure practical and responsive performance [8].

To address these challenges, we propose **AS-ASR**, a lightweight and aphasia-specific ASR framework based on Whisper, optimized for edge deployment and disordered speech recognition. As illustrated in Fig. 1, AS-ASR can correctly interpret a fragmented aphasic utterance (e.g., "Cuh... I... brown..water") as "I want a coffee," whereas a conventional ASR system such as Wav2Vec generates an incomplete or misleading output. This example highlights the importance of domain adaptation and intent recovery when dealing with aphasia speech. The contributions include:

1) We propose AS-ASR, a lightweight and aphasia-specific ASR framework based on Whisper, tailored for deployment on edge devices.
2) We construct a hybrid dataset combining pathological and normal speech with varying mixing ratios, enabling systematic analysis of data composition effects on recognition performance.
3) We introduce a GPT-4–based transcript enhancement method that refines noisy aphasic references, improving training quality and model generalization.

II. Related Work

Early efforts to apply automatic speech recognition (ASR) to disordered speech focused on narrative tasks to distinguish types of aphasia. Fraser et al. [9] explored this for PPA using ASR-derived lexical features, demonstrating the potential of automated diagnosis even under high word error rates (WERs). Le and Mower Provost [10] later established a large-vocabulary continuous speech recognition baseline on AphasiaBank and investigated domain adaptation strategies, showing that even modest datasets could improve ASR performance with DNNs.

More recent work has emphasized the need to account for speech variability among aphasia patients. Perez et al. [11] addressed this with a mixture of experts (MoE) model guided by a speech intelligibility detector, yielding significant gains across severity levels. The scarcity of labeled data in non-English languages led Chatzoudis et al. [12] to propose a zero-shot cross-lingual framework, utilizing pre-trained multilingual ASR models and language-agnostic features to detect aphasia in Greek and French.

Sanguedolce et al. [13] first carried out a clinical evaluation of Whisper, revealing that its WER strongly tracks aphasia severity and underscoring the limits of models trained solely on fluent speech for post-stroke patients. Building on this, they later released the SONIVA corpus and fully fine-tuned Whisper, attaining state-of-the-art accuracy while illuminating how model size shapes generalization [14]. These open resources neatly complement Tang et al.'s multi-task CTC/Attention + E-Branchformer benchmark [15], together laying a reproducible foundation for aphasia-focused ASR.

Separately, Liu et al. [16] showed that parameter-efficient fine-tuning can yield similar gains in low-resource scenarios, paving the way for the practical deployment of Whisper on clinical and edge devices serving people with disordered speech.

III. Methodology

This section introduces our aphasia-specific fine-tuning framework, with a focus on baseline model selection, dataset construction, speech reference enhancement, and training data composition strategies.

A. Efficient Model Selection

This project employs Whisper-tiny-en, the smallest model in the Whisper series with approximately 39 million parameters. As shown in Table I, this English-only variant is optimized for English tasks, requires only 1GB of VRAM, and offers the fastest inference speed in the series which are about 10 times faster than the large model. These properties make it well-suited for real-time deployment on resource-constrained edge devices such as the Jetson Nano. Despite its compact size, the model achieves notable gains through targeted fine tuning on the acoustic and linguistic features of aphasic speech, making it a strong candidate for low-latency pathological speech recognition.

TABLE I: Comparison of Whisper model sizes in terms of parameter count, memory requirements, and relative inference speed.

Model Size	Parameters	Required VRAM	Relative Speed
Tiny	39M	∼1 GB	∼10×
Base	74M	∼1 GB	∼7×
Small	244M	∼2 GB	∼4×
Medium	769M	∼5 GB	∼2×
Large	1550M	∼10 GB	1×

B. Hybrid Data Fine-tuning Strategy

To enhance the model's adaptability to aphasia speech, we fine-tuned the pre-trained Whisper-tiny-en model using a hybrid training set composed of both aphasic and normal speech data. This mixed dataset combines speech samples from individuals with aphasia (AphasiaBank) and normal speakers (TED-LIUM v2), enabling the model to learn both pathological acoustic patterns and standard phonetic structures. By exposing the model to diverse speech characteristics during training, we aim to improve its robustness and generalization in real-world scenarios where speech can vary significantly. The ratio of aphasic-to-normal speech in the training data was systematically varied in subsequent experiments to study its effect on recognition performance.

C. Enhanced Aphasia-Centric Dataset Construction

For aphasia speech corpus, we utilize speech samples from five sub-corpora of the AphasiaBank database, namely the ACWT Corpus [17], Adler Corpus [18], APROCSA Corpus [19], Boston University Corpus [20], and University of Kansas Corpus [21]. These corpora provide structured interviews and narrative speech data from individuals with post-stroke aphasia, collected across various clinical centers in the United States. The corpus size is strategically matched to TEDLIUM v2 (around 14,000 segments) to enable controlled cross-corpus comparisons.

Importantly, we apply GPT-4–based reference enhancement to improve the quality and clarity of pathological speech labels and provide contextually plausible interpretations, preserving speaker intent while handling aphasia speech. This step provides higher-quality supervision for model learning. Specifically, from the raw .cha transcripts provided by AphasiaBank, we first extract dialog segments between the patient and the clinician, along with their corresponding timestamps. We then remove special symbols in the patient's utterances that denote nonverbal actions or vocalizations (e.g., gestures, sound effects), to isolate the linguistic content. The cleaned patient utterances are subsequently fed into GPT-4 with a prompt instructing it to produce concise, faithful rewrites in standard English without adding extraneous details. These enhanced references are then used to fine-tune the model for improved robustness and intelligibility in recognizing aphasic speech.

IV. Experimental Setup

A. Data Partitioning

For each mixing ratio between pathological and normal speech (e.g., 10:90, 50:50), we adopt a consistent partitioning strategy: 11,200 segments for the training set (80%), 1,400 segments for the validation set (10%) and 1,400 segments for test set (10%). Table II details the specific composition under different mixing ratios. For instance, in the 10%:90% ratio configuration: 1,120 Aphasia training samples and 10,080 TED-LIUM training samples.

TABLE II: Data distribution under different Aphasia:TED-LIUM mixing ratios. Each configuration totals 14,000 samples, partitioned into training (80%), development (10%), and test (10%) sets.

Ratio	Aphasia			TED-LIUM		
	Train	Dev	Test	Train	Dev	Test
10%:90%	1120	140	140	10080	1260	1260
30%:70%	3360	420	420	7840	980	980
50%:50%	5600	700	700	5600	700	700
70%:30%	7840	980	980	3360	420	420
90%:10%	10080	1260	1260	1120	140	140

B. Fine-tuning Details

Fine-tuning was performed on Quechua, a high-performance GPU server equipped with eight NVIDIA RTX A6000 GPUs (48 GB GDDR6 each). The Whisper-tiny model was trained for 10 epochs using a per-device batch size of 16. Mixed-precision (fp16) training and gradient checkpointing were enabled to reduce memory footprint and accelerate throughput. The training pipeline was implemented with the HuggingFace `Trainer` framework, using the AdamW optimizer (learning rate = 5×10^{-5}) and a linear learning rate scheduler. Model evaluation was conducted after each epoch, with checkpoints saved accordingly.

C. Evaluation Metric

We adopt the standard Word Error Rate (WER) as the primary evaluation metric:

$$\text{WER} = \frac{S + D + I}{N} \times 100\% \tag{1}$$

where S denotes substitutions, D deletions, I insertions, and N the total words in reference transcripts. All evaluations are conducted on the same test sets to ensure comparability.

D. Baseline Systems

We define two baseline systems for comparative evaluation:

Baseline-1 evaluates the zero-shot performance of the pre-trained Whisper-`Tiny.en` model on three datasets: TED-LIUM only (representing typical speech), Aphasia only (pathological speech), and a merged set combining both. This serves as a reference for the model's generalization without task-specific adaptation.

Baseline-2 reports the average WER of multiple Whisper model variants (Tiny, Base, Small, and Medium) on the pure Aphasia test set. This baseline reflects the inherent challenges of recognizing disordered speech and highlights how model scale affects robustness.

V. Results & Analysis

This section first presents the baseline performance of the untuned Whisper-tiny-en model on the standard TED-LIUM v2 and the aphasic speech corpus to establish reference metrics. Subsequently, we demonstrate the performance of models fine-tuned on TED-LIUM-only and mixed corpora (via separate and merged approaches) to assess the effectiveness of different fine-tuning methodologies. Finally, we systematically analyze the impact of varying aphasia-to-normal speech data ratios on fine-tuning outcomes by comparing model performance across different mixing proportions.

A. Baseline Performance

Table III summarizes the average WER of different Whisper model sizes on the pure Aphasia test set (Baseline-2). Among the five Whisper model variants evaluated on the aphasic speech dataset, the Whisper-Large model achieved the lowest average WER of 0.622, demonstrating the best overall recognition accuracy for disordered speech. The `Medium.en` and `Small.en` models followed closely, with average WERs of 0.677 and 0.686, respectively, suggesting diminishing returns as model size increases. Surprisingly, the `Base.en` model performed the worst with a WER of 0.886, even higher than the much smaller `Tiny.en` model (WER = 0.787). This indicates that larger models do not consistently guarantee better performance in the context of aphasic speech. Given its lightweight architecture (39M parameters), small memory footprint (1GB VRAM), and moderate recognition ability, the Whisper-`Tiny.en` model remains a promising choice for deployment on low-power, resource-constrained edge devices where real-time inference is required.

TABLE III: Baseline-2 WER Results on Aphasia Datasets by different Whisper models.

Model Size	Average WER
`Tiny.en`	0.787
`Base.en`	0.886
`Small.en`	0.686
`Medium.en`	0.677
`Large`	0.622

B. Fine-tuning Results

Following the baseline evaluation across multiple Whisper variants (Table III), we selected Whisper-`Tiny.en` as the foundation for further fine-tuning due to its balance between model size and performance. As shown in Table IV, we fine-tuned Whisper-`Tiny.en` on a combined dataset of TED-LIUM and AphasiaBank, and evaluated it under three testing

conditions: TED-LIUM only, Aphasia only, and a merged set combining both domains.

Compared to the original Whisper-`Tiny.en` baseline (Baseline-1 in Table IV), which operates without any task-specific fine-tuning, the fine-tuned model demonstrates substantial improvements in aphasia and mixed speech. Baseline-1 reflects the model's zero-shot generalization capability, while it performs well on clean TED-LIUM speech (WER = 0.119/0.117 on dev/test), its performance on aphasic speech deteriorates drastically (WER = 0.808/0.755), and it struggles to generalize under mixed-domain conditions (WER = 0.515/0.498 on the merged set). After fine-tuning on hybrid data, the model retains its recognition capability on fluent speech (WER = 0.116 on both TED-LIUM dev and test), indicating no significant degradation. More importantly, it achieves dramatic gains on aphasic speech, reducing WER to 0.430 (dev) and 0.454 (test). It also shows enhanced robustness on the merged set (WER = 0.162/0.170), underscoring improved domain adaptability.

These results highlight that fine-tuning not only mitigates the baseline model's limitations on pathological speech but also strengthens its generalization across domains, without sacrificing performance on clean speech. This makes the fine-tuned Whisper-`Tiny.en` model a strong candidate for deployment in real-time clinical ASR systems operating under diverse acoustic and linguistic conditions.

TABLE IV: WER of Baseline-1 and Fine-tuned Variants on TED-LIUM and Aphasic Speech

Model	Dataset	Dev WER	Test WER
Whisper-`Tiny.en` (Baseline-1)	TED-LIUM only	0.119	0.117
	Aphasia only	0.808	0.755
	Merged	0.515	0.498
Fine-tuned Whisper-`Tiny.en`	TED-LIUM only	0.116	0.116
	Aphasia only	0.430	0.454
	Merged	0.162	0.170

C. Impact of Data Ratios

We further investigated how varying the mixing ratios of aphasic and normal speech data (Aphasia:TED-LIUM = 10:90, 30:70, 50:50, 70:30, and 90:10) affects the performance of the fine-tuned Whisper-`Tiny.en` model. As shown in Fig. 2, we evaluated the WER on both aphasic and TED-LIUM dev/test sets across different training compositions.

As the proportion of aphasic data increases, a clear trade-off emerges between the two domains:

- **Aphasia (black and red lines):** WER steadily decreases on both dev and test sets as more aphasic speech is introduced during training. This indicates improved adaptation to disordered speech and greater robustness in recognizing pathological utterances.
- **TED-LIUM (blue and green lines):** WER gradually increases, suggesting reduced generalization to clean, fluent speech when the model is increasingly specialized for aphasic input.

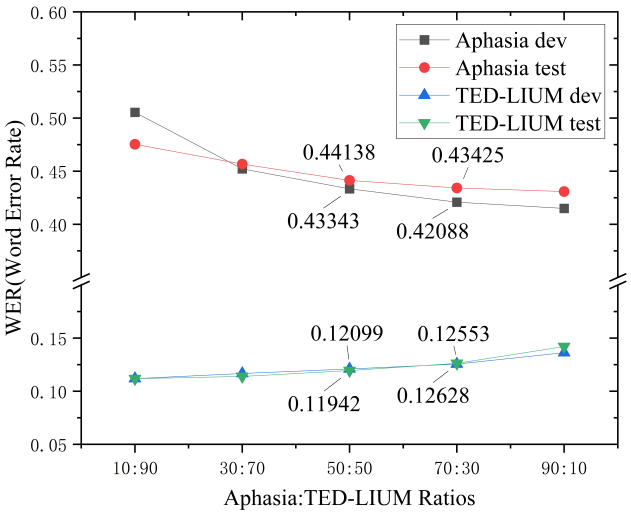

Fig. 2: Impact of Aphasia-to-TED-LIUM ratios on WER performance.

These findings underscore the importance of data composition in multi-domain ASR. Notably, a balanced configuration (e.g., 50:50 or 70:30) offers a practical compromise, maintaining reasonable recognition performance on both disordered and fluent speech. Such balance is especially relevant for clinical ASR systems intended to operate across diverse real-world conditions.

VI. Conclusion & Future Work

This work introduced AS-ASR, a lightweight, aphasia-specific speech recognition framework that achieves robust performance at a low computational cost suitable for edge deployment. By leveraging a hybrid dataset of pathological and typical speech, along with GPT-4-enhanced transcripts, our fine-tuned Whisper-based model significantly improves recognition quality for aphasic speech. The next logical step is to design a dedicated silicon hardware accelerator, drawing inspiration from prior work [22], [23], to achieve real-time inference. This would unlock the full potential of AS-ASR in a new generation of clinical and wearable AIoT healthcare products.

Acknowledgment

This work was partially supported by the Dutch Research Council (NWO) under the Talent Programme Veni 2023 scheme in Applied and Engineering Sciences (AES), Grant No. 21132 (Energy-Efficient Real-Time Edge Intelligence for Wearable Healthcare Devices), and LIACS Strategic Postdocs and PhD Researchers Program 2024 (Next-Generation LLM System for Post-Stroke Speech Assistance and Rehabilitation). We also thank AphasiaBank for granting access to the aphasic speech dataset, and acknowledge its support by NIH-NIDCD Grant R01-DC008524 (2022–2027).

REFERENCES

[1] A. R. Damasio, "Aphasia," *New England Journal of Medicine*, vol. 326, no. 8, pp. 531–539, 1992.

[2] I. G. Torre, M. Romero, and A. Álvarez, "Improving aphasic speech recognition by using novel semi-supervised learning methods on aphasiabank for english and spanish," *Applied Sciences*, vol. 11, no. 19, p. 8872, 2021.

[3] A. Radford, J. W. Kim, T. Xu, T. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2023, pp. 28 492–28 518.

[4] D. Le, K. Licata, and E. M. Provost, "Automatic paraphasia detection from aphasic speech: A preliminary study," in *Proc. Interspeech*, Aug. 2017, pp. 294–298.

[5] G. Sanguedolce, D. C. Gruia, S. Brook, P. Naylor, and F. Geranmayeh, "Universal speech disorder recognition: Towards a foundation model for cross-pathology generalisation," in *Advances in Medical Foundation Models: Explainability, Robustness, Security, and Beyond*, 2024.

[6] A. T. Nguyen, M. W. Drealan, D. K. Luu, M. Z. Jiang, J. Xu, J. Cheng, and Z. Yang, "A portable, self-contained neuroprosthetic hand with deep learning-based finger control," *J. Neural Eng.*, vol. 18, no. 5, p. 056051, 2021.

[7] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.

[8] S. Gondi and V. Pratap, "Performance and efficiency evaluation of asr inference on the edge," *Sustainability*, vol. 13, no. 22, p. 12392, 2021.

[9] K. C. Fraser, F. Rudzicz, N. Graham, and E. Rochon, "Automatic speech recognition in the diagnosis of primary progressive aphasia," in *Proc. 4th Workshop on Speech and Language Processing for Assistive Technologies*, Grenoble, France, Aug. 2013, pp. 47–54.

[10] D. Le and E. M. Provost, "Improving automatic recognition of aphasic speech with aphasiabank," in *Proc. Interspeech*, San Francisco, CA, USA, Sep. 2016, pp. 2681–2685.

[11] M. Perez, Z. Aldeneh, and E. M. Provost, "Aphasic speech recognition using a mixture of speech intelligibility experts," arXiv preprint arXiv:2008.10788, 2020.

[12] G. Chatzoudis, M. Plitsis, S. Stamouli, A.-L. Dimou, A. Katsamanis, and V. Katsouros, "Zero-shot cross-lingual aphasia detection using automatic speech recognition," arXiv preprint arXiv:2204.00448, 2022.

[13] G. Sanguedolce, P. A. Naylor, and F. Geranmayeh, "Uncovering the potential for a weakly supervised end-to-end model in recognising speech from patient with post-stroke aphasia," in *Proc. 5th Clinical Natural Language Processing Workshop*, Toronto, Canada, Jul. 2023, pp. 182–190.

[14] G. Sanguedolce, S. Brook, D. C. Gruia, P. A. Naylor, and F. Geranmayeh, "When whisper listens to aphasia: Advancing robust post-stroke speech recognition," in *Proc. of Interspeech*, 2024, pp. 1995–1999.

[15] J. Tang, W. Chen, X. Chang, S. Watanabe, and B. MacWhinney, "A new benchmark of aphasia speech recognition and detection based on e-branchformer and multi-task learning," arXiv preprint arXiv:2305.13331, 2023.

[16] Y. Liu, X. Yang, and D. Qu, "Exploration of whisper fine-tuning strategies for low-resource asr," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2024, no. 1, p. 29, 2024.

[17] K. Binek and K. Shelley, "Acwt corpus," Aphasia Center of West Texas, doi:10.21415/X4RC-R061.

[18] G. Szabo, "Adler corpus," Adler Aphasia Center, doi:10.21415/PM0P-5E52.

[19] Z. Ezzes, S. M. Schneck, M. Casilio, A. Mefford, M. de Riesthal, and S. M. Wilson, "An open dataset of connected speech in aphasia with consensus ratings of auditory-perceptual features," *Data*, vol. 7, no. 11, p. 148, 2019.

[20] E. Hoover, "Boston university corpus," Aphasia Resource Center, Boston University, doi:10.21415/TSJZ-S629.

[21] S. Jackson, "University of kansas corpus," University of Kansas Medical Center, doi:10.21415/N5XE-Q619.

[22] C. Gao, S. Braun, I. Kiselev, J. Anumula, T. Delbruck, and S.-C. Liu, "Real-time speech recognition for iot purpose using a delta recurrent neural network accelerator," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1–5.

[23] C. Gao, T. Delbruck, and S.-C. Liu, "Spartus: A 9.4 top/s fpga-based lstm accelerator exploiting spatio-temporal sparsity," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 1, pp. 1098–1112, 2024.