**Journal of Information Systems Engineering and Management**
2025, 10(15s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

**Research Article**

# Designing and Optimizing Deep Learning Models for Speech Recognition in the Albanian Language

Ardiana Topi[1*], Adelina Albrahimi[2] & Reinald Zykaj[1]
[1]*European University of Tirana, Faculty of Engineering, Informatics and Architecture, Department of Informatics and Technology, Street Xhanfize Keko, Kompleksi Xhura, Tirana, 1000, Albania*
[2] *University of Tirana, Faculty of Foreign Languages, Department of English Letters and History, Tirana, Albania*
**Corresponding author: ardiana.topi@uet.edu.al*

**ARTICLE INFO**

Received: 02 Dec 2024
Revised: 25 Jan 2025
Accepted: 02 Feb 2025

**ABSTRACT**

Recent studies have highlighted the design and optimization of deep learning models tailored for language speech recognition. Utilizing this tool for underrepresented languages, such as Albanian, which features intricate phonetic and syntactic structures and is regarded as a limited resource language, marks a significant step forward in speech recognition technologies within communities that speak it. While highly effective for widely spoken languages like English and Mandarin, languages like Albanian face technological challenges due to insufficient linguistic resources and a lack of research focus. It is crucial to develop speech recognition systems that accurately capture the distinctive linguistic traits of Albanian, taking into account the challenges posed by its various dialects and complex grammar, for the design and optimization of deep learning models. Computational linguistics, which explores deep learning applications in natural language processing (NLP), involves qualitative and quantitative analyses, developing speech datasets, training and testing different deep learning architectures, and employing optimization techniques. These models are assessed using standard speech recognition metrics, such as word error rate (WER) and computational efficiency. In summary, our findings provide a robust, efficient, and scalable framework for Albanian speech recognition, contributing to the broader objective of enhancing language inclusion in AI technologies.

**Keywords:** Deep Learning, Natural Language Processing, Speech Recognition Technology, Convolutional neural networks, Whisper model, Seq2Seq, Albanian

## INTRODUCTION

### 1.1. Speech Recognition Technology

Speech Recognition Technology (SRT) has ushered in a new era of human-computer interaction, fundamentally transforming how we engage with our devices and broadening the range of what automated systems can achieve. Initially seen as a "hot" technology with limited applications, speech recognition rapidly evolved into a crucial element across various sectors, including telecommunications, automotive, and consumer electronics. Its capacity to convert spoken language into text or executable commands enables seamless, human-free operations, greatly enhancing user convenience and operational efficiency (Kamath, Liu, & Whitaker, 2019).

SRT services go beyond simple command-based interactions with personal devices. In terms of accessibility, it serves as a lifeline for individuals with disabilities, offering a means to engage with technology without the physical barriers posed by traditional data entry methods (Hepsiba & Justin, 2019). In the global context of an aging population, SRT can help this demographic maintain their independence longer, support daily activities, and enhance their quality of life through voice-driven technology (Greig et al., 2019). In healthcare, speech recognition technology is transforming workflows and patient care processes. It allows healthcare providers to dictate notes and update patient records in real time, reducing administrative burdens. This technology also supports medical diagnostics and remote patient monitoring by integrating voice commands into medical devices (Mehrish et al., 2023). Furthermore, SRT is essential in breaking down language barriers and facilitating real-time translation and multilingual communication, which can revolutionize personal and business interactions in our increasingly globalized society. This ability is invaluable for

regions with multiple official languages or significant immigrant populations, ensuring that all community members can access services and information in their preferred language. Despite these advancements, the development and refinement of SRT are not without challenges. Most speech recognition models have been built with resource-rich languages such as English, backed by extensive data and research support. This emphasis has inadvertently led to a gap where resource-poor languages like Albanian suffer from a lack of robust, accurate speech recognition systems. With its unique phonetic and syntactic complexities, Albanian exemplifies the linguistic diversity that current models often struggle to accommodate effectively. Developing SRT capable of handling such diversity is crucial for the inclusion and practical applicability of these systems in varied linguistic landscapes (Schultz & Kirchhoff, 2006). This paper aims to bridge this gap by optimizing deep learning architectures designed for Albanian speech recognition. By exploring different architectures, such as convolutional neural networks (CNN) and recurrent neural networks (RNN), this work seeks to create a highly accurate and efficient model for recognizing the Albanian language. Additionally, this research will address the specific challenges and opportunities presented by the Albanian language, aiming to develop optimized solutions that can be scaled and adapted to meet the needs of other similar languages worldwide.

### 1.2. The Albanian language and SRT application challenges

The development of speech recognition technology (SRT) for the Albanian language presents unique challenges and opportunities. Unlike more widely studied languages that have benefited from extensive research and development, Albanian is a resource-constrained language lacking comparable advancements in speech recognition technology. This discrepancy highlights a broader issue within the group of languages applicable to SRT, where economic and cultural factors often dictate the allocation of resources for language technology development (Belinkov, Ali, & Glass, 2019).

The phonetic and syntactic complexity of the Albanian language is underscored by its rich phonetic and morphological structure, presenting significant challenges for speech recognition systems. Notable phonetic differences, such as nasal vowels and a series of palatalized consonants, are less common in dominant global languages. Additionally, Albanian employs a flexible syntax and extensive inflationary morphology, which can alter the pronunciation and length of words based on grammatical context, complicating language modeling for speech recognition systems. The syntactic flexibility of Albanian, wherein word order can change significantly without altering the meaning of sentences, adds another layer of complexity. This variability requires robust language models capable of interpreting and predicting a wide range of syntactic constructions without compromising the understanding of spoken content (Nirenburg, 2009).

One of the primary obstacles to advancing SRT for this language is the lack of linguistic resources and the compilation of comprehensive language datasets essential for training effective speech recognition models. This gap significantly impedes the implementation of data-intensive machine learning techniques that are fundamental to modern speech recognition systems (Hepsiba & Justin, 2019). This situation reflects a broader issue in the development of technology for less economically dominant languages, which often receive inadequate attention from the global technology community.

### 1.3. Evolution and Impact of Deep Learning in Speech Recognition

Automatic speech recognition (ASR) has transformed significantly with deep learning technologies. Traditionally, ASR relied on hand-crafted features and linear classifiers, but deep learning has revolutionized this. It enables automatic learning of feature representations, enhancing performance and accuracy (Kamath, Liu, & Whitaker, 2019). Deep learning's integration into ASR began with studies on GMM-HMM systems, focusing on improving accuracy through joint learning of classifiers and feature transforms, marking a shift from traditional methods where features like Mel Frequency Cepstral Coefficients (MFCCs) went through multiple stages. Deep Neural Networks (DNNs) can learn transformations directly from raw audio, removing complex preprocessing. The evolution from simple models to intricate neural architectures is notable, particularly in models combining CNNs and RNNs, such as LSTM networks. This combination utilizes CNNs' spatial feature extraction and RNNs' temporal sequence modeling, significantly advancing ASR. Research shows LSTM models using CNN outputs can capture subtle speech variations, reducing error rates (Hepsiba & Justin, 2019).

## 1.4. Key technologies in speech recognition

Deep learning techniques have transformed speech signal processing. Traditional methods like Hidden Markov Models (HMMs) provided a strong statistical foundation but require extensive feature engineering (Mehrish et al., 2023). Deep learning enables models to learn features directly from raw audio, improving performance and scalability (Mehrish et al., 2023). The field has shifted from linear classifiers and manual feature engineering to advanced architectures that automate feature extraction and learning (Mehrish et al., 2023). Convolutional neural networks (CNNs) effectively process time-invariant speech features through convolution layers mimicking human hearing. Recurrent neural networks (RNNs), including long short-term memory networks (LSTMs) and gated recurrent units (GRUs), model the temporal dynamics of speech (Mehrish et al., 2023). Temporal convolutional neural networks (TCNNs) enhance RNNs by allowing parallel processing of input sequences without losing learning depth, making them effective for long sequences (Mehrish et al., 2023). The Conformer model merges CNN feature extraction with Transformers' context integration, addressing local and global dependencies while improving interpretability and training efficiency (Mehrish et al., 2023). Additionally, model efficiency and optimization are crucial for deploying advanced speech recognition systems in resource-limited settings. Menghani (2023) highlights the need for efficient models that ensure high accuracy while minimizing computational costs, vital for the broad adoption of speech recognition technologies (Menghani, 2023).

### 1.5. Advances in Deep Learning models for speech recognition

#### 1.5.1 Static *versus* Dynamic Neural Networks

Speech recognition has evolved from static neural networks to dynamic architectures. Early methods primarily utilized static models like Deep Neural Networks (DNNs) and Convolutional Neural Networks (CNNs), which were effective in handling speech tasks but limited in modeling temporal dynamics. Dynamic neural networks, especially Recurrent Neural Networks (RNNs), advanced the field by processing sequences and maintaining memory of past inputs. This capability helps RNNs understand context better than static networks. However, RNNs face challenges like the vanishing gradient problem, prompting the creation of improved variants such as long short-term memory units (LSTMs) and gated recurrent units (GRUs), designed to address these limitations. LSTMs feature gates that manage information flow, enhancing their effectiveness in capturing long-term dependencies. GRUs offer a simplified mechanism while achieving comparable performance, providing an efficient option for certain tasks. Improvements in LSTM and GRU architectures have fostered deeper learning and better sequential data handling, marking significant progress in neural network capabilities for speech recognition.

#### 1.5.2 Improvements through LSTM and GRU for Optimal Performance

LSTM and GRU networks have resolved key issues in temporal sequence modeling and enhanced speech recognition standards. These models are essential for modern systems, improving accuracy and real-time processing.

Research indicates that dynamic neural networks, particularly LSTM and GRU, outperform traditional models in benchmarks, especially in phoneme recognition, speech synthesis, and voice activity detection (Sak et al., 2014). Integrating LSTMs has enhanced robustness, effectively managing speech variability and background noise in real-world applications.

End-to-end trainable systems using dynamic architectures have transformed the field. These systems combine dynamic networks with CNNs and Transformers, allowing training from audio input to text output, thus streamlining speech recognition and reducing feature engineering reliance. For instance, the Conformer model amalgamates transformers, CNNs, and GRUs or LSTMs to adeptly model speech data (Gulati et al., 2020). Continuous advancements in LSTM and GRU technology are expanding possibilities in speech recognition, paving the way for more adaptive, robust, and efficient systems that bridge human-machine communication.

## 1.6. Optimization strategies for robust speech recognition

#### 1.6.1. Error reduction and model efficiency techniques

Developing robust speech recognition systems relies on effective error reduction strategies and improved model efficiency. A major advance is model quantization, which reduces computational needs and enhances neural network training efficiency. It converts floating-point to fixed-point format, easing computational demands on devices with limited power (Siddegowda et al., 2022). Post-training quantization allows conversion of pre-trained models to fixed-

point without full retraining, streamlining optimization while preserving accuracy (Luhman & Luhman, 2022). These strategies are crucial for real-time applications like mobile voice assistants, where latency and energy efficiency are vital. Techniques like Quantization Attenuation Training (QAT) optimize models during training, ensuring the quantization does not impair performance and helps maintain near-original accuracy in complex networks (AIMET, 2022).

#### 1.6.2. Applying advanced training methods across models

Advanced training methodologies enhance the accuracy of speech recognition models. The Convolutional Temporal Classification (CTC) loss function allows networks to learn sequence predictions without segmented data (Graves et al., 2013). This is particularly useful for continuous speech recognition and improves networks like LSTMs. Additionally, combining Convolutional Neural Networks (CNN) with LSTMs in a CLDNN framework leverages each model's strengths, capturing data's temporal dynamics and spatial hierarchies to enhance accuracy (Sainath et al., 2015). Transfer learning adapts pre-trained models to new tasks with minimal training, accelerating the process and improving model generalization across different languages (Panayotov et al., 2015). These strategies, including efficient hardware techniques, are crucial for advancing speech recognition performance and efficiency in real-world scenarios.

#### 1.6.3. Emerging Trends: Generative Models and Multi-Modal Systems

Advancements in generative models are transforming speech recognition. These models generate synthetic speech that mimics human interaction, enhancing communication through personalized voice assistants and aiding those with speech impairments. Advanced models like Conformer adapt speech to emotional and contextual cues (Gulati et al., 2020).

The rise of multi-modal systems is another significant trend in research. These integrate visual, auditory, and textual data for context-aware responses, improving accuracy over audio-only systems. Conformer's ability to process varying data streams enhances its effectiveness in multi-modal applications (Gulati et al., 2020). Looking ahead, AI integration in speech recognition will tackle ongoing challenges and inspire innovative applications, aiming for accessible and user-friendly systems.

Current literature highlights the rapid evolution and challenges of speech recognition. Initially reliant on manual engineering and basic models, the field has shifted to sophisticated, deep learning-based approaches emphasizing architecture and efficiency.

Key advances include transitioning from GMM-HMM systems to complex architectures like DNNs, CNNs, LSTMs, and GRUs. These technologies improve recognition of nuanced human speech variability (Hepsiba & Justin, 2019; Gulati et al., 2020). Conformer's integration of CNNs for feature extraction and Transformers for context improves accuracy and generalization, addressing data imbalance (Gulati et al., 2020). Deployment has broadened from niche industries to widespread use in electronics, healthcare, and customer service, highlighting the potential to redefine human-computer interactions.

### 1.7. Introduction to the Albanian language

The Albanian language is a unique branch of the Indo-European language family, with a rich and complex history that dates back to ancient times. Spoken by over 10 million people, geographically located in Western Balkans, it serves as the official language of Albania and Kosovo, as well as holds co-official status in parts of North Macedonia and Montenegro, also widely spoken in diaspora communities in Greece, Italy, Switzerland, the United States. Despite these, Albanian remains one of the most under-researched and under-resourced languages in speech recognition (Dedvukaj & Gehringer, 2023).

The language has evolved into two main dialects: Tosk, spoken in southern Albania, North Macedonia, and Greece, and Gheg, spoken in northern regions of Albania, Kosovo, and parts of Montenegro and North Macedonia (Newmark, 1999). The Tosk dialect forms the basis for Standard Albanian, codified in the 20$^{th}$ century and used in official settings, including education, government, and the media. In contrast, the dialects continue to be spoken daily (Rista & Kadriu, 2022). From a grammatical perspective, Albanian is highly inflectional, using a wide range of suffixes and prefixes to convey grammatical relations such as case, gender, and number. Albanian has five noun cases (nominative, genitive, dative, accusative, and dative) and a complex verbal system that includes multiple tenses,

moods, and aspects (Buchholz & Fiedler, 1987). In addition to dialectal differences, linguistic borrowing can complicate the development of models, as training data may include variations inconsistent across Albanian-speaking populations (Rista & Kadriu, 2022).

Despite these challenges, the development of speech recognition systems is very important. Phonetically, Albanian is characterized by several features that distinguish it from other Indo-European languages, including 36 phonemes, and some have no direct equivalents in more widely spoken languages. In particular, it contains nasal vowels, a wide range of palatalized consonants, and a vowel length accent that plays a role in distinguishing word meaning. Such phonetic nuances are essential to recognize and capture accurately in any speech recognition system, as failing to do so would lead to high error rates and inaccuracies (Chaudhary, 2022). In Albanian-speaking regions, SRT may transform how individuals access digital services, educational tools, and government support. For example, voice-activated services in rural or remote areas could bridge the digital divide, providing access to information and resources without requiring technical knowledge or expertise. In addition, SRT could facilitate the language learning and preservation process, especially among diaspora communities, where the younger generation may risk losing fluency (Chaudhary, 2022).

Furthermore, the importance of developing speech recognition systems for the Albanian language goes beyond simple technological advancement. It also addresses issues of linguistic equity. Most research in automatic speech recognition (ASR) has focused on resource-rich languages such as English, Mandarin, and Spanish, leaving many minority languages, such as Albanian, with limited technological means to integrate into the digital world. This lack of support reinforces linguistic inequalities, limiting access to technology for speakers of under-resourced languages and contributing to the erosion of these languages in the face of globalization and digitalization (Schultz & Kirchhoff, 2006).

The development of accurate models for Albanian speech recognition is not simply a technological advance but a step towards linguistic equality, where every language can benefit from the latest artificial intelligence technologies. Beyond Albanian, speech recognition technologies for other underrepresented languages, including languages from remote regions and the diaspora, must be developed and adapted.

### 1.8. Challenges in speech recognition for the Albanian language

Developing speech recognition systems for Albanian faces critical challenges due to a lack of written resources and linguistic data. Albanian is an under-resourced language in computational linguistics, limiting access to annotated datasets necessary for training modern machine learning models. Most advancements have been made in resource-rich languages like English and Mandarin.

Phonetic and morphological features of Albanian, such as nasalized vowels, palatalized consonants, and complex affricates, complicate model adaptation. Additionally, the language's morphological richness—in noun declension and verb conjugation—requires a deep understanding of its grammatical structure.

Dialectal differences between Gheg and Tosk add another layer of difficulty. Many speakers, particularly in Kosovo, often switch dialects, known as code-switching, challenging real-time adaptation in recognition systems. Effective systems must be trained in both dialects or adapt dynamically, necessitating extensive data and model development.

The lack of standardization between spoken and written Albanian further complicates the recognition systems' development.

### 1.9. Computational challenges in real-time speech recognition

Real-time speech recognition systems that process speech quickly while maintaining accuracy are challenging due to Albanian's complex phonetic and morphological features. Models must navigate various phonetic variations, dialect differences, and grammatical structures with a low error rate, demanding high computing power and optimized algorithms to handle real-time data without delays.

Transformer models and attention mechanisms represent major advances in speech recognition deep learning. Unlike RNNs and LSTMs, transformers process input in parallel, using attention to focus on essential input aspects. This greatly improves efficiency and handling long-range dependencies in speech data. For Albanian recognition, transformers are promising due to their adaptability to dialect variations and code-switching, making them ideal for

Gheg and Tosk recognition. Additionally, attention mechanisms help the model focus on critical sentence parts, enhancing its capability to recognize complex grammar.

### 1.10. Transfer Learning and Data Augmentation

One challenge in Albanian speech recognition is the lack of extensive datasets. Researchers use transfer learning, adapting models pre-trained in resource-rich languages to Albanian. This technique minimizes the data needed to train models for resource-constrained languages. Applying pre-trained models from languages like English allows researchers to extract phonetic and structural insights. However, challenges arise since English has simpler morphology and structure compared to Albanian. Retraining models requires substantial computational resources, which are often scarce. Data augmentation techniques are also used to increase Albanian speech datasets artificially. By adding noise or altering pitch and speed of audio files, researchers create new training examples to help the model recognize diverse speech patterns, aiding in handling dialect variation and phonetic complexity.

## 2. METHODOLOGY

### 2.1. Similar Case Study Applications

A prominent example of speech recognition technology for resource-constrained languages is Basque, a minority language in Spain and France. Like Albanian, Basque has minimal data for training advanced models. Researchers improved accuracy using transfer learning, first training models in resource-rich languages like Spanish before adapting them to Basque. Pre-trained Spanish models reduced the need for extensive data sets, proving effective and achieving accuracy on par with Spanish and English. Another case is the Catalan speech recognition project. Despite millions of speakers, Catalan faces challenges against more dominant languages. Researchers applied Deep Neural Networks (DNNs) trained in English and adapted them for Catalan. Data augmentation, adjusting sound frequencies and intensities, improved the models' ability to handle pronunciation variations. Similarly, Albanian speech recognition can utilize transfer learning and data augmentation techniques. Models pre-trained in related languages like Italian or Greek can mitigate the data scarcity for Albanian while augmenting data sets with dialectal variations like Gheg and Tosk can enhance model accuracy for distinguishing phonetic nuances.

### 2.2. Deep Learning Architectures: Seq2Seq and Attention

Deep learning architectures intended for speech recognition include various approaches, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and more advanced ones, such as Transformers with Attention Mechanisms. The Whisper model uses the Transformer-based seq2seq architecture. The model is trained on large audio data sets, and this pre-training is very useful in cases where data in a specific language, such as Albanian, is limited. This approach has significantly succeeded in many natural language processing and speech recognition applications. Seq2seq is a fundamental approach for machine translation and audio transcription tasks, as it translates an input sequence (audio) into an output sequence (text).

*Seq2Seq (Sequence-to-Sequence)*: This model uses a two-way process where the audio signal is processed through an encoder, which extracts features from the audio input, and a decoder, which generates the text string. The advantage of this approach is the ability to convert extended signals into continuous text, which is essential for natural speech. This makes it more potent than older architectures like RNNs, which can have limitations in handling long sequences.

*Attention Mechanisms*: To improve transcription performance, Whisper uses attention mechanisms, which make the model focus on specific parts of the audio that have important information. This mechanism is especially valuable for the Albanian language, where intonation and phonetic nuances are diverse and complex.

Multi-head attention layers: Transformers use many layers of attention (multi-head attention layers), which process information in parallel, improving processing speed and the model's ability to make accurate decisions.

### 2.3. Technique Optimization

Optimization techniques implementation consists of the following steps:

*Transfer Learning*: employment of a model trained on a large dataset (in this case, Whisper) to improve speech recognition in our study. The pre-training process makes recognizing unique Albanian phonemes characterized by limited data easier.

*Model Adaptation*: Several changes are implemented to adapt to language speech characteristics, such as adjusting the translated sequences for length and the phonetic specifications of the Albanian.

*Regulated Attention Mechanisms*: Due to Albanian phonetic challenges, focused attention improves transcription accuracy by concentrating processing power on important audio signal segments. This allows for a more apparent distinction of phonemes that are similar to sounds in other languages but differ subtly.

*Data Padding and Adaptive Batches*: A specific padding mechanism ensures that all data sequences have the same length in the training process. This increases training efficiency and minimizes losses resulting from variations in the length of audio signals.

*Gradient Accumulation*: Gradient accumulation is used to overcome hardware resource limitations and increase the adequate batch size. This process collects gradients for several steps before updating the model parameters. This helps train large models like Whisper even with limited resources.

### 2.4. Evaluation metrics and performance standards of the Model

The model was evaluated using the Word Error Rate (WER) metric, a standard in speech recognition (Ejigu and Tesfa, 2024). This metric measures the differences between the model-generated string and the correct transcription, where errors are quantified as:

Substitutions: Words that have been incorrectly replaced.

*Deletions*: Words that have been deleted from the overall string.

*Insertions*: Words that have been incorrectly added to the final text.

Achieving a low WER is critical for the model's success in languages like Albanian, where recognition accuracy is essential to improving speech recognition technology for resource-poor languages.

## 3. THE DATA COLLECTION AND PRE-PROCESSING

The data for this project was sourced from the Albanian Common Voice dataset. This dataset is a public audio data repository collected from various Albanian speakers. The data preprocessing includes:

Dataset extraction: The audio data was extracted from a compressed archive and arranged into a format suitable for training.

Removing unnecessary columns: To enhance model efficiency during the processing phase, columns irrelevant to transcription (e.g., client ID, votes, accents) were eliminated.

Resampling: All audio data was converted to a uniform sampling rate of 16kHz to ensure the model received data with consistent resolution during training.

In this project, a public Albanian dataset called Common Voice was utilized. This dataset offers audio recordings from various individuals who contributed their voices in several languages, including Albanian. Given that the Albanian language has fewer resources dedicated to speech recognition, using Common Voice is ideal for providing a diverse collection of Albanian voices, representing variations in words, accents, and different speaker registers.

### 3.1. Downloading and Extracting the Dataset

The Common Voice dataset is compressed as a .tar.gz archive, which includes the audio data and accompanying text files for each recording. To begin the processing, the dataset was downloaded and extracted using Python and the tarfile library, a reliable tool for handling compressed formats. This step involved specifying the file paths of the downloaded dataset and dividing the extracted data into separate files for training and testing.

This data extraction is crucial for preparing the audio and associated information for the preprocessing and training phase. The archive contains file structures featuring .tsv files for each audio recording, which hold details such as the transcribed sentence. Additional metadata is available, which can be utilized or concealed for more efficient processing.

### 3.2. Removing Unnecessary Columns

Once the data is extracted, the next crucial step is to select the relevant columns and remove those that do not contribute to the speech recognition model. The dataset includes several columns, such as client_id, sentence_id, up_votes, down_votes, age, gender, and others. While potentially useful for demographic analysis, these columns are not relevant to this project's speech recognition.

Consequently, functions from the Hugging Face datasets library were employed to eliminate unnecessary columns, retaining only those containing the audio file path and the corresponding transcript, which are essential for the model. This data reduction process enhanced processing efficiency and significantly decreased overall processing time, enabling the model to concentrate solely on the essential data for transcription.

### 3.3. Resampling and Sample Rate Uniformization

A resampling process ensured the model received consistent data by transforming the audio signal to a standard sampling rate of 16kHz. This step is critical because a uniform sampling rate guarantees that all audio recordings are processed with the same quality, allowing for easy comparison by the model. Resampling was conducted using torchaudio, a powerful audio processing library that includes functions for converting and transforming sample rates. Through the use of a Resampler, each recording was converted to 16kHz, ensuring compatibility and compliance with the Whisper model's requirements. This resampling process was essential for the next phase of dataset preparation, as it enabled the data to align with the model's characteristics.

### 3.4. Preparing Input and Output Features

After resampling and column selection, the final preprocessing stage involves preparing the input and output features for the Whisper model. WhisperFeatureExtractor extracts audio features, converting audio data into a numeric format usable by the model. WhisperTokenizer encodes text transcripts into a numeric format for the model, tailored for Albanian. Each audio recording becomes an input array, while transcripts are encoded as labels for output, ensuring standardized data for processing.

### 3.5. Implementing Deep Learning Models Using PyTorch

In this project, the speech recognition model was implemented using PyTorch and the Transformers library from Hugging Face, which provides full support for seq2seq models such as Whisper. The approach to building and training the model is detailed below:

#### Data Collator for Sequence Padding

Since Transformer models require uniform sequences during training, a specific Data Collator was used to ensure that all audio and text sequences have the same length for each batch. This collator applies padding to the data, filling the gaps of shorter sequences to match the length of other batches.

The collator is a class that uses WhisperProcessor to maintain consistency between inputs and outputs in each batch. This process helps improve data processing efficiency and prevents the loss of important information for shorter recordings. Padding also ensures that gradients are calculated accurately across all batches, facilitating the training process.

#### Model Training Trainer

The Whisper model is trained using Seq2SeqTrainer, a specific class in the Transformers library for training seq2seq models. The trainer provides a structured training platform for efficient training, including gradient tuning, model storage, and key parameter tuning. The training parameters are configured to maximize model performance under limited computing resource conditions:

*Small Batch Size*: Using a small batch size helps overcome RAM limitations during processing, reducing memory requirements and adapting the dataset.

*Gradient Accumulation Steps*: Using a gradient accumulation step, gradient calculations are performed over several batches before updating the model parameters. This method allows the model to be trained more efficiently, even on devices with lower memory capacity.

*Learning Rate and Warm-Up Steps*: Using a relatively low learning rate and a warm-up phase helps to maintain model stability during the initial training and to ensure that the model focuses on clearer signals during the initial stages.

#### Using Advanced Performance Enhancement Mechanisms

To increase training efficiency and better manage memory capacity, gradient checkpointing is used. This mechanism allows the model to store only the gradients needed during some training stages, reducing memory requirements and speeding up the training process for the Albanian dataset. Gradient checkpointing is used to handle large models like Whisper, where resource requirements are high, and computing resources may be limited.

### 3.6. Modifications and Customizations for the Albanian language

To make the Whisper model suitable for the Albanian language, several essential modifications and customizations have been made:

#### Dedicated Albanian Tokenizer and Processor

A dedicated Albanian WhisperTokenizer has been used to present transcripts and results in Albanian. The tokenizer has been trained to recognize Albanian characters and grammatical structure, enabling the model to understand and produce accurate transcripts in Albanian. This includes adapting the Albanian language parameters in the tokenizer and optimizing the tokenizer to consider Albanian's phonetic intricacies.

#### Adapting Inputs and Labels for Albanian Word Recognition

The dataset preparation process has been adapted for Albanian, including handling specific characters and words only used in Albanian. Suppose the model encounters unique elements of Albanian, such as some specific letters like *ë* and *ç*. In that case, including these characters in the tokenizer's dictionary is necessary.

#### Using Transfer Learning

Instead of building a new model from scratch, the pre-trained Whisper model was used. This project uses transfer learning to take advantage of the knowledge gained on other datasets, improving its performance on a small Albanian dataset. This technique helps to reduce the need for large Albanian datasets and achieve higher accuracy with a limited amount of data.

#### Strengths

*Capacity for Transfer Learning*: Using the Whisper model, pre-trained on various languages and audio data, can leverage prior knowledge to better adapt to Albanian. This transfer learning allows the model to achieve higher accuracy even when using a small dataset.

*Use of Attention Mechanisms*: Attention mechanisms, especially in Whisper's Transformer architecture, improve the model's focus on key parts of the audio signal, making the model more effective in recognizing deep phonetic nuances and minimizing errors in language transcription.

*Modular code structure*: The codes and classes built to process data and train the model, such as DataCollator and compute_metrics, are written in a modular manner, which facilitates their analysis, modification, and adaptation for further research.

*Adapted Evaluation Method (WER):* Using an internationally recognized metric, Word Error Rate (WER), as a standard for transcription evaluation, an objective and standardized way to measure the model's performance has been created, which makes it possible to compare results across different platforms.

#### Limitations

*Limitation on the Albanian Dataset*: The Albanian dataset in Common Voice contains limited data, which can negatively affect the model's performance. This limits the model's ability to learn different speech variations, such as Albanian dialects or regional accents.

*Limited Resource Usage*: The Whisper model is large and requires high memory capacity and processing power requirements. For computers with limited resources, this poses a barrier to efficient training and practical implementation of the model on smaller devices.

*Complexity of Optimization Mechanisms*: Although attention mechanisms improve the model's accuracy, using them increases the complexity of the optimization process and requires time and effort to achieve the desired quality in languages with fewer resources.

*Lack of Capabilities for Contextualizing Specific Language Requirements*: Languages like Albanian have several phonetic and grammatical characteristics that can be challenging for a universal model, causing it to fail to recognize all the language's details and nuances.

### *3.7. Implications of Findings for Speech Recognition Technology in Resource-poor Languages*

Speech recognition in resource-constrained languages like Albanian requires a commitment to developing architectures and techniques that meet the unique needs of these languages. This approach highlights several implications and recommendations for the future of speech recognition technology in resource-constrained languages:

*The Need for Larger and Richer Datasets*: This project emphasizes creating larger and more diverse datasets for languages like Albanian. Additional data could include regional, age, and occupational variations to improve models' performance further.

*Transferring Learning as an Effective Strategy for Specific Languages*: Using pre-trained models incorporating knowledge from similar resource-constrained languages holds great potential for improving performance. This could create a foundation that can support the development of language-specific models.

*Model Optimization and Efficiency Improvements*: This project highlights the importance of improving attention and optimization mechanisms for speech recognition models to make them more efficient and suitable for languages with fewer resources.

*The Importance of Community and Contribution to Providing New Datasets*: Open-source projects like Common Voice emphasize that contributions from individuals can create a rich database for different languages. This demonstrates the importance of community in underserved language technologies, where individual contributions and collaborations between data scientists and native speakers are essential to creating more effective models.

### *1. Preparing the Work Environment*

The first part of the code installs the necessary libraries and authenticates with the Hugging Face platform. Using pip install with the—upgrade flag ensures that the latest libraries (datasets, transformers, accelerate, evaluate, jiwer, tensorboard, and radio) are installed to support audio processing and model training. The notebook_login function from Hugging Face is used to authenticate, which is necessary to access the models and datasets stored in the Hugging Face Hub.

**!pip install --upgrade datasets[audio] transformers accelerate evaluate jiwer tensorboard gradio from huggingface_hub import notebook_login notebook_login()**

### *2. Dataset Extraction and Loading*

Defining the paths for the compressed data file (set-sq.tar.gz) and the destination for the data extraction. Using Python's tarfile library, we ensure that the audio data is extracted to the specified directory (extracted_path). This step is essential as it prepares the raw data for structured processing.

The load_dataset function from datasets loads the train and test partitions from the .tsv files into the extracted directory. Using a delimiter ensures that the tab-separated format is interpreted correctly.

*import tarfile*

*import os*

*import torchaudio*

*from datasets import load_dataset, DatasetDict*

*# Përcaktimi i rrugëve*

*dataset_path = "/workspace/set-sq.tar.gz"  # Rruga për skedarin .tar.gz*

*extracted_path = "/workspace/whisper/extracted"  # Rruga për direktorinë ku do të nxirren të dhënat*

*clips_dir = os.path.join(extracted_path, "cv-corpus-19.0-2024-09-13/sq/clips")*

*# Nxjerrja e skedarit .tar.gz*

*with tarfile.open(dataset_path, 'r:gz') as tar:*

*tar.extractall(path=extracted_path)*

*# Ngarkimi i dataset-it*

*common_voice = {*

*"train": load_dataset("csv", data_files=os.path.join(extracted_path, "cv-corpus-19.0-2024-09-13/sq/train.tsv"), delimiter="\t")["train"],*

*"test": load_dataset("csv", data_files=os.path.join(extracted_path, "cv-corpus-19.0-2024-09-13/sq/test.tsv"), delimiter="\t")["train"]*

*}*

### *3. Clearing data*

To optimize the dataset, specific columns that are not related to the main objective of the model (e.g., demographic data or voting) are removed. This step optimizes memory usage and ensures that only the important features remain. By reducing the dimensions, we give the model the ability to focus only on the audio inputs and the corresponding transcriptions, thus improving training efficiency.

*# Përcaktimi i kolonave që duhet të hiqen nëse ekzistojnë*

*columns_to_remove = ["client_id", "sentence_id", "sentence_domain", "up_votes", "down_votes", "age", "gender", "accents", "variant", "locale", "segment"]*

*# Heqja e kolonave të panevojshme*

*common_voice["train"] = common_voice["train"].remove_columns([col for col in columns_to_remove if col in common_voice["train"].column_names])*

*common_voice["test"] = common_voice["test"].remove_columns([col for col in columns_to_remove if col in common_voice["test"].column_names])*

### *4. Initializing the model*

The Whisper model, designed for speech-to-text tasks, requires components for feature extraction, tokenization, and processing. We use WhisperFeatureExtractor to extract features from audio, WhisperTokenizer to tokenize Albanian text (the language used), and WhisperProcessor to combine these functions.

*from transformers import WhisperFeatureExtractor, WhisperTokenizer, WhisperProcessor*

*# Inicializimi i komponentëve të Whisper*

*feature_extractor= WhisperFeatureExtractor.from_pretrained("openai/whisper-small")*

*tokenizer=WhisperTokenizer.from_pretrained("openai/whisper-small", language="Albanian", task="transcribe")*

*processor=WhisperProcessor.from_pretrained("openai/whisper-small", language="Albanian", task="transcribe")*

### *5. Data preparation function*

This function reads any audio file, resamples it to 16 kHz (the standard for Whisper), and extracts the audio features. Sentences are tokenized to create input features, with care taken about the token length (maximum 448 tokens) to optimize memory usage and avoid congestion. Each batch now contains two important elements: the input features from the audio and the tokenized tags.

```
def prepare_dataset(batch):
    audio_path = os.path.join(clips_dir, batch["path"])
    audio_array, sampling_rate = torchaudio.load(audio_path)
    if sampling_rate != 16000:
        resampler = torchaudio.transforms.Resample(sampling_rate, 16000)
        audio_array = resampler(audio_array)
batch["input_features"]=feature_extractor(audio_array.squeeze().numpy(),
sampling_rate=16000).input_features[0]
batch["labels"]=tokenizer(batch["sentence"],max_length=448,                padding="max_length",
truncation=True).input_ids
    return batch
```

### 6. Dataset Design

This step runs prepare_dataset on each row in the training and testing partitions. Removing unnecessary columns minimizes memory usage and ensures that only relevant features remain.

```
common_voice["train"]=common_voice["train"].map(prepare_dataset,
remove_columns=common_voice["train"].column_names)
common_voice["test"]=common_voice["test"].map(prepare_dataset,
remove_columns=common_voice["test"].column_names)
```

### 7. Setting parameters for training

To evaluate the model performance, we use Word Error Rate (WER), a metric for transcription accuracy. Training parameters are defined to configure learning parameters and checkpoint controls. Using low-level batching and gradient checking helps manage GPU memory. At the same time, regular logging and evaluation allow monitoring of the model's progress.

```
from transformers import Seq2SeqTrainer, Seq2SeqTrainingArguments, WhisperForConditionalGeneration
import evaluate
from dataclasses import dataclass
from typing import Any, Dict, List, Union
import torch
# Ngarkimi i metric-it WER
wer_metric = evaluate.load("wer")
# Funksioni për llogaritjen e metric-it
def compute_metrics(pred):
    pred_ids = pred.predictions
    label_ids = pred.label_ids
    label_ids[label_ids == -100] = tokenizer.pad_token_id
    pred_str = tokenizer.batch_decode(pred_ids, skip_special_tokens=True)
    label_str = tokenizer.batch_decode(label_ids, skip_special_tokens=True)
    wer = wer_metric.compute(predictions=pred_str, references=label_str)
    return {"wer": wer}
```

```
# Vendosja e parametrave të trajnimit
training_args = Seq2SeqTrainingArguments(
    output_dir="./whisper-small-sq",
    per_device_train_batch_size=1,
    gradient_accumulation_steps=8,
    learning_rate=1e-5,
    max_steps=2000,
    logging_steps=5,
    save_steps=500,
    eval_steps=400,
    report_to="none",
    fp16=False,
    gradient_checkpointing=True
)
```

### 8. Data Collator for Padding usage

The data collator adapts and structures the inputs and labels for training. The padding ensures that all inputs in a batch have consistent dimensions, a requirement for efficient model training.

```
@dataclass
class DataCollatorSpeechSeq2SeqWithPadding:
    processor: Any
    decoder_start_token_id: int
    def __call__(self, features: List[Dict[str, Union[List[int], torch.Tensor]]]) -> Dict[str, torch.Tensor]:
        input_features = [{"input_features": feature["input_features"]} for feature in features]
     batch=processor.feature_extractor.pad(input_features, return_tensors="pt")
    label_features = [{"input_ids": feature["labels"]} for feature in features]
      labels_batch=processor.tokenizer.pad(label_features, return_tensors="pt")
        labels = labels_batch["input_ids"].masked_fill(labels_batch.attention_mask.ne(1), -100)
        if (labels[:, 0] == self.decoder_start_token_id).all().cpu().item():
            labels = labels[:, 1:]
        batch["labels"] = labels
        return batch
```

### 9. Model Training and Evaluation

Finally, the model is trained using Seq2SeqTrainer, with the arguments, datasets, and configuration metrics. After training, evaluation of the test set produces the WER metric, providing an insight into the model's accuracy.

```
data_collator=DataCollatorSpeechSeq2SeqWithPadding(processor=processor,
decoder_start_token_id=tokenizer.pad_token_id)

trainer=Seq2SeqTrainer(
```

args=training_args, model=WhisperForConditionalGeneration.from_pretrained("openai/whisper-small"), train_dataset=common_voice["train"], eval_dataset=common_voice["test"], data_collator=data_collator, compute_metrics=compute_metrics, tokenizer=processor.feature_extractor, ) trainer.train() # Vlerësimi i modelit metrics = trainer.evaluate() print("Evaluation metrics:", metrics)

*11. Word Error Rate*

This code evaluates the model's performance using Word Error Rate (WER), a metric that measures how well the model generates accurate transcriptions relative to the reference data..

from torch.utils.data import DataLoader

import numpy as np

import gc import torch

**Importing libraries:**

Here, you import the libraries needed for data loading,

tensor manipulation, and GPU memory management are especially important for complex models like Whisper.

# Pastrimi i memorjes së mbetur të GPU-së

torch.cuda.empty_cache()

**GPU Cleanup:** This command removes any remaining data in the GPU memory, thus avoiding its unnecessary use during model evaluation.

# Konfigurimi i DataLoader-it për vlerësim

eval_dataloader = DataLoader(common_voice["test"], batch_size=8, collate_fn=data_collator)

**Preparing evaluation data:** DataLoader helps in batching the data for evaluation and allows models to use large amounts of data efficiently. batch_size=8 increases processing efficiency, while collate_fn is used to prepare and collate the data.

# Vendosja e modelit në gjendjen e vlerësimit

model.eval()

predictions = []

references = []

**Putting the model into evaluation state:** The model.eval() function ensures that no updates to the model weights occur during evaluation, a key action for the stability of the results.

for step, batch in enumerate(eval_dataloader):

with torch.no_grad():

# Konvertimi i inputeve në tensorë dhe transferimi në GPU

input_features = torch.stack([f.clone().detach().float() for f in batch["input_features"]]).to("cuda")

**Evaluation loop:** torch.no_grad() helps save GPU memory during evaluation by not storing the history of calculations for the backward function. This is an optimal approach to conserve resources, especially for intensive evaluation models.

# Gjenerimi i transkriptimeve nga modeli

generated_ids = model.generate(

input_features,

max_length=128,

return_timestamps=True

)

**Generating transcripts**: The *model.generate* function generates transcripts for the given inputs. By limiting the maximum length (max_length=128), processing time is improved, and unnecessary generation is avoided.

```
# Dekodimi i parashikimeve dhe referencave
pred_str = tokenizer.batch_decode(generated_ids, skip_special_tokens=True)
ref_str = tokenizer.batch_decode(batch["labels"].cpu().numpy(), skip_special_tokens=True)
# Shtimi i parashikimeve dhe referencave për kalkulimin e WER
predictions.extend(pred_str)
references.extend(ref_str)
```

**Decoding and saving transcriptions**: These lines take the model predictions and references from the data to later calculate the WER. *skip_special_tokens=True removes* any unnecessary characters, improving the accuracy of the comparison.

# Llogaritja e WER wer = wer_metric.compute(predictions=predictions, references=references) print(f"Word Error Rate (WER): {wer * 100:.2f}%")

**Calculating and displaying WER:** WER measures the number of errors in the model's transcriptions, with a lower value indicating better model performance. This metric is critical for assessing the model's accuracy in Albanian language translation.

### *12. Visualizing loss during training*

This part of the code is focused on creating graphs to visualize the change in loss during training, which helps monitor the model's performance and stability.

import matplotlib.pyplot as plt

**Visualization library**: *matplotlib.pyplot* creates graphs reflecting the model's progress.

```
# Nxjerrja e të dhënave të humbjes nga log-et e trajnimit
training_steps = []
training_loss = []
for log in trainer.state.log_history:
  if 'loss' in log:
    training_steps.append(log['step'])
    training_loss.append(log['loss'])
```

**Preparing data for graphing:** This segment outputs the training loss for each step, using the records saved during training to create a clear picture of the model's progress.

```
# Krijimi i grafikut për humbjen e trajnimit
plt.figure(figsize=(10, 5))
plt.plot(training_steps, training_loss, label='Training Loss')
plt.xlabel('Hapat e Trajnimit')
plt.ylabel('Humbja')
plt.title('Humbja e Trajnimit me Kalimin e Kohës')
plt.legend()
plt.show()
```

**Loss visualization:** Loss visualization: The graph is created to show the progress of the loss during training. This visualization is important to understand when the model starts to stabilize and to observe any problems that may occur, such as overfitting or underfitting.

### 3.8. SPT interaction and accessibility

The importance of speech recognition technology lies in its broad potential to improve human-machine interaction, improve accessibility, drive healthcare operations, and facilitate multilingual communication.

The results of the transcribed samples show an overall improvement in the recognition of Albanian texts by the Whisper model. However, some phonetic and orthographic irregularities are still present. For example, the model shows a lack of precision on specific words such as "ndalja" (identified as "dolja") and "mjetet" (appeared as "mjetetet"). Improvement in these aspects may require increasing the dataset and applying advanced optimization techniques to capture the Albanian language's phonetic nuances better.

Sample 1:
Prediction: Vëntet kundaljot dolja e mjetetet.
Reference: Vendet ku ndalohet ndalja e mjetit.

Sample 5:
Prediction: Protokoli ujor oshto i perherjshom dhe perbanë redisën eddokumentasjon teknik.
Reference: Protokoli ujor është i përhershëm dhe përmban regjistrin e dokumentacionit teknik.

Sample 26:
Prediction: Te shkuar shun pak.
Reference: Të shkruar shumë pak.
Sample 31:
Prediction: Furnize me telefona mobir pur konsumatoret e ptikas.
Reference: Furnizim me telefona mobil per konsumatoret e ptk-se

Sample 76:
Prediction:  një marveshe pritët nejavët ardshma.
Reference: Një marrëveshje pritet në javët e ardhshme.

### 3.9. Challenges presented and possible causes

This project significantly contributes to speech recognition, especially for resource-constrained languages such as Albanian, by using advanced deep-learning architectures such as Whisper. Through model implementation and optimization, this study highlights the benefits and challenges of adapting large models to transcribe languages with limited datasets (Figure 1).

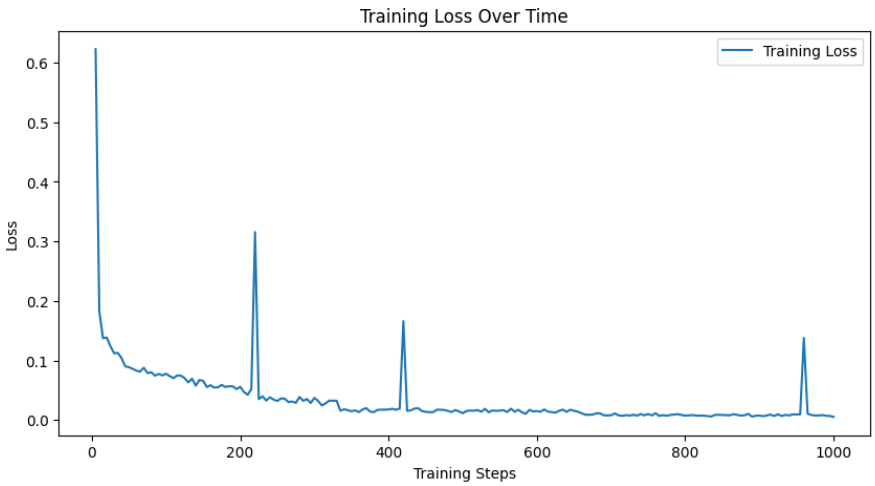


Figure 1: Training loss chart

The training loss plot shows a significant drop in the initial loss, indicating that the model quickly learned basic patterns in the early stages of training. This is often expected as the model learns simpler structures and connections. Occasional increases in loss may result from gradient accumulation or complex packets containing more difficult words to transcribe. However, the overall trend towards a low and stable loss indicates that the model is converging well, demonstrating the effectiveness of the training.

Results:
TrainOutput(global_step=50, training_loss=0.29685484886169433, metrics={'train_runtime': 123.9296, 'train_samples_per_second': 3.228, 'train_steps_per_second': 0.403, 'total_flos': 1.15434160128e+17, 'train_loss': 0.29685484886169433, 'epoch': 0.15186028853454822})

Based on this graph and the draft conclusions I have written, we can further strengthen the conclusions by emphasizing:

**Convergence and Stability of the Model:** The loss graph shows that the training process has successfully converged to a low final loss. This indicates that the model has learned the representations needed to transcribe the Albanian language. The stability of the loss at low levels towards the end of training also indicates that overfitting was minimal – a positive result, given the limitations of the dataset.

**Model Evaluation via Word Error Rate (WER):** After training, the Word Error Rate (WER) was calculated to evaluate the performance. A lower WER is ideal for practical applications, so mention the WER value achieved in this study and how this reflects on the model's usefulness for transcribing the Albanian language.

**Loss Growth Points:** At around 200, 400, and 1000 training steps, a sudden increase in loss can be observed. This could be due to particular recordings containing different accents or words with more complex structures, which require additional effort for the model to learn. Another cause could be the variation in the speakers' voices and pitch changes, making it more difficult for the model to maintain consistent performance.

**Characteristics of the Albanian Language**: The Albanian language has some complex phonetic elements, such as the unique sounds of certain letters and words with different structures. These characteristics require the model to learn variations in intonation and capture important nuances to generate accurate transcriptions.

*1. Increase in Batch Size*

**Current Configuration:** per_device_train_batch_size=1 and gradient_accumulation_steps=8

**Suggestion**: To keep memory capacity within limits, increase per_device_train_batch_size to 2 or 4 and adjust gradient_accumulation_steps.

A larger batch size creates a more stable gradient, which is important for large models like Whisper. As the batch size increases, the model has more data to make decisions on, which helps in learning general features and increases accuracy. Combining a higher batch size with gradient accumulation in memory-constrained conditions is useful for maintaining learning stability.

*2. Expanding the Dataset or Using Data Augmentation*

**Current Configuration:** Utilizing limited Albanian data from Common Voice.

**Suggestion:** If more data is available, broaden the dataset with additional Albanian recordings from different sources. Alternatively, apply data augmentation using techniques like SpecAugment, which processes audio features to enhance the model's robustness against variations. Employing amplified or augmented data will create a more resilient model and boost accuracy, particularly in resource-limited languages such as Albanian. SpecAugment modifies the audio spectrogram by masking specific frequencies or durations, making the model better equipped to handle variations in accent, intonation, and ambient noise.

*3. Optimizing Learning Speed and Using Learning Rate Scheduling*

**Current configuration:** learning_rate=1e-5, without a specific scheduler.

**Suggestion:** Experiment with different learning rate values, such as 5e-6 or 2e-5, to find the optimal balance for stability and improvement. Alternatively, a scheduler like get_linear_schedule_with_warmup can be applied, which gradually increases the rate at the beginning of training and decreases it during the final stages.

A learning rate scheduler helps improve model convergence, allowing Learning to occur at an optimal pace. During the beginning of training, a low learning rate can prevent oscillations, while in the final stages it can improve the model's focus on small audio details.

*4. Increasing the Maximum Training Steps or Number of Epochs*

**Current Configuration:** max_steps=1000

**Suggestion:** Increase max_steps to 2000 or more if resources allow. Alternatively, you can specify num_train_epochs instead of max_steps for better control when the dataset size changes.

Increasing the number of training steps will allow the model to have a deeper approach to the dataset and improve its convergence, increasing the overall accuracy of the transcription. Longer training helps extract more phonetic features and linguistic details of Albanian.

*5. Application of Mixed Precision Training*

**Current Configuration:** fp16=False

**Suggestion:** Enable fp16=True to enable mixed precision training.

Mixed precision training reduces memory usage and speeds up training, allowing for larger batch sizes or longer training times. This optimization is especially useful for large models like Whisper, and helps conserve resources without sacrificing much accuracy.

*6. Experimenting with Dropout Regularization*

**Current Configuration:** Overall dropout is 0.

**Suggestion:** You can add dropout to the attention layers (attention_dropout=0.1) and feed-forward layers (dropout=0.1).

Dropout helps prevent overfitting by temporarily stopping some neurons during training, which helps the model generalize better. This method is especially useful for small or low-variance datasets.

*7. Applying Weight Decay for Better Generalization*

**Tip**: Add a weight_decay parameter to the optimizer, usually with values like 0.01 or 0.1.

Weight decay is a regularization technique that helps avoid overestimating large weights. This leads to a simpler model and reduces the risk of overfitting. This parameter is handy for long training runs or datasets of limited size, increasing the model's ability to generalize better to unknown data.

*8. Regular Assessment and Maintenance Checkpointing*

**Current Configuration:** eval_steps=400, save_steps=500

**Suggestion:** Reduce eval_steps to a more frequent value like 200 and use save_total_limit to control the number of points saved.

More frequent evaluation allows for better model performance monitoring, especially when experimenting with new configurations. Saving regular and limited checkpoints saves resources and helps to preserve good points, which can be used for comparisons and optimizations at different stages of training.

## CONCLUSIONS

Developing localized speech recognition for Albanian is essential for communities in Albania, Kosovo, North Macedonia, Montenegro, Serbia, Greece, and the global diaspora. This technology addresses technical challenges and highlights cultural significance, demonstrating a commitment to linguistic diversity and technological equity in the digital age. It also serves as a model for AI and machine learning advantages across various language communities, expanding the global reach of speech technology.

Effective speech recognition technology improves access to digital services and educational resources, reducing language barriers and promoting inclusion. In academia, it supports language learning by providing tools for pronunciation, grammar, and vocabulary. Economically, effective speech recognition enhances customer service in

businesses and streamlines government services for Albanian speakers, contributing to the reduction of technological inequality. Customized deep learning models for Albanian will ensure equal access to AI and speech recognition advancements for all language groups.

Studying Albanian phonetic diversity and syntax can enhance the design of speech recognition systems. This study involves evaluating deep learning architectures like CNNs, RNNs, and transformer models to assess their effectiveness in understanding spoken Albanian. Additionally, optimizing models through techniques such as attention mechanisms, transfer learning, and model compression seeks to improve speed and accuracy in real-world applications.

The Albanian dataset in Common Voice has limited data, which hinders the model's capacity to learn about various speech variations, including dialects and regional accents. The large Whisper model requires substantial memory and processing power, complicating efficient training and implementation on smaller devices. Employing attention mechanisms to improve the model's accuracy adds complexity to the optimization process. Delivering quality results for low-resource languages demands considerable time and effort. Languages like Albanian contain phonetic and grammatical features that challenge universal models, often resulting in overlooked details and nuances.

## REFERENCES


[1] Agalliu, F., Angoni, E., & Demiraj, Sh. (2002). *Gramatika e gjuhës shqipe 1: Morfologjia*. Instituti i Gjuhësisë dhe i Letërsisë (Akademia e Sciences of Albania), (In Albanian).

[2] Belinkov, Y., Ali, A., & Glass, J. (2019). Analyzing phonetic and graphemic representations in end-to-end automatic speech recognition. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2019-September, 81-85. https://doi.org/10.21437/Interspeech.2019-2599

[3] Buchholz, O., & Fiedler, W. (1987). *Albanische Grammatik*. VEB Verlag Enzyklopädie.

[4] Chaudhary, A. (2022). *Automatic Extraction and Application of Language Descriptions for Under-Resourced Languages* (Doctoral dissertation, Carnegie Mellon University).

[5] Dedvukaj, L., & Gehringer, P. (2023). Morphological and phonological origins of Albanian nasals and its parallels with other laws. *Proceedings of the Linguistic Society of America*, *8*(1), 5508-5508.

[6] Dozat, T., Qi, P., & Manning, C. D. (2017). Stanford graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*.

[7] Ejigu, Y. A., and Tesfa T. A. (2024) Enhancing Amharic Speech Recognition in Noisy Conditions through End-to-End Deep Learning, https://doi.org/10.20944/preprints202402.0754.v1.

[8] Graves, A., Mohamed, A., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks.

[9] Greig, J., Rehman, S-U., Ul-Haq, A., Dresser, G. and Burmeister, O. K. (2019). Transforming Ageing in Community: addressing global ageing vulnerabilities through smart communities. In Proceedings of the 9th International Conference on Communities & Technologies - Transforming Communities (C&T '19). Association for Computing Machinery, New York, NY, USA, 228–238. https://doi.org/10.1145/3328320.3328380.

[10] Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., & Pang, R. (2020). Conformer: Convolution-augmented Transformer for speech recognition. arXiv preprint arXiv:2005.08100.

[11] Hamp, E. P. (2016). *Albanian language*. Encyclopedia Britannica.

[12] Hepsiba, D., & Justin, J. (2019). Role of deep neural network in speech enhancement: A review. In: Hemanth, J., Silva, T., Karunananda, A. (eds) Artificial Intelligence. SLAAI-ICAI 2018. Communications in Computer and Information Science, vol 890. Springer, Singapore. https://doi.org/10.1007/978-981-13-9129-3_8.

[13] Kamath, U., Liu, J., & Whitaker, J. (2019). *Deep learning for NLP and speech recognition*. Springer. https://doi.org/10.1007/978-3-030-14596-5

[14] Kote, N., Biba, M., Kanerva, J., Rönnqvist, S., & Ginter, F. (2020). Morphological tagging and lemmatization of Albanian: A manually annotated corpus and neural models. *Proceedings of the International Conference on Language Resources and Evaluation (LREC'20)*.

[15] Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., Mori, R. D., & Bengio, Y. (2021). SpeechBrain: A general-purpose speech toolkit. arXiv:2106.04624.

[16] Luhman, T., & Luhman, E. (2022). Improving Diffusion Model Efficiency Through Patching. arXiv:2207.04316v1, https://doi.org/10.48550/arXiv.2207.04316

[17] Mehrish, A., Majumder, N., Bhardwaj, R., Mihalcea, R., & Poria, S. (2023). A Review of Deep Learning Techniques for Speech Processing. Singapore University of Technology and Design, Singapore; University of Michigan, USA.

[18] Menghani, G. (2023). *Efficient deep Learning: A survey on making deep learning models smaller, faster, and better*. ACM Computing Surveys. https://doi.org/10.1145/3578938

[19] Newmark, L. (Ed.). (1999). *Oxford Albanian-English dictionary*. Oxford University Press.

[20] Nirenburg, S. (2009). *Language engineering for lesser-studied languages*. Volume 21 Nato Science For Peace And Security Series D Information And Communication Security And Communications Security Vol 20, IOS Press, pp. 344.

[21] Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books, 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, pp. 5206-5210, doi: 10.1109/ICASSP.2015.7178964.

[22] Rista, A., & Kadriu, A. (2022). A model for Albanian speech recognition using end-to-end deep learning techniques. *Interdisciplinary Journal of Research and Development*, *9*(3), 1, https://doi.org/10.56345/ijrdv9n301.

[23] Sainath, T.N., Vinyals, O., Senior, A., & Sak, H. (2015). "Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), South Brisbane, QLD, Australia, pp. 4580-4584, doi: 10.1109/ICASSP.2015.7178838.

[24] Sak, H., Senior, A., & Beaufays, F. (2019). Long short-term memory recurrent neural network architectures for Urdu acoustic modeling. *Int. J. Speech Technol.* 22, 1 , 21–30. https://doi.org/10.1007/s10772-018-09573-7

[25] Schultz, T., & Kirchhoff, K. (2006). Multilingual speech processing. Elsevier. Amsterdam, Netherlands, pp. 508, https://doi.org/10.1016/B978-0-12-088501-5.X5000-8.

[26] Siddegowda, S.M., Fournarakis, M., Nagel, M., Blankevoort, T., Patel, C., & Khobare, A. (2022). Neural Network Quantization with AI Model Efficiency Toolkit (AIMET). ArXiv, abs/2201.08442.

[27] Zhang, Y., Han, W., Qin, J., Wang, Y., Bapna, A., Chen, Z., & Wu, Y. (2023). Google usm: Scaling automatic speech recognition beyond 100 languages. arXiv preprint arXiv:2303.01037. https://doi.org/10.48550/arXiv.2303.01037.