NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

M.Sc. in Language Technology

MSc THESIS

Automatic Lyrics Transcription for Greek Songs

Maria Frangiadaki

Supervisor: **Kosmas Kritsis,** Associate Researcher, Institute for Language and Speech Processing, Athena Research Center

ATHENS

OCTOBER 2025

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΙΔΡΥΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΓΛΩΣΣΙΚΗ ΤΕΧΝΟΛΟΓΙΑ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αυτόματη Μεταγραφή Στίχων για Ελληνικά Τραγούδια

Μαρία Φραγγιαδάκη

Επιβλέπων: **Κοσμάς Κρίτσης,** Συνεργαζόμενος Ερευνητής, Ερευνητικό Κέντρο «Αθηνά», Ινστιτούτο Επεξεργασίας του Λόγου (ΙΕΛ)

ΑΘΗΝΑ

Οκτώβριος 2025

MSc THESIS

Automatic Lyrics Transcription for Greek Songs

Maria Frangiadaki

S.N.: 7115182300028

SUPERVISOR: **Kosmas Kritsis,** Associate Researcher, Institute for Language and Speech Processing, Athena Research Center

EXAMINATION COMMITTEE:

Vassilis Katsouros, Researcher Director**,** Institute for Language and Speech Processing, Athena Research Center
Georgios Paraskevopoulos, Associate Researcher, Institute for Language and Speech Processing, Athena Research Center
Kosmas Kritsis, Associate Researcher, Institute for Language and Speech Processing, Athena Research Center

Examination Date: 20 OCTOBER, 2025

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αυτόματη Μεταγραφή Στίχων σε Ελληνικά Τραγούδια

Μαρία Φραγγιαδάκη

Α.Μ.: 7115182300028

SUPERVISOR: **Κοσμάς Κρίτσης**, Συνεργαζόμενος Ερευνητής, Ινστιτούτο Επεξεργασίας του Λόγου (ΙΕΛ), Ερευνητικό Κέντρο «Αθηνά»

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: **Βασίλης Κατσούρος, Διευθυντής Ερευνών, Ινστιτούτο Επεξεργασίας του Λόγου (ΙΕΛ), Ερευνητικό Κέντρο «Αθηνά»**
Γιώργος Παρασκευόπουλος, Συνεργαζόμενος Ερευνητής, Ινστιτούτο Επεξεργασίας του Λόγου (ΙΕΛ), Ερευνητικό Κέντρο «Αθηνά»
Κοσμάς Κρίτσης, Συνεργαζόμενος Ερευνητής, Ινστιτούτο Επεξεργασίας του Λόγου (ΙΕΛ), Ερευνητικό Κέντρο «Αθηνά»

Ημερομηνία Εξέτασης 20 Οκτωβρίου 2025

ABSTRACT

This thesis investigates adapting Whisper models to Automatic Lyrics Transcription (ALT) for Greek singing voice. The study pursues two main goals: (i) achieving robust transcription of Greek lyrics from polyphonic music, and (ii) examining whether multitask and staged fine-tuning strategies can improve generalization and domain adaptation to singing.

The experiments are based on the Greek Audio Dataset (GAD), an existing corpus to which a processing pipeline tailored for ALT research is curated. Specifically, CTC-based forced alignment is performed to obtain time-aligned lyric segments, Greek lines are paired with their English translations, and the material is converted into segment-level Hugging Face datasets with consistent pre-/post-processing. This leads to an aligned, segment-level dataset with Greek audio slices and paired Greek and English transcriptions. Then a five-regime comparison is conducted: (i) zero-shot Whisper, (ii) transcription-only fine-tuning, (iii) multi-task fine-tuning with task-pure batch alternation, (iv) two-stages fine-tuning -firstly on CommonVoice Greek-speech dataset and then on the Greek songs-, and (v) model scaling (Small/Medium/Large).

Zero-shot Whisper-Medium scores 92,1% WER in lyric transcription, highlighting the speech-to-singing gap. After adaptation, Whisper-Large (transcribe-only) attains 30% WER, a ~ 67% relative error reduction over the zero-shot baseline. For Whisper-Small, a 2:1 transcribe: translate mix (33,6%) marginally outperforms both transcription-only (36,7%) and 4:1 (34,9%), indicating that a moderate auxiliary translation signal regularizes low-capacity models. In contrast, medium and large models favor the transcribe-only regime, suggesting that higher-capacity models benefit from specialized singing adaptation.

Error analysis reveals singing-specific phenomena, such as consonant-cluster inflation in rhythmically dense lines, vowel confusions induced by melisma, function-word deletions/insertions, and occasional lyric-prior hallucinations. Larger models reduce severity rather than type of errors, explaining the WER gains. The staged fine-tuning scheme produced no additional gains, implying that Whisper already exhibits strong multilingual and multi-acoustic generalization, and the possible insufficiency of the Common Voice dataset.

This work constitutes the first systematic ALT study of Greek songs. Future work includes expanding the Greek dataset, incorporating expressive features and enabling Large Language Models (LLMs) like Llama-Krikri for post-processing of transcriptions.

Subject Area: Speech & Audio Processing; Language Technology; Multimodal Modeling for Music and Lyrics

Keywords: Automatic lyrics transcription, Whisper, Multi-task learning, Singing voice, Greek, Low-resource language

ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία εστιάζει στην ανάπτυξη και προσαρμογή συστημάτων Αυτόματης Μεταγραφής Στίχων, για τη νεοελληνική τραγουδιστική φωνή, αξιοποιώντας τις δυνατότητες των πολυγλωσσικών μοντέλων Whisper. Ο στόχος της έρευνας είναι διττός. Αφενός η αξιόπιστη μεταγραφή ελληνικών στίχων από τραγούδια και η διερεύνηση διαφορετικών στρατηγικών εκπαίδευσης, όπως η multitask (μεταγραφή - μετάφραση) και η εκπαίδευση δύο σταδίων, ως μέσο ενίσχυσης της γενίκευσης και προσαρμοστικότητας των μοντέλων στα τραγούδια.

Η Αυτόματη Αναγνώριση Ομιλίας έχει σημειώσει εντυπωσιακή πρόοδο τα τελευταία χρόνια, χάρη στην εμφάνιση μεγάλων μοντέλων τύπου Transformer, όπως τα wav2vec 2.0, XLS-R και Whisper, τα οποία επιτρέπουν εκπαίδευση end-to-end χωρίς την ανάγκη χειροκίνητων χαρακτηριστικών. Ωστόσο, η τραγουδιστική φωνή παρουσιάζει ιδιαιτερότητες που δυσχεραίνουν την απευθείας εφαρμογή αυτών των συστημάτων. Οι μεγάλες φωνητικές διακυμάνσεις, οι παρατεταμένες διάρκειες φωνηέντων, η μελωδική αστάθεια, η συνοδεία μουσικής και η ελεύθερη προσωδία διαταράσσουν τις ακουστικές και στατιστικές υποθέσεις των μοντέλων που έχουν εκπαιδευτεί σε καθαρή ομιλία. Κατά συνέπεια, παρατηρείται σημαντική πτώση της ακρίβειας, γεγονός που καθιστά απαραίτητη την εξειδικευμένη προσαρμογή των μοντέλων στην τραγουδιστική φωνή.

Ως βάση χρησιμοποιείται ένα segment-level σύνολο δεδομένων σε μορφή Hugging Face, το οποίο περιλαμβάνει ελληνικά ηχητικά αποσπάσματα και αντιστοιχισμένους ελληνικούς και αγγλικούς στίχους. Τα δεδομένα προέρχονται από πολυφωνικές ηχογραφήσεις ελληνικών τραγουδιών, οι οποίες απομονώθηκαν σε καθαρά φωνητικά κανάλια μέσω του μοντέλου source-separation Demucs_ft. Στη συνέχεια, οι στίχοι ευθυγραμμίστηκαν αυτόματα με τα αντίστοιχα ηχητικά αποσπάσματα με χρήση CTC-based forced aligner. Η διαδικασία προεπεξεργασίας περιλάμβανε εξισορρόπηση έντασης, επαναδειγματοληψία στα 16 kHz, καθαρισμό ειδικών χαρακτήρων και μετατροπή της δομής Kaldi σε μορφή HF Dataset (με πεδία audio, transcription, translation), γεγονός που επιτρέπει την απευθείας αξιοποίησή τους στη βιβλιοθήκη Transformers.

Η μεθοδολογία που ακολουθήθηκε οργανώθηκε σε τέσσερα κύρια στάδια. Στο πρώτο στάδιο πραγματοποιήθηκε zero-shot αξιολόγηση του Whisper-Medium χωρίς καμία προσαρμογή, ώστε να υπολογιστεί η βασική επίδοση του μοντέλου στην τραγουδιστική φωνή. Στο δεύτερο στάδιο το μοντέλο εκπαιδεύτηκε αποκλειστικά για μεταγραφή στα ελληνικά, με στόχο την προσαρμογή του στα φωνητικά χαρακτηριστικά του τραγουδιού. Στο τρίτο στάδιο εφαρμόστηκε πολυ-λειτουργική εκπαίδευση (multitask learning), συνδυάζοντας ταυτόχρονα μεταγραφή και μετάφραση με αναλογίες batches 1:1, 2:1 και 4:1, προκειμένου να διερευνηθεί αν η μετάφραση λειτουργεί ως μορφή κανονικοποίησης της εκπαίδευσης. Τέλος, στο τέταρτο στάδιο εξετάστηκε η κλιμάκωση των ίδιων ρυθμίσεων σε τρία μεγέθη μοντέλων (Small, Medium, Large), ώστε να διερευνηθεί η σχέση μεταξύ χωρητικότητας του μοντέλου και απόδοσης. Επιπρόσθετα, διερευνήθηκε ένα staged fine-tuning σχήμα, στο οποίο το μοντέλο προσαρμόζεται αρχικά σε δεδομένα ομιλίας (Greek Common Voice) και στη συνέχεια σε δεδομένα τραγουδιού, με στόχο τη σταδιακή μεταφορά γνώσης από το πεδίο της ομιλίας στο πεδίο του τραγουδιού.

Η εκπαίδευση πραγματοποιήθηκε σε GPU περιβάλλοντα υψηλής υπολογιστικής ισχύος (Borges και Leonardo supercomputer) με χρήση του Seq2SeqTrainer της βιβλιοθήκης

Transformers, batch size 4-8, μικτή ακρίβεια FP16 και optimizer AdamW (learning rate 5×10^{-5}). Η αξιολόγηση της απόδοσης βασίστηκε στη μετρική normalized Word Error Rate (WER), η οποία εξουδετερώνει την επίδραση της στίξης και των μικρο-ορθογραφικών διαφορών.

Πειραματικά, το zero-shot Whisper-Medium σημείωσε WER 92,1%, αναδεικνύοντας το έντονο χάσμα ανάμεσα στη φωνή ομιλίας και τραγουδιού. Μετά το fine-tuning, το Whisper-Large (transcribe-only) πέτυχε WER 30%, επιτυγχάνοντας περίπου 67% μείωση σφάλματος σε σχέση με το baseline. Η multitask εκπαίδευση αποδείχθηκε ωφέλιμη για μικρότερα μοντέλα λόγω της γλωσσικής κανονικοποίησης που προσφέρει, ωστόσο στα μεγαλύτερα μοντέλα η απλή εκπαίδευση μεταγραφής παρήγαγε πιο σταθερά και καθαρά αποτελέσματα. Η 2-stages εκπαίδευση δεν οδήγησε σε περαιτέρω βελτίωση, υποδηλώνοντας ότι το Whisper διαθέτει ήδη εγγενή πολυγλωσσική και ακουστική γενίκευση, καθώς και ότι το μικρό διαθέσιμο ελληνικό Common Voice σύνολο δεδομένων δεν επαρκούσε.

Η ποιοτική ανάλυση των παραγόμενων μεταγραφών φανερώνει χαρακτηριστικά μοτίβα που σχετίζονται με τη φύση του τραγουδιού, όπως διόγκωση συμφωνικών συμπλεγμάτων σε δύσκολες φράσεις, σύγχυση φωνηέντων και ορθογραφικά λάθη, αποσιώπηση μικρών λειτουργικών λέξεων, και περιστασιακές «παραισθήσεις» στίχων όταν το ηχητικό σήμα είναι αμφίβολο. Παρότι τα μεγαλύτερα μοντέλα δεν εξαλείφουν πλήρως τα λάθη, μειώνουν αισθητά τη σοβαρότητά τους, οδηγώντας σε σημαντική βελτίωση του WER και πιο φυσικές μεταγραφές.

Η παρούσα εργασία αποτελεί την πρώτη συστηματική μελέτη ALT για ελληνικά τραγούδια και αποδεικνύει ότι τα μοντέλα Whisper μπορούν να προσαρμοστούν αποτελεσματικά στην τραγουδιστική φωνή μέσω στοχευμένου fine-tuning, χωρίς ανάγκη εκπαίδευσης από το μηδέν. Η multitask μάθηση αποδεικνύεται χρήσιμη μόνο σε μικρότερα μοντέλα, ενώ η εκπαίδευση αποκλειστικά για μεταγραφή παραμένει η πιο αποδοτική στρατηγική για τα μεγαλύτερα μοντέλα.

Μελλοντικά, προτείνεται η επέκταση του ελληνικού συνόλου δεδομένων με περισσότερα είδη μουσικής και μεγαλύτερη ποικιλία φωνών, η αξιοποίηση ημι-επιβλεπόμενων τεχνικών (semi-supervised, pseudo-labeling) για την αύξηση του όγκου των δεδομένων, και η διερεύνηση παραμετρικά αποδοτικών μεθόδων fine-tuning, όπως LoRA και adapters, ώστε να μειωθεί το υπολογιστικό κόστος. Ιδιαίτερα ενδιαφέρουσα κατεύθυνση συνιστά η ενσωμάτωση μεγάλων γλωσσικών μοντέλων (LLMs), όπως το Krikri, ή GPT, για τη μετα-επεξεργασία των μεταγραφών. Τέτοια μοντέλα μπορούν να χρησιμοποιηθούν για διόρθωση λαθών, ορθογραφική και γραμματική εξομάλυνση, καθώς και για αναδόμηση των στίχων με βάση τη ρυθμική και νοηματική τους συνοχή, επεκτείνοντας τη λειτουργικότητα του συστήματος πέρα από το επίπεδο της απλής αναγνώρισης.Τέλος, προτείνεται η ενσωμάτωση εκφραστικών χαρακτηριστικών όπως ύψος (pitch), ρυθμός και συναίσθημα, προκειμένου το σύστημα να συλλάβει πληρέστερα τη μουσική διάσταση της φωνής.

Θεματική περιοχή: Επεξεργασία Ομιλίας & Ήχου, Γλωσσική Τεχνολογία, Αυτόματη Αναγνώριση Φωνής

Λέξεις-κλειδιά: Αυτόματη μεταγραφή στίχων, Whisper, Multitask learning, Τραγουδιστική φωνή, Ελληνική γλώσσα

Dedicated to my brother, Dimitris

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor, Kosmas Kritsis, Associate Researcher at the ILSP Athena Research Center, for his guidance and encouragement and continuous support throughout this thesis. I would also like to thank Dimitris Damianos, Associate Researcher at Athena Research Center, for his assistance, technical guidance and support throughout this project.

I am further grateful to the research team at the Athena Research Center for providing computational resources and technical assistance during the thesis' experiments, particularly on the BOrges Cluster. Also, I acknowledge the EuroHPC Joint Undertaking for awarding this project access to the EuroHPC supercomputer LEONARDO, hosted by CINECA (Italy) and the LEONARDO consortium through an EuroHPC Development Access call.

I would also like to thank my friends and classmates from the Master's program, who made these two years an inspiring experience. Finally, I would like to thank my family for their love, patience, and belief in me throughout my studies. This work marks the completion of a meaningful two-year journey and is dedicated to them.

TABLE OF CONTENTS

I. Introduction 25

1. Motivation 25

2. Research Objective & Contribution 25

3. Outline 26

II. Theoretical Part: Background and Related Research 26

2.Background Knowledge 26

2.1 Machine Learning and Deep Learning 26

2.2 Automatic Speech Recognition (ASR) 32

3. Related Work 36

3.1 ALT 36

3.2 Lyrics Alignment 38

3.3 Low-Resource and Greek-Specific Research 40

III. PRACTICAL PART: EXPERIMENTS 41

4. Methodology 41

4.1 Overview 41

4.2 Dataset 42

4.3 Source Separation with Demucs 43

4.4 Kaldi Data Conversion 43

4.5 Lyrics Alignment and Segmentation 43

4.6 Translation 44

4.7 Kaldi to Hugging Face Dataset Conversion 44

4.8 Lyrics Transcription with Whisper 44

4.9 Fine-Tuning Whisper for Lyrics Transcription and Translation 45

4.10. Staged fine tuning 45

4.11 Testing & Evaluation Protocol 46
5: Experiments & Results 46
5.1 Zero-Shot Baseline (Before Fine-Tuning) 46
5.2 Results Overview on Fine-Tuned models 47
IV: Conclusions & Future Work 52
6. Conclusions 52
7. Limitations and Future Work 53
ABBREVIATIONS – ACRONYMS 54
References 55

LIST OF FIGURES

Figure 1 ……..28

Figure 2 ……..29

Figure 3 ……..29

Figure 4 ……..30

Figure 5 ……..31

Figure 6 ……..32

Figure 7 ……..33

Figure 8 ……..42

Figure 9 ……..42

LIST OF TABLES

Table 1 ……..…..47

Table 2 ……..……..48

Table 3 ……..…..49

PREFACE

This thesis was completed as part of my Master's studies in Language Technology and explores the automatic transcription of Greek song lyrics. The project grew from a personal interest in how music and language intertwine, and from the practical value of reliable lyric transcriptions in research and creative applications. The work reflects the contributions of many people. I am deeply grateful to everyone who contributed guidance, feedback, and technical support throughout this work. Their support shaped both the scope and the rigor of the research. I hope the findings and resources presented here will prove useful to others working toward a deeper understanding and more effective processing of the Greek singing voice.

I. Introduction

1. Motivation

Automatic Speech Recognition (ASR) has enabled a broad spectrum of everyday applications, from voice assistants and dictation to searchable media archives. However, ASR systems often degrade sharply when the deployment domain differs from the training distribution. Such domain shifts can stem from environmental noise, recording conditions, speaker accent and vocabulary, and -crucially for this work- from singing voice, where pitch excursions, melisma (vowel elongation), instrumental accompaniment, and expressive articulation violate the assumptions learned from speech. This work addresses Automatic Lyrics Transcription (ALT) for Greek singing, focusing on Whisper as a strong pretrained baseline.

In the context of ASR, domain adaptation ranges from fully supervised fine-tuning on in-domain labels to weakly/unsupervised schemes that leverage large unlabeled corpora. While these strategies have improved robustness, singing voice remains particularly challenging, due to its linguistic deviations from normal speech. For low-resourced languages, the problem is amplified by limited labeled resources, especially in musical or singing domains. This thesis addresses these difficulties by studying how and when adaptation of Whisper models improves ALT for Greek songs, providing insights into domain and language adaptation strategies for low-resource singing ASR.

2. Research Objective & Contribution

This thesis's primary goal is to explore how and to what extent can Whisper adaptation and task design improve lyrics transcription accuracy in a low-resource language such as Greek, particularly for singing voice, a task that remains challenging due to both the acoustic variability of singing and the scarcity of labeled data in low-resourced languages. To address this question, the following objectives are set:

1. Fine-tune Whisper for the Greek singing voice domain and analyze how adaptation from speech to singing influences recognition accuracy.

2. Conduct a controlled comparison across Whisper Small, Medium, and Large checkpoints, contrasting transcription-only fine-tuning with multi-task training (transcribe:translate ratios of 1:1, 2:1 and 4:1) to quantify how task composition and training curricula interact with model scale.

3. Evaluate a two-staged fine-tuning approach that first adapts on Greek speech before specializing on singing voice.

4. Assess performance using normalized Word Error Rate (WER) and a structured error analysis tailored to the unique properties of singing.

Contributions of this thesis are:

1. A controlled study of task composition and model scale for Greek ALT,

2. a task-pure batching scheme with language-aware preprocessing that stabilizes training on singing voice,

3. quantitative and qualitative error taxonomy tailored to Greek lyrics, and

4. evidence that large, transcription-only adaptation is the most effective path for Greek ALT, while multi-task learning serves as useful regularization only at small capacity.

3. Outline

The thesis is structured as follows: Chapter 2 reviews foundational concepts in machine learning and deep learning relevant to ASR, with emphasis on encoder–decoder Transformers and data normalization for sequence generation. Chapter 3 discusses the specific challenges of ASR on singing voice and summarizes prior approaches to lyrics transcription, alignment and low resource specific research. Chapter 4 presents our methodology: datasets, preprocessing, training schedules (transcription-only, multi-task with 2:1 and 4:1 ratios), staged fine-tuning, decoding, and evaluation metrics; Chapter 5 reports experimental results and an error analysis, while Chapter 6 concludes with key findings, limitations and directions for future work.

II. Theoretical Part: Background and Related Research

2.Background Knowledge

2.1 Machine Learning and Deep Learning

Machine Learning (ML) focuses on developing algorithms capable of automatically discovering patterns and regularities in data, without being explicitly programmed. In contrast to rule-based systems, which rely on human-crafted heuristics, ML algorithms learn from examples, improving their performance as they are exposed to more data. Deep Learning (DL), a subfield of ML, leverages multi-layer neural networks that can learn hierarchical representations, enabling systems to automatically extract high-level features from raw data. These techniques have transformed fields such as Natural Language Processing (NLP), and particularly speech recognition, forming the foundation upon which current Automatic Speech Recognition (ASR) systems are built.

The aim of this section is to present the theoretical principles of machine and deep learning that underpin ASR. It begins with a conceptual overview of learning paradigms, followed by the mathematical fundamentals of neural computation, model training, and optimization. It then discusses advanced deep learning architectures, attention mechanisms, and the challenges of generalization and representation learning.

2.1.1 Definition

Machine Learning has three key elements: i) the task, ii) the experience, and iii) the performance measure. The goal of an ML algorithm is to build a predictive model that

maps an input vector (x) to an output (y). Learning occurs by adjusting parameters so as to minimize a defined loss function, which quantifies the variance between predictions and ground truth. In the context of ASR, the task involves transcribing speech into text, the experience corresponds to exposure to large collections of labeled speech data, and the performance measure is typically accuracy or WER.

2.1.2 Types of Learning

Machine learning can be categorized into several paradigms, depending on the availability of labeled data and the nature of the learning process. Supervised Learning involves learning from pairs of inputs and corresponding outputs. The objective is to find a mapping function that generalizes well to unseen examples. ASR systems typically fall into this category, as they are trained on paired audio and transcription data. Unsupervised Learning deals with unlabeled data, discovering hidden structures or patterns within it. Techniques such as clustering, dimensionality reduction and autoencoders are commonly used. In modern speech processing, unsupervised learning plays a crucial role in pretraining models through self-supervised approaches. Semi-Supervised Learning combines both labeled and unlabeled data, leveraging pseudo-labels or consistency regularization to improve performance when annotations are scarce. Finally, Reinforcement Learning (RL) focuses on agents that learn through interaction with an environment. The agent receives rewards or penalties and gradually learns a strategy that maximizes the reward. Though less common in ASR, RL has been explored for optimizing decoding strategies and dialogue system responses.

2.1.3 Fundamentals of Neural Computation

Artificial Neural Networks (ANNs) constitute a core approach within Machine Learning, forming the foundation of most modern Deep Learning methods. They are computational frameworks inspired by the structure and function of biological neurons. Each neuron receives one or more inputs, applies a linear transformation using weights and a bias term, and then passes the result through a nonlinear activation function. The introduction of nonlinear activation functions enabled networks to capture complex, non-linear relationships. Common functions include:

- Sigmoid: $\sigma(x) = 1/(1 + e^{-x})$
- Hyperbolic tangent: $tanh(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x})$
- Rectified Linear Unit (ReLU): $ReLU(x) = max(0, x)$

Modern deep networks primarily use ReLU and its variants because they mitigate the vanishing-gradient problem, allowing efficient optimization of deep models. A feed-forward neural network (or multilayer perceptron) consists of layers of neurons organized sequentially, as shown in Figure 1.

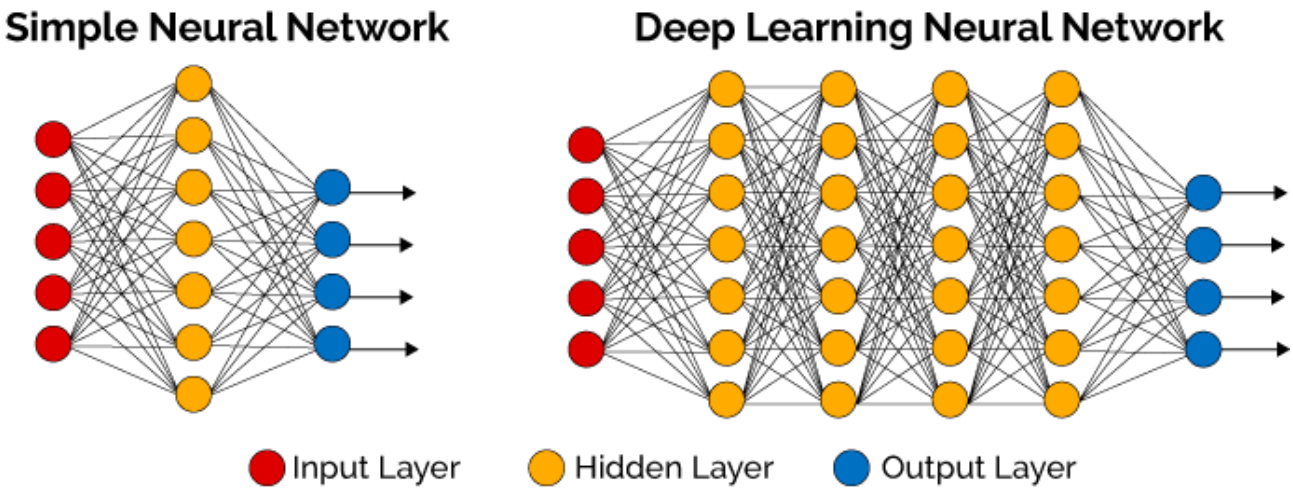

Figure 1: Basic architecture of a feed-forward neural network

2.1.4 Model Training

Training a neural network involves optimizing its parameters to minimize a chosen loss function. The loss quantifies how far predictions deviate from true labels. Common loss functions include:

- Mean Squared Error (MSE) for regression tasks:

$$L = (1/N)\sum_i (y_i - \hat{y}_i)^2$$

- Cross-Entropy Loss for classification:

$$L = -\sum_i y_i \log(\hat{y}_i)$$

Optimization is typically performed using gradient descent, which updates model parameters in the direction that reduces the loss. Variants such as Stochastic Gradient Descent (SGD), Adam and others adapt learning rates dynamically and help avoid local minima.

2.1.5 Deep Architectures

As networks grew deeper, they gained the ability to model highly complex functions but also became more challenging to train. Recurrent Neural Networks (RNNs) introduced the concept of temporal recurrence to process sequential data. Each time step's hidden state (h_t) is computed as:

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

Their basic architecture is depicted in FIgure 2. RNNs capture temporal dependencies but often suffer from vanishing or exploding gradients when modeling long sequences.

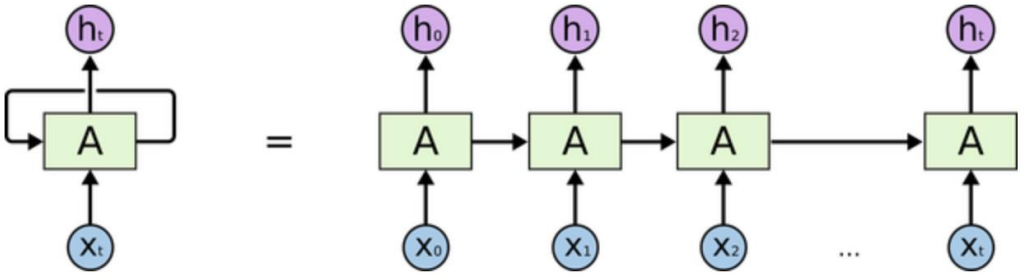

Figure 2: Basic Architecture of Recurrent Neural Networks

To address this, Long Short-Term Memory (LSTM) networks [1] introduced memory cells and gating mechanisms. Figure 3 shows these gates -input, forget, output- control which information to keep, update, or output, enabling the network to preserve long-term dependencies more effectively. Gated Recurrent Units (GRUs) [2] further simplify LSTMs by merging gates. GRUs maintain similar accuracy while requiring fewer parameters, making them computationally efficient. In addition to these architectures, Bidirectional LSTMs (BiLSTMs) [3] became particularly influential in ASR and NLP. As depicted in Figure 4, these models process sequences in both forward and backward directions, allowing the network to leverage context from both past and future tokens, which significantly improves performance in tasks where full-sequence context is essential.

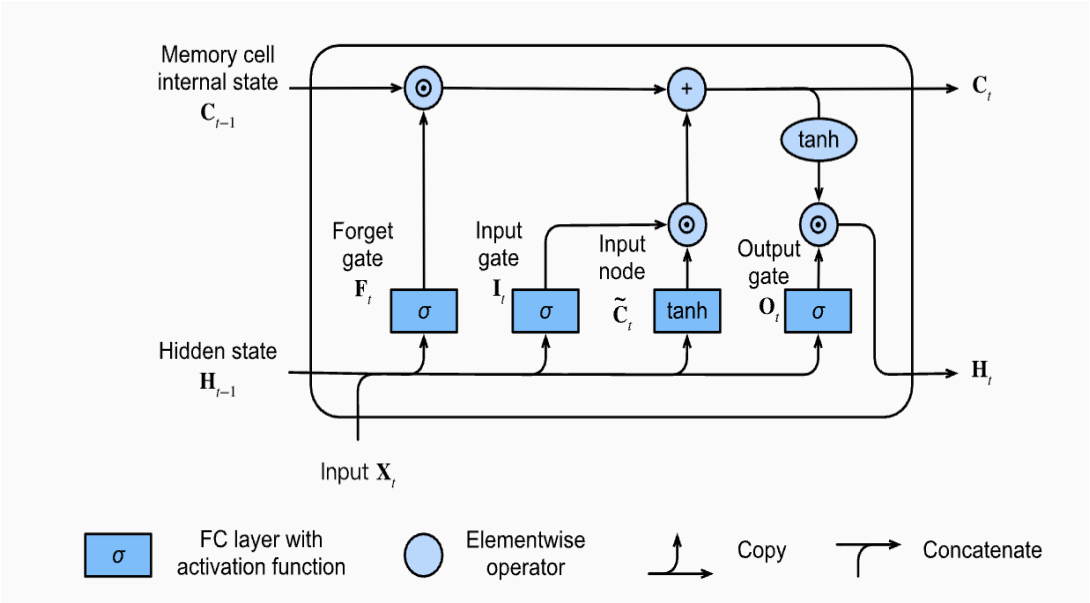

Figure 3: Structure of an LSTM cell showing the flow of information through input, forget and output gates.

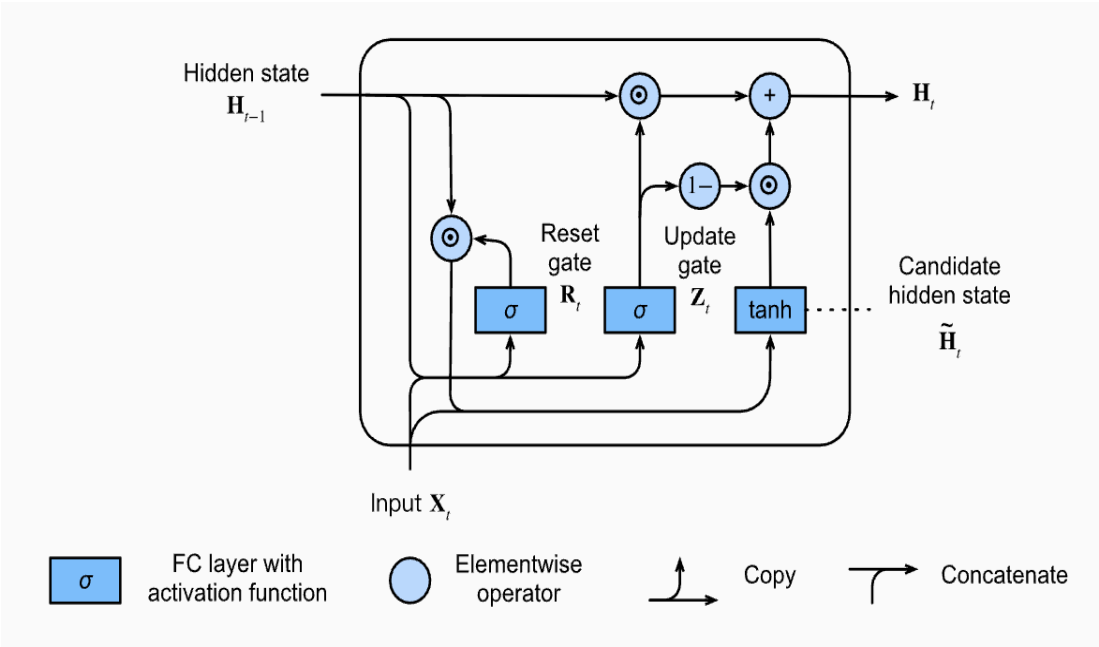

Figure 4: Gated Recurrent Unit (GRU model showing the update and reset gates controlling hidden-states

2.1.6 Attention and Transformers

The introduction of attention mechanisms marked a major milestone in deep learning. Attention allows the model to dynamically focus on the most relevant parts of an input sequence when generating each output. Traditional recurrent networks compress all contextual information into a single hidden vector. Rather than relying solely on the last hidden state, attention computes a context vector as a weighted combination of all encoder representations, where the weights indicate the relevance of each input token to the current output token. This capability proved crucial in overcoming the information bottlenecks and dependency problems that characterize RNN-based architectures.

In mathematical terms, an attention module operates on three sets of vectors: Queries (Q), Keys (K), and Values (V). The mechanism computes compatibility scores between queries and keys, converts these scores into a probability distribution (usually via the softmax function), and uses the resulting weights to form a context vector as a weighted sum of the values. The general formulation can be expressed as:

$$Attention(Q, K, V) = softmax(f(Q, K))V$$

where $f(Q, K)$ is the compatibility function that determines the relevance of each key to the query.

There are several variants of attention that have been proposed over time, each differing in how these weights are computed. Additive (Bahdanau) attention [4] uses a small feed-forward neural network that jointly processes the query and key vectors to produce alignment scores. In contrast, Multiplicative (Luong) attention [5] measures similarity through the dot product between query and key vectors, sometimes with an additional learnable weight matrix, offering greater computational efficiency. The Scaled Dot-

Product attention [6] normalizes these dot-product scores by the square root of the key dimension $\sqrt{d_k}$ to stabilize gradients when the dimensionality is large. Formally:

$$Attention(Q, K, V) = softmax(QK^T/\sqrt{d_k})V$$

This computes a weighted sum of values, where the weights are determined by the similarity between query and key representations.

Soft attention computes continuous, differentiable weights across all input positions, while hard attention makes discrete selections, often requiring reinforcement-learning techniques for training. Location-based attention [7] explicitly incorporates positional information. Local attention focuses only on a restricted window of positions around the most relevant region, whereas global attention allows every query to attend to all input tokens simultaneously.

A particularly powerful development is self-attention [6] , where different positions within the same sequence attend to one another. This enables the model to capture relationships across all time steps in parallel rather than sequentially. Cross-attention, on the other hand, relates two different sequences, such as the encoder outputs and decoder states in sequence-to-sequence models. Multi-head attention extends these mechanisms by computing multiple attention distributions in parallel subspaces, allowing the model to capture patterns simultaneously.

The Transformer architecture [6] builds entirely on self-attention, eliminating recurrence. As Figure 5 illustrates, it consists of multiple encoder and decoder layers, each containing multi-head attention and feed-forward sublayers with residual connections and normalization. Transformers efficiently model long-range dependencies and have become the dominant framework for speech and language applications, including Whisper and XLS-R.

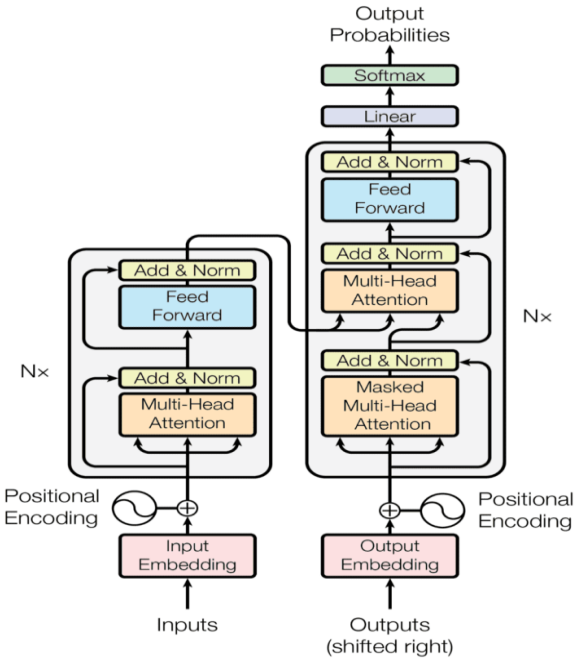

Figure 5: Transformer architecture consisting of stacked encoder and decoder layers, each with multihead attention and feed-forward layers. Adapted form [6].

2.1.7 Generalization and Overfitting

Generalization refers to a model's ability to perform well on unseen data. Underfitting occurs when a model lacks complexity, while overfitting arises when it memorizes training examples instead of learning general patterns. Common regularization strategies include adding a penalty to discourage large weights, Early Stopping which stops training once validation performance ceases to improve (Figure 6), Data Augmentation which increases diversity in the training data, reducing sensitivity to noise. These methods improve the robustness and generalization of deep networks, especially when training data is limited.

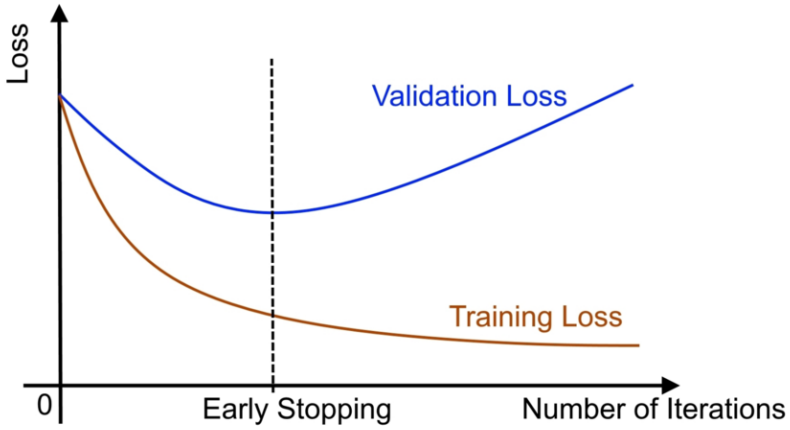

Figure 6: Early stopping strategy based on the relation between validation and training loss.

2.2 Automatic Speech Recognition (ASR)

Automatic Speech Recognition (ASR) is the process of converting an acoustic speech signal into a sequence of words (Figure 7). It lies at the intersection of signal processing, machine learning, and linguistics. Modern ASR systems form the foundation of many human-computer interaction technologies such as voice assistants, transcription tools, and subtitle generation. This section reviews the theoretical background of ASR including its probabilistic formulation, computational architectures, and evaluation metrics, providing the necessary foundation for understanding singing-voice recognition.

2.2.1 Speech Recognition Fundamentals

The goal of ASR is to find the word sequence Ŵ that maximizes the posterior probability given the observed acoustic signal X. Using Bayes' theorem, this is expressed as:

$$\hat{W} = \arg\max_W P(W|X) = \arg\max_W P(X|W)P(W)$$

where:

- P(X|W) is the *acoustic model* - the likelihood of observing acoustic features X given word sequence W.
- P(W) is the *language model* - the prior probability of the word sequence based on linguistic context.

This formulation separates acoustic and linguistic knowledge; The acoustic model links the signal to phonetic units, while the language model imposes grammatical plausibility. The decoder then searches for the most probable word sequence that maximizes their product. In practice, decoding combines the acoustic model likelihoods with the language model priors using a search algorithm such as the Viterbi decoder or beam search. The decoder explores alternative hypotheses in a lattice, scoring each path as a weighted sum of acoustic and linguistic log-probabilities. The search finds the most probable word sequence that maximizes the combined likelihood P(X|W)P(W) under the constraints of pronunciation lexicons and phonetic context.

Before modeling, the raw waveform is transformed into compact, perceptually motivated representations. Typical features include Mel-Frequency Cepstral Coefficients (MFCCs) or log-Mel spectrograms, extracted from short overlapping frames of about 20–25 ms using Fourier analysis. First and second order temporal derivatives capture dynamics, while Cepstral Mean and Variance Normalization mitigates channel variability. These normalized feature sequences serve as input to the acoustic model. ASR is fundamentally a sequence-to-sequence task, where both speech and text are time-dependent and of variable length, with no explicit frame-to-symbol alignment.

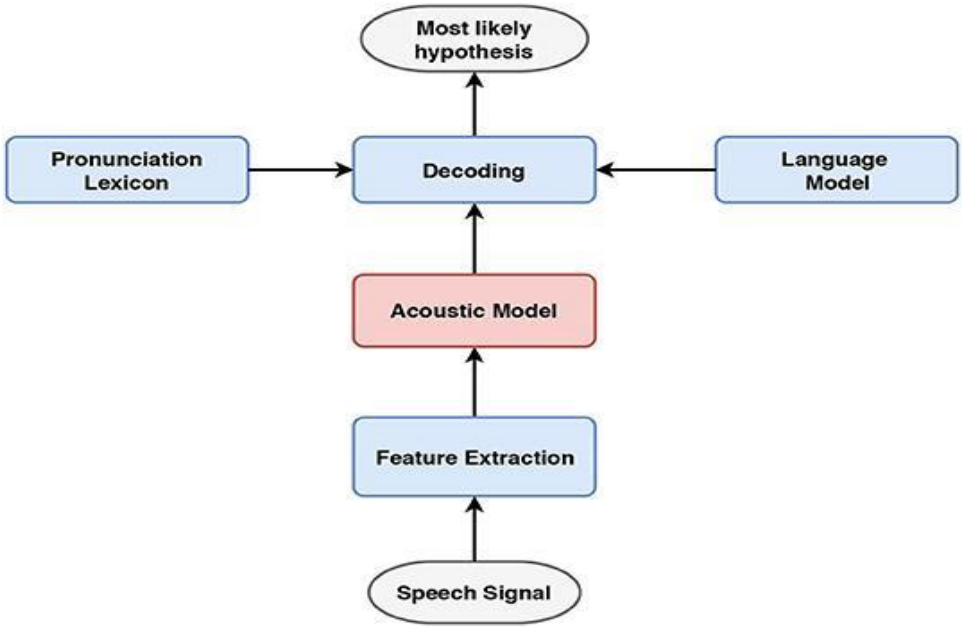

Figure 7:Simplified ASR pipeline

2.2.2 Conventional (Statistical) Approaches

In early ASR, temporal variability was modeled using Hidden Markov Models (HMMs). Each phoneme is represented by a sequence of hidden states, and each state emits acoustic features with certain probabilities.

An HMM is defined by:

- States $S = \{s_1, s_2, ..., s_N\}$
- Transition Probabilities $a_{ij} = P(s_j|s_i)$

- Observation Probabilities $b_j(x)$
- Initial state distribution π

The joint probability of an observation sequence X and a state sequence S is

$$P(X, S) = \pi_{s_1} \, b_{s_1}(x_1) \prod_{t=2}^{T} a_{s_{t-1} s_t} \, b_{s_t}(x_t)$$

The Markov assumption simplifies modeling by assuming each state depends only on the previous one. Training uses the Baum-Welch(forward-backward) algorithm, a special case of Expectation-Maximization that estimates transition and emission probabilities from unaligned data. Decoding employs the Viterbi algorithm, a dynamic-programming procedure that finds the most likely state path given the observed features. A single phoneme, for instance, may be represented by three sequential HMM states (onset, steady, offset), capturing its temporal evolution.

To model the continuous distribution of acoustic features, each HMM state uses a Gaussian Mixture Model (GMM):

$$b_j(x) = \sum_m c_{jm} \, \mathcal{N}(x; \mu_{jm}, \Sigma_{jm})$$

where c_{jm} are mixture weights, and $\mathcal{N}(x; \mu_{jm}, \Sigma_{jm})$ is a Gaussian component. Although effective, GMM-HMM systems relied on extensive feature engineering and large labeled datasets. The parameters $\mu_{jm}, \Sigma_{jm}, c_{jm}$ are estimated via Expectation-Maximization (EM), alternating between computing component responsibilities and maximizing expected likelihood.

The language model estimates the probability of a word sequence as

$$P(W) = \prod_{t=1}^{T} P(w_t \mid w_{t-1}, w_{t-2}, ..., w_{t-N+1})$$

where N-gram statistics are learned from text corpora. During decoding, the system searches for the word sequence maximizing the product $P(X|W)P(W)$. Early decoders applied the beam-search algorithm to maintain multiple candidate hypotheses and combined acoustic and language-model scores.

2.2.3 Transition to Neural Acoustic Models

The limitations of GMMs led to Deep Neural Networks (DNNs) replacing them as acoustic models [8]**.** In a DNN-HMM hybrid, a deep feed-forward network predicts posterior probabilities of HMM states:

$$P(s_t|x_t)$$

which are converted to likelihoods during decoding. Training typically minimizes cross-entropy loss between predicted and reference state labels. Input features are often stacked over a temporal context window to provide temporal context. RNNs and LSTM units [9] captured long-range temporal dependencies that HMMs could not model. These hybrid architectures achieved major accuracy gains and served as a bridge to fully end-to-end systems.

2.2.4 End-to-End ASR Architectures

A major shift occurred with end-to-end training, where the model directly maps acoustic features to character or word sequences without explicit state alignment. Connectionist Temporal Classification (CTC) [10] introduced a differentiable loss for unsegmented sequences, while attention-based encoder–decoder models [11] jointly learned alignment and transcription. CTC enables training without frame-level alignment by summing over all valid alignments between input and output. A special *blank* symbol accounts for variable phoneme durations. The model maximizes

$$P(Y|X) = \sum_{\pi \in \mathcal{B}^{-1}(Y)} \prod_{t=1}^{T} P(\pi_t|x_t)$$

where $\mathcal{B}^{-1}(Y)$ is the set of frame sequences that collapse to the output label sequence Y. CTC assumes monotonic alignment and works well for short or medium-length utterances.

The Listen, Attend and Spell model [11] introduced attention mechanisms that learn soft alignments between audio frames and output tokens:

$$c_t = \sum_i \alpha_{t,i}\, h_i$$

where h_i are encoder states and $\alpha_{t,i}$ are attention weights. This approach jointly learns alignment and language modeling, though it can struggle with very long sequences.

Modern systems use Transformer [12] and Conformer [13] architectures, which combine self-attention with convolution to model both global and local context. These frameworks unify the acoustic and language modeling stages within a single network, trained with hybrid CTC and attention objectives. RNN-T [14] extends CTC by including a prediction network for previous outputs and a joint network combining encoder and prediction states. It computes symbol probabilities conditioned on both acoustic and linguistic context, enabling online decoding.

2.2.5 Self-Supervised and Multilingual Learning

Recent advances in self-supervised learning have enabled pre-training on massive collections of unlabelled audio. wav2vec 2.0 [15] learns contextualized speech embeddings by predicting masked segments of the audio representation. XLS-R [16] scales this approach to hundreds of languages, demonstrating strong cross-lingual transfer for under-resourced domains.

At an even larger scale, Whisper [17] uses 680 000 hours of weakly supervised multilingual speech-text pairs to train a universal encoder-decoder model capable of transcription, translation, and speech detection. These foundation models learn language-agnostic acoustic features that can be efficiently fine-tuned or adapted to new domains such as singing voice. They represent a shift from task-specific training toward universal speech encoders that require minimal labeled data.

2.2.6 Evaluation Metrics and Benchmark Corpora

ASR performance is commonly measured by the WER and Character Error Rate (CER), defined as the normalized edit distance between the predicted and reference transcriptions [18]:

$$WER = \frac{S + D + I}{N}$$

Widely used benchmarks include LibriSpeech [19], consisting of 1 000 hours of English audiobook recordings, and Common Voice [20], a multilingual dataset that supports research on diverse accents and speaking styles. These corpora provide controlled, speech-oriented data that underpins most ASR evaluation. However, they lack the prosodic and expressive variability found in music.

3. Related Work

3.1 ALT

Ongoing progress in speech recognition has provided a solid basis for modeling the relationship between sound and language. However, extending these systems to singing voice introduces a new challenge that ASR frameworks have to handle. Automatic Lyrics Transcription (ALT) refers to the process of automatically converting sung vocals in music recordings into textual lyrics. It lies at the intersection of Automatic Speech Recognition (ASR) and Music Information Retrieval (MIR). ASR for singing remains a distinct and highly challenging subfield of speech technology. [21] provides one of the most comprehensive analyses of why recognition of sung vocals diverges fundamentally from spoken language processing. Singing introduces extreme pitch variability, melismatic vowel extension, rhythmic irregularity, and expressive pronunciation, while being further complicated by background accompaniment that masks vocal harmonics and phoneme boundaries. These properties mess with the acoustic and temporal assumptions of ASR models trained on clean, conversational speech. Early surveys and application overviews also emphasized how these properties complicate phonetic modeling and alignment compared with spoken language [22], [23], [24].

The first generation of ALT systems adapted classical ASR pipelines to solo singing. HMM-based recognizers with MFCC features decoded phoneme sequences from monophonic vocals, often after rudimentary vocal enhancement [25]. However, the presence of accompaniment in commercial mixes introduced strong interference that degraded recognition. Even with better front-ends, traditional systems struggled to generalize across genres and vocal styles [26]. As deep learning matured, neural architectures supplanted hand-engineered features and became the default in ALT (see §2.2.4). In parallel, community datasets, including DAMP-Sing! karaoke vocals and Jamendo, established reproducible benchmarks for singing voice research [27].More

recently, large multilingual, weakly supervised speech models (Whisper, XLS-R) provided strong zero-shot baselines and compelling starting points for lyric-domain adaptation [16], [17], [28].

3.1.1 Classical and Early Machine-Learning Approaches

Early research in lyrics transcription primarily relied on adapting speech-recognition architectures to singing. These systems employed HMM-GMM frameworks trained on speech corpora and adapted to the singing voice through simple duration modeling and speaker adaptation [25]. The introduction of monophonic karaoke datasets, such as those developed by [29] using the DAMP Sing! corpus, provided the first reproducible benchmarks for lyrics recognition. Their baseline system, implemented in Kaldi with TDNN-F acoustic models, achieved a 19.6% WER, demonstrating that even without accompaniment, singing presents challenges unseen in conversational speech.

Classical systems typically performed poorly under polyphonic conditions, as they lacked mechanisms to disentangle vocal energy from accompaniment. Preprocessing steps such as melody extraction or vocal source separation were therefore introduced to enhance vocal clarity prior to recognition. Despite these improvements, the dependence on hand-engineered features and language models limited scalability and cross-genre generalization, paving the way for deep-learning-based methods.

3.1.2. Deep Learning and End-to-End ALT Systems

The next generation of ALT systems adopted end-to-end neural architectures, unifying acoustic, alignment, and language modeling into a single trainable framework. Notably, Gao, Gupta, & Li [30] proposed a multi-task learning approach, where the computational model simultaneously predicts lyrics and chord progressions. By sharing encoder representations between lyric and chord tasks, the model captures harmonic context that guides lyrical decoding. Using a Transformer encoder-decoder trained with a hybrid CTC and attention objective, it achieved state-of-the-art word-level accuracy on the DALI and Jamendo datasets, demonstrating that harmonic information can serve as an effective structural prior for transcription. More recently, unified frameworks jointly perform transcription and alignment, further reducing reliance on external aligners and hand-crafted priors [31].

3.1.3 Self- and Semi-Supervised Learning for ALT

Given the limitation of annotated singing corpora, ALT has adopted semi-supervised learning and self-training. The Self-Transcriber framework [32] introduces a “Noisy Student” strategy, in which a teacher model generates pseudo-labels for unlabeled singing audio, and a student model retrains on these pseudo labels with heavy data augmentation. This approach achieved strong performance improvements with only a few hours of labeled data. A related technique proposed by Pham et al [33], refines pseudo-label generation through meta learning, by optimizing the teacher model to produce labels that best improve student generalization. Such optimization provides theoretical grounding for iterative pseudolabeling in ALT pipelines. Both frameworks contribute to efficient training paradigms that are vital for low resource musical genres and languages.

Large pretrained language models have further advanced ALT. Radford et al. [17] demonstrated remarkable zero-shot transcription ability across dozens of languages by

training on 680,000 hours of weakly labeled speech–text pairs. Building on Whisper, Zhuo et al. [34] combined Whisper transcriptions with GPT-based post-processing to improve semantic and syntactic coherence in multilingual lyrics. Similarly, Song et al. [35] introduced parameter-efficient fine-tuning for multilingual ASR by attaching language-specific LoRA adapters to the Whisper model, mitigating interference between languages and enabling efficient language expansion. Beyond Whisper-style pretraining, SpeechLM [36] enhanced speech pretraining with unpaired textual data, unifying speech and text into the same discrete semantic space with a unified Transformer network, and achieving stronger low-resource ASR performance through tighter integration of acoustic and linguistic representations.

3.1.4 Foundation and Cross-Lingual Speech Models Relevant to ALT

The ASR models discussed in chapter 2.2.5 -namely wav2vec 2.0, XLS-S and Whisper- have become essential for ALT in recent years. Their ability to learn universal acoustic representations that generalize across languages and domains, and to transfer knowledge from speech corpora allows them to perform effectively in diverse settings, such as singing voice recognition. In practice, researchers conduct pretraining, fine-tuning and prompt-based adaptations on them to enable accurate lyric transcription.

3.2 Lyrics Alignment

Lyrics-to-audio alignment aims to temporally synchronize the words or phonemes of lyrics with the corresponding regions in a song's audio waveform. This process is complementary to ALT but places its emphasis on temporal precision rather than lexical accuracy. Accurate alignment is essential for applications such as karaoke displays, automatic subtitling, singing voice analysis and data preparation for supervised ALT training.

3.2.1 Early Forced-Alignment and HMM-Based Models

The earliest frameworks adapted speech forced-alignment systems to the musical domain by employing HMMs and manually designed acoustic features. An example is Mesaros & Virtanen [23], who introduced a two-stage pipeline combining melody extraction and sinusoidal modeling to isolate vocal segments before applying a phoneme-level HMM recognizer tailored to singing. Their alignment grammar allowed optional pauses and instrumental gaps between lines, resulting in a median line-boundary error of 0.64 seconds on a corpus of 17 songs.

Subsequent studies refined this paradigm by incorporating additional musical cues. Fujihara et al. [37] proposed synchronizing lyrics and CD recordings through Viterbi alignment of separated vocal tracks, while Mauch et al. [26] integrated chord-progression information into the HMM to constrain temporal transitions, improving robustness to instrumental overlaps.

3.2.2 Semi-Supervised and Genre-Aware Alignment

Researchers explored semi-supervised and genre-conditioned strategies. Gupta et al. [38] introduced a segmentation-based semi-supervised alignment method in which a cappella performances are divided into short, high-energy segments, transcribed by an ASR model, and matched to written lyrics using Levenshtein distance to identify reliable

anchor segments. These anchors are then used to adapt the acoustic model iteratively, achieving substantial WER reduction and ~73% human-verified correctness.

A complementary line of research investigates genre-informed acoustic modeling, motivated by the observation that vocal clarity, spectral density, and rhythmic regularity vary widely across musical styles. For instance, Gupta et al. [39] extended conventional ASR frameworks with genre-specific phone and silence models, training both on polyphonic mixtures rather than isolated vocals. Their experiments showed that polyphonic training and genre-aware silence modeling significantly reduced alignment errors and word error rate compared to genre-agnostic baselines, achieving state-of-the-art performance across multiple public song datasets.

3.2.3 CTC-Based Forced Alignment & Segmentation

Bridging traditional HMM pipelines and recent cross-modal methods, a widely adopted paradigm for precise lyrics-to-audio alignment builds upon Connectionist Temporal Classification (CTC) models. Unlike HMM-based approaches that rely on explicit state transitions, CTC provides a differentiable objective for modeling monotonic alignments between an acoustic sequence and a known transcription. During alignment, a pretrained CTC acoustic model outputs frame-level posteriors over tokens and a special blank symbol; the optimal path is then obtained via Viterbi decoding programming, constrained to match the given lyrics in order. This creates timestamped boundaries for words or phonemes and supports accurate segmentation of long recordings without additional supervision.

A widely adopted paradigm for precise lyrics-to-audio alignment builds upon Connectionist Temporal Classification (CTC) models. Unlike HMM-based approaches that rely on explicit state transitions, CTC provides a differentiable objective for modeling monotonic alignments between an acoustic sequence and a known transcription. During alignment, a pretrained CTC acoustic model outputs frame-level posteriors over tokens and a special blank symbol; the optimal path is then obtained via Viterbi decoding programming, constrained to match the given lyrics in order. This creates timestamped boundaries for words or phonemes and supports accurate segmentation of long recordings without additional supervision. The approach was formalized by CTC-Segmentation [40], which demonstrated that CTC posteriors can reliably derive segment boundaries at scale for end-to-end ASR corpora. Subsequent toolkits and implementations, such as the NeMo Forced Aligner [41] and TorchAudio's forced alignment utilities, have operationalized these ideas for multilingual and word-level alignment. More recent work [42] refines alignment accuracy by introducing label priors to mitigate "peaky" CTC posteriors and improve onset and offset estimates.

3.2.3 Neural and Cross-Modal Alignment

Recent advances leverage neural sequence modeling and cross-modal contrastive learning to align audio and text directly. Demirel et al. [39] proposed a pipeline to identify reliable anchor words with a biased ASR/LM, segment the performance around these anchors, and then perform a second-pass, CTC-based alignment on each segment. Building on this direction, Durand, Stoller, & Ewert [44] proposed a fully contrastive audio-text embedding framework, where an audio encoder maps short spectrogram patches to latent vectors, while a text encoder embeds lyric tokens. During training, it maximizes similarity for matched pairs and minimizes for mismatches.

Parallel efforts address real-time alignment for karaoke and live performance. Park et al. [45] proposed a lyrics-to-audio alignment system for karaoke and live performance by combining phonetic and musical cues. Instead of relying solely on low level acoustic features such as mel-spectrograms, they extracted Phonetic Posterior Gram (PPG) embeddings -from a pre-trained ASR acoustic model used purely as a feature extractor- to represent what phoneme is being sung at each frame as a vector of phoneme posteriors. These PPGs are independent to speaker and pitch, making them more robust to the melodic variability in singing. PPGs are then combined with chroma features, which summarize the underlying harmonic progression, resulting in a joint representation that captures both linguistic content and musical structure. This design enables real-time alignment of the incoming audio to the target lyrics.

3.3 Low-Resource and Greek-Specific Research

Automatic lyrics transcription in low-resource languages remains an under-explored yet critical research frontier. The challenge extends beyond the scarcity of labeled data to encompass dialectal variation, orthographic complexity, and domain mismatch between spoken and sung speech. Consequently, researchers have focused on transfer learning, self-supervised pre-training, and domain adaptation techniques that can leverage high-resource data while efficiently adapting to new acoustic and linguistic domains [15], [17], [28].

3.3.1 Low-Resource ASR and Domain Adaptation

Building on the foundation models described in Section 2.2.4, domain adaptation techniques allow effective fine-tuning when data are limited. The M2DS2 framework [46] combines self-supervised fine-tuning and pseudo-labelling on in-domain audio, improving recognition accuracy for Modern Greek. Similar principles have been explored by Damianos et al. [47], who introduce MSDA, a two-stage unsupervised domain adaptation pipeline that combines self-supervised pre-training with pseudo-label-based teacher-student training to reduce domain mismatch in ASR, showing strong gains in low-resource settings such as Greek.

Complementary strategies include parameter-efficient transfer learning which further reduces computational demands, and multistage or staged fine-tuning for extremely small corpora. For instance, Pillai et al. [48] proposed a multistage strategy based on Whisper, where they first fine-tune on a higher-resource, linguistically related language (Tamil), and then adapt to the low-resource target, resulting in lower WER than direct adaptation.

Another promising strategy is curriculum pretraining. Rather than training models uniformly across tasks of varying difficulty, one gradually increases task complexity, such as from pure transcription to semantic or translation tasks. For instance, Bansal et al. [49] show that pretraining an encoder on a high-resource ASR task, then fine-tuning on a low-resource Speech Translation task, results in substantial gains, especially via transferred acoustic representations.

3.3.2 Greek Speech and Music Resources

The Greek Audio Dataset (GAD) [21] was among the first publicly described corpora linking Greek music audio with metadata and lyrics, later expanded through the Greek Music Dataset [51] and subsequent work on metadata-driven music retrieval. In the

broader speech domain, recent large-scale efforts have strengthened Greek ASR infrastructure. The Greek Podcast Corpus [52] aggregates over 800 hours of spontaneous, conversational speech, while the Greek Dialect Benchmark [53] assesses recognition robustness across regional dialects.

Together, these resources expand acoustic and linguistic coverage across genres, registers, and dialects, providing a foundation for modern Greek ASR models. However, despite these advances, there remains no publicly available Greek lyrics transcription pipeline. Existing Greek ASR models, such as wav2vec 2.0 XLS-R (Greek) achieve high accuracy on spoken language but struggle with the prosodic, melodic, and expressive variability of singing. Consequently, the development of a specialized Greek ALT framework remains an open research need.

3.3.3 Toward a Greek ALT Pipeline

A practical Greek ALT system can be conceptualized as a multi-stage pipeline: i) vocal isolation using modern source-separation tools, ii) acoustic modeling via pre-trained multilingual encoders fine-tuned on Greek singing using parameter-efficient adapters, iii) temporal synchronization using contrastive or CTC-based alignment, iv) and weak supervision through pseudo-label generation and iterative re-training. This design merges state-of-the-art techniques from multilingual ASR, self-supervised learning, and alignment, tailored to the Greek musical context.

III. PRACTICAL PART: EXPERIMENTS

4. Methodology

4.1 Overview

The methodology adopted in this thesis is designed to build a robust pipeline for ALT. All computational experiments were conducted on two systems, the ILSP Borges Cluster and EUroHPC Leonardo Supercomputer [54]. Both environments provided multi-GPU nodes, CUDA-enabled Pytorch installations and shared storage systems. The overall workflow is divided into two major stages: i) Dataset Preparation and ii) Lyrics Transcription. Each stage addresses specific challenges posed by singing voice data, from removing instrumental interference to creating clean, segmented datasets, and finally adapting state-of-the-art ASR models to the domain of singing.

The Dataset Preparation stage's pipeline as illustrated in figure 8, constructs a complete training corpus from raw Greek music data. It begins with source separation using Demucs_ft to extract clean vocal stems from polyphonic recordings, followed by Kaldi-style segmentation and transformation. Subsequently, a CTC-based aligner generates time-synchronized lyric segments, which are then enriched with English translations through GPT-4o-mini to create a multitask aligned dataset. The processed data is converted into a Hugging Face-compatible dataset for downstream model training.

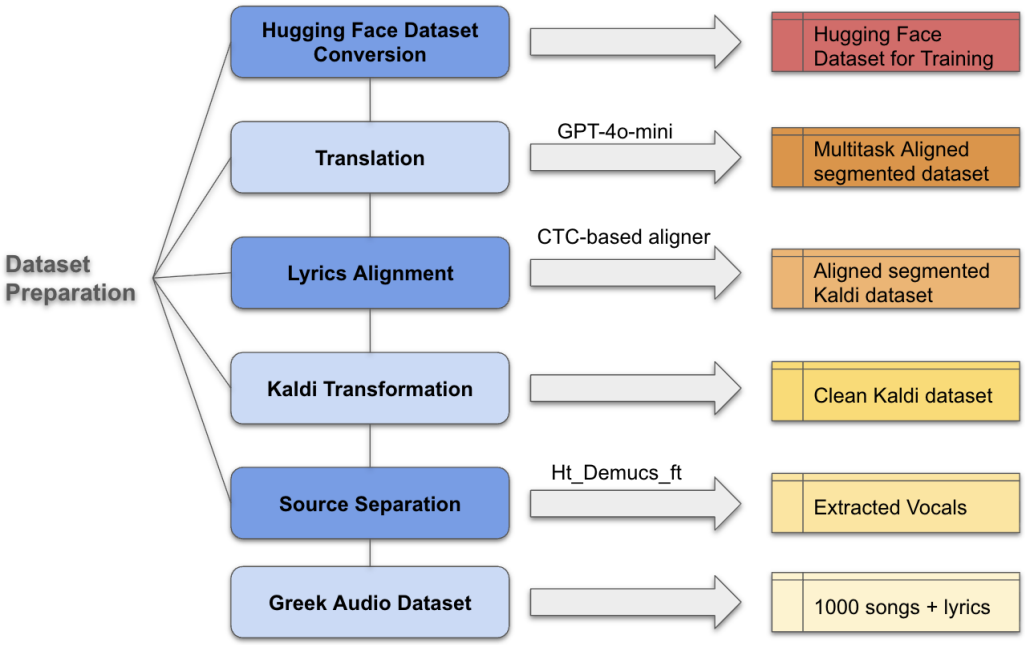

Figure 8 : Methodology Overview Pipeline: Dataset Preparation

In the Lyrics Transcription stage (Figure 9), Whisper models are fine-tuned for both transcription and translation under three configurations: i) transcription-only, ii) multitask, and iii) two-stage adaptation. The performance of each configuration is assessed using WER, complemented by error analysis and a qualitative discussion of system outputs.

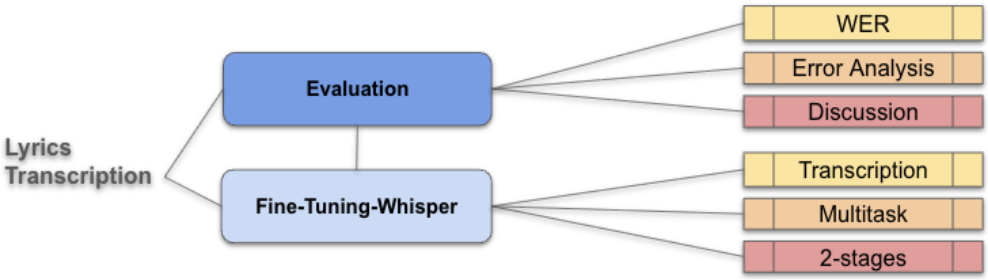

Figure 9: Methodology Overview Pipeline: Training and Evaluation

4.2 Dataset

The primary resource for this study is the Greek Audio Dataset (GAD) [50], which constitutes the first standardized collection of Greek music data for MIR research. The dataset contains 1,000 popular Greek songs covering a broad spectrum of genres, including Urban and Popular folk, Rock, Hip Hop/R&B, Pop. In addition to genre annotations, each track is also accompanied by lyrics, manually annotated mood labels following the Thayer model, and metadata such as artist, title, and genre. A distinctive feature of GAD is that it does not directly provide copyrighted audio content, but it offers extracted audio features (timbral, rhythmic, and pitch-related) and links to the

corresponding songs on YouTube, thus enabling researchers to obtain the raw audio when required. The dataset therefore provides both a rich linguistic resource, through its lyrics, and a musically diverse collection of audio features and metadata, making it particularly suitable for research on ALT and related MIR tasks.

4.3 Source Separation with Demucs

Since most publicly available music corpora are polyphonic, it is necessary to isolate the singing voice from instrumental accompaniment before training and evaluation. To facilitate large-scale processing, the raw audio files were first organized into batches of 100 WAVs each. This step ensured efficient scheduling and minimized file handling overhead during source separation, while also making the pipeline more resilient to interruptions.

In this work, Demucs is employed as a source separation tool, using the htdemucs_ft configuration [55]. This configuration corresponds to the Hybrid Transformer Demucs model, which has been additionally fine-tuned on curated music separation data. Compared to the base model, htdemucs_ft provides improved separation quality, particularly for vocals, which makes it especially suitable for lyrics transcription tasks.

To ensure reproducibility, separation was carried out with deterministic settings (--shifts 0). Two-stem separation was chosen in order to output only the vocal and accompaniment tracks, thereby reducing storage requirements while maintaining high-quality vocal stems. After separation, the vocal tracks were downmixed to mono and resampled to 16 kHz, matching the input assumptions of Whisper and lowering computational cost without compromising intelligibility. Normalization was applied to maintain consistent loudness levels across files. This pipeline design ensures that the training and evaluation corpora contain vocal content with minimal residual accompaniment. Performing lyric alignment on the separated stems rather than on the polyphonic mixture avoids timing mismatches and improves the reliability of both alignment and transcription.

4.4 Kaldi Data Conversion

Before running the forced alignment pipeline, the corpus is converted into Kaldi-data format. Starting from a curated list of recording identifiers, each one is mapped to its corresponding WAV file and lyric line, producing the standard Kaldi files (wav.scp, text, utt2spk, and spk2utt). Lyric files are normalized and flattened into sentence-like lyric lines. Since the experiments in this phase did not model multiple speakers, a constant speaker label (singer1) is assigned to all utterances in utt2spk, and the reciprocal spk2utt mapping was generated automatically. This Kaldi representation of the corpus serves as the single source of truth for subsequent steps.

4.5 Lyrics Alignment and Segmentation

Following source separation and transformation into kaldi form dataset, the vocal stems are aligned to their corresponding lyrics to obtain time-stamped supervision suitable for fine-tuning. Alignment is performed with a CTC-based forced aligner implemented in a custom pipeline built on the Wav2Vec2 architecture. The system processes each recording in windowed audio chunks of 30 s with 2 s context overlap, producing word-level timestamps that indicate when each lyric fragment is sung. For languages unsupported by the alignment model’s vocabulary, the pipeline can enable romanization.

To quantify alignment reliability, each utterance is scored using a composite alignment quality score that combines the percentage of aligned words, the average CTC log-probability over aligned tokens, and a duration regularity penalty that down-weights segments containing abnormally long words. The output of the pipeline is a segmented Kaldi-style dataset comprising:

- segments with lines of the form <utt_seg_id> <utt_id>
- text mapping each <utt_seg_id> to its aligned lyric fragment
- wav.scp mapping each <utt_id> to the absolute path of the separated vocal WAV
- utt2spk propagating speaker identifiers
- utt2score with per-utterance alignment confidence.

4.6 Translation

To enable parallel transcription-translation experiments, we augment the segmented Kaldi corpus with English translations. For each Greek segment, the corresponding English text is generated using OpenAI GPT-4o-mini) Each Greek lyric segment is translated individually, and the resulting English texts are paired with their corresponding segment identifiers, forming a bilingual Kaldi dataset. This setup supports both Greek transcription and speech-to-English translation within a unified framework. By aligning translations at the same segment level as the forced-aligned audio, the dataset maintains synchronization between audio, Greek text, and English text, minimizing alignment drift and ensuring consistency for multitask Whisper fine-tuning.

4.7 Kaldi to Hugging Face Dataset Conversion

In the last preprocessing step, the segmented Kaldi corpus is converted into a Hugging Face dataset, suitable for model training and evaluation. The converter receives the Kaldi files and builds a manifest keyed by segment identifiers. For each entry, the corresponding portion of the vocal stem is extracted based on its aligned start and end times, downmixed to mono, and resampled to 16 kHz to match Whisper's input specifications. When alignment confidence scores are available, a filtering step removes low-quality segments to maintain dataset reliability. The resulting dataset contains one record per aligned lyric segment, including i) the audio array and sampling rate, ii) the normalized Greek transcription, and iii) when applicable, the English translation.

4.8 Lyrics Transcription with Whisper

As a baseline system, the OpenAI Whisper model is employed for both transcription (Greek lyrics) and translation (Greek to English). Whisper is particularly suitable for this task due to its multilingual coverage and robustness to domain shifts. Two modes of operation are investigated. First, zero-shot inference is carried out using the original pretrained models, in particular Whisper-small and Whisper-medium. Second, the models are fine-tuned on singing voice data, using the aligned datasets prepared in the previous stage. In this way, the experiments examine both the performance of foundation models and the gains that can be achieved through targeted domain adaptation.

4.9 Fine-Tuning Whisper for Lyrics Transcription and Translation

Model adaptation is conducted by fine-tuning the Whisper family on the aligned, segment-level Hugging Face dataset described earlier. Since the goal of this stage was to adapt the pretrained speech recognition model to the domain of Greek singing voice, while simultaneously investigating how task composition (transcription versus multitask training) and model scale influence recognition accuracy, multiple model scales such as Small, Medium and Large Whisper checkpoints and multiple training configurations are examined. The idea was that multitask training can improve transcription by jointly regularizing the encoder-decoder; ASR sharpens acoustic-phonetic representations while translation supplies a clean language objective that enhances normalization and long-range consistency, reducing overfitting, hallucinations, and repetition, thereby lowering WER in practice.

Firstly, with the transcription-only configuration the model learns to map Greek audio segments directly to Greek text. Then, fine-tuning is conducted with a combination of both transcription (Greek audio to Greek text) and speech-to-text translation (Greek audio to English text) tasks in a unified training loop. For multitask experiments, the same dataset serves as the source for both tasks. The Greek transcriptions and their corresponding English translations (generated in Section 4.6) are paired by segment identifiers, allowing a one-to-one alignment between input audio and both output languages.

To alternate systematically between the two tasks, a custom sampler is implemented. This sampler interleaves transcription and translation batches in a fixed pattern, ensuring that each batch contains examples from only one task. The sampler receives two subsets, one for transcription examples and one for translation examples, and creates indices in a deterministic sequence such that a fixed number of transcription batches are followed by translation batches. For transcription-only experiments, the sampler is disabled, and the model is trained purely on the Greek subset.

Input features are extracted at 16 kHz mono using the Whisper feature extractor. For each experiment, two WhisperProcessor instances were initialized. One with task="transcribe" and language="el", and one with task="translate" and language="en". A multilingual collator is designed to dynamically select the transcription processor for Greek and the translation processor for English. The collator ensured that each batch is internally consistent in both task and language, and automatically tokenized the corresponding text or translation fields. Each example therefore contains the waveform array, sampling rate, and either a Greek or English text target.

The entire pipeline is implemented in Python using the Transformers and Datasets libraries, ensuring full compatibility with Hugging Face interfaces. Training uses the Seq2SeqTrainer class from the Transformers library, configured with the AdamW optimizer, a learning rate of 5×10^{-5}, and mini-batches of 4-8 segments per GPU. Mixed-precision training (FP16) is enabled automatically on CUDA devices to accelerate computation and reduce memory usage. Each run consists of five epochs, with checkpoints saved once per epoch and all intermediate results are stored in structured experiment directories.

4.10. Staged fine tuning

A two-stage fine-tuning is also adopted to reduce the domain gap between speech and singing. Stage-1 is the fine-tuning of Whisper-Medium on Greek speech (Common Voice

EL) using a transcription objective with fixed language (“el”) and consistent decoding settings. This stage adapts the acoustic encoder to Greek phonotactics, prosody, and orthography under cleaner speech conditions. In Stage-2, the training is continued at the same checkpoints on the Greek singing corpus, using the same tokenizer and normalization and the same evaluation protocol. Depending on the experiment, the multitask mixture is either kept or switched to transcribe-only to specialize on ALT. A smaller learning rate is employed in Stage-2, gradient accumulation to match effective batch size across stages, and early stopping on validation normalized WER computed with identical text normalization. Decoding (beam size, language setting, task flag) is held constant across stages to ensure comparability.

4.11 Testing & Evaluation Protocol

Model evaluation is performed with a dedicated inference script that generates transcriptions and translations from the held-out for testing HF segment-level dataset and computes normalized WER. The evaluator first loads the on-disk Hugging Face dataset and, when necessary, flattens DatasetDict splits into a single iterable dataset. It supports record-level filtering based on segment identifiers. Before scoring all references and hypotheses undergo normalization that produces a “normalized WER” which measures lexical accuracy.

At inference time, the script loads the Whisper weights from the fine-tuned checkpoint. Decoding is performed on GPU and uses a generation configuration initialized from openai/whisper-large-v2 for modern token maps. For each batch of segments, the evaluator constructs input features with the processor’s feature extractor and generation is performed with task-specific and language arguments. The evaluator runs two passes per recording, one for transcription and one for translation. WER is computed using the jiwer library over all normalized hypotheses and references aggregated by task.

The evaluation pipeline produces two main outputs. The first is a TSV file containing one row per segment, including the recording ID, segment ID, task type (transcribe or translate), reference text, and model hypothesis. The second is a JSON summary reporting normalized WER per task, along with metadata such as model and dataset paths, task type, language, timestamp, and computational host.

5: Experiments & Results

5.1 Zero-Shot Baseline (Before Fine-Tuning)

As a first experiment, Openai/whisper-medium is evaluated in zero-shot mode (no singing-domain adaptation) on the same held-out test set. The decoding is performed in both tasks (Greek transcription and English translation) with language fixed to Greek to mirror later conditions.

The model’s performance in WER is 92,1% for transcription and 83,5% for translation. These values indicate severe weakness on singing acoustics compared to speech-domain expectations, and they set a clear lower bound for subsequent adaptation. Manual inspection reveals systematic failures, characteristic of singing. Very common are the severe phonotactic drift and cluster corruption, hallucinations and semantic drift, orthographic instability and substitutions that distort lexical identity.

Overall, the baseline transcriptions are frequently unintelligible, with bursts of grammatical Greek interleaved with nonsense. This is a clear sign that the acoustic and prosodic characteristics from speech do not transfer to singing without adaptation. Zero-shot Whisper relies on speech-domain priors such as stable pitch trajectories, conversational timing, and low background interference. But as mentioned before, singing violates these assumptions. The encoder's acoustic invariances and the decoder's language priors therefore mis-align, producing the observed cluster errors, deletions, and hallucinations.

5.2 Results Overview on Fine-Tuned models

5.2.1 Major Improvements

After fine-tuning on Greek singing data, best transcription performance is achieved by Whisper Large (transcribe-only) with normalized WER of 30%. Relative to the zero-shot Whisper-Medium baseline, this is an absolute improvement of ≈ 67.4%. This magnitude of gain substantiates the claim that domain adaptation is indispensable for ALT in Greek songs.

For translation, the zero-shot baseline is actually better than the fine-tuned translation scores. The fine-tuning data and objective emphasize on Greek transcription, not cross-lingual generation, so adaptation strengthens the Greek bias and destabilizes language identification in lines that are actually English. In short, transcription benefits, while translation does not, given our training mix. Of course, the goal of this thesis is exploring ways to improve lyrical transcription and for this reason only transcription results will be discussed thoroughly.

Table 1: Baseline vs. Fine-Tuned best performance

Model	Task	WER (%)
Zero-shot Whisper-Medium	Transcribe	92,1
Fine-tuned Whisper-Large, Transcribe-only	Transcribe	30

5.2.2 Quantitative Results Overview

Table 1 presents the normalized WER values obtained across all Whisper model sizes and experimental configurations. The evaluation considered the three fine-tuned model capacities: Small, Medium, and Large, and the corresponding fine-tuning schemes:

i. Model fine-tuned with 1:1 mixed multitask in translation and transcription

ii. Model fine-tuned with 2:1 mixed multitask in translation and transcription

iii. Model fine-tuned with 4:1 mixed multitask in translation and transcription

iv. Model fine-tuned with Transcription only

Table 2: Normalized WER in transcription across all fine-tuned Whisper models

Model	Training Setup	WER (%)
Whisper Small	1:1 Mixed	51,3
Whisper Small	2:1 Mixed	**33,6**
Whisper Small	Transcribe-only	36,7
Whisper Small	4:1 Mixed	34,9
Whisper Medium	2:1 Mixed	32,3
Whisper Medium	Transcribe-only	**30,3**
Whisper Medium	4:1 Mixed	31,6
Whisper Large	2:1 Mixed	32,3
Whisper Large	Transcribe-only	**30,0**
Whisper Large	4:1 Mixed	31,2

5.2.3 Performance Trends by Model Size

i. Whisper Small

For the smallest variant, the 2:1 mixed configuration achieved the best transcription performance (33,6%), improving upon the transcription-only setup (36,7%). This suggests that a moderate amount of auxiliary translation data provides helpful regularization, nudging the encoder to learn slightly richer, more general representations than purely monolingual fine-tuning. When increasing the transcription proportion by 4:1 (less translation), performance slightly worsens relative to 2:1 (34,9%) but still outperforms transcription-only, indicating that the auxiliary signal remains useful even in small doses. Given the small model's limited capacity (244M parameters), the pattern is consistent with multitask learning theory. Low-capacity models benefit from a balanced auxiliary objective up to a point. Too little auxiliary data weakens the regularization benefits while removing it entirely is still a worse strategy.

ii. Whisper Medium

For the medium-sized model (769M parameters), the trends become more nuanced. The transcribe-only configuration achieved the lowest WER score of 30,3%, outperforming both 2:1 (32,3%) and 4:1 (31,6%) setup. This reversal to the small model suggests that higher-capacity encoders already internalize multilingual and multitask priors from pretraining, and therefore benefit more from focused adaptation rather than additional task mixing. The improvements here can be attributed to the model's ability to reallocate representational power to capture the specific characteristics of the singing domain without interference from translation signals. In other words, linguistic focus outperforms multitask regularization once the model's scale is sufficient.

iii. Whisper Large

The Whisper Large model (1.55B parameters) followed the same pattern but with even greater consistency. The transcribe-only variant achieved the lowest overall WER across all experiments (30%), slightly improving upon the 2:1 (32,3%) and 4:1 (31,2%) configuration. This confirms that as the model size increases, the benefits of multitask training diminish. Instead, targeted fine-tuning on domain-specific transcriptions becomes more effective for adaptation. The stability of results across the 2:1 and 4:1 configuration also suggests that large models are more resilient to task imbalance. They maintain performance regardless of the exact task ratio.

5.2.4 Results Overview on 2-stages fine-tuned model

Table 3: Performance of 2-stages Models

Model	Training Setup	WER (%)
Whisper Small	2-Stages 2:1	42,3
Whisper Small	2-Stages Transcribe-only	36,6
Whisper Medium	2-Stages Transcribe-only	33,1

Table 3 summarizes the results of the second stage of fine-tuning, where Whisper models previously adapted on Greek speech were further specialized on singing voice. Among the tested configurations, the Whisper-Medium transcription-only model achieved the best normalized Word Error Rate (33%), confirming that larger model capacity and single-task adaptation are most effective for lyric transcription. In contrast, the Small multitask model, trained jointly on transcription and translation, produced a substantially higher WER of 45%, the highest of all the models.

When compared to single-stage fine-tuning, the two-stage adaptation scheme resulted in only marginal or negative effects on performance. The Whisper Small transcribe-only model showed a slight improvement, suggesting that pre-adaptation on Greek speech

may offer minimal stabilization but no meaningful accuracy gain. In contrast, the Whisper Small 2:1 multitask configuration degraded sharply to 42.3 % WER, indicating that task mixing after speech adaptation disrupted convergence rather than enhancing generalization.

A likely explanation for this outcome lies in both the limited size of the Greek Common Voice corpus and the characteristics of the Whisper pretraining. Since Whisper already encodes strong multilingual and cross-domain representations, additional fine-tuning on a small, clean speech corpus provides minimal to none new acoustic knowledge and may even narrow the model's distributional coverage, reducing its ability to handle the wider variability of singing. Overall, these findings suggest that two-stage adaptation does not offer much for models like Whisper that already exhibit strong cross-lingual transfer.

5.2.5 Qualitive Results Overview: Error-Type Shifts

Across fine-tuned systems, errors cluster into a small set of stable categories:

(i) consonant-cluster corruption producing long invented spans in rhythmically dense lines

(ii) vowel/orthographic confusions driven by melisma

(iii) function-word deletions/insertions altering clitics and particles

(iv) semantic substitutions for phonetic neighbors

(v) hallucinations of culturally salient lyrics

(vi) boundary drift, where melismas shift word boundaries

(vii) morphology drift, preserving lemmas but altering inflection

(viii) named or lexicalized phrase distortions.

(i) More analytically, the dominant source of errors remains consonant-cluster corruption in rhythmically dense lines, where music and the way sounds blend together make the consonants hard to hear clearly. So, the model guesses and turns tight consonant groups into longer, made-up chunks. An example case is *«κοντερ πατας γκαζωνεις»* decoded as *«κοντεριακος ατρειατα πατασιαζουν της»*, where the /ndr–gk–z/ complex detonates into multiple syllables that preserve local phonetics but obliterate lexical identity. Another example is *«*κανε τα μουτρα μας ν' αστραφτουνε*»* vs *«*κανε τα ουτρα μας τα φτυρια*».* These patterns illustrate the model's tendency to resolve masked consonants by interpolating high-probability phoneme n-grams, trading accuracy for rhythmic fluency.

(ii) A second, persistent phenomenon is vowel and orthographic confusion, especially among ι/η/ει/υ and ο/ω pairs or between homophones created by melisma and accent suppression. For example, *«σειρήνες»* becoming *«συρίνες»*, *«ερεβος»* drifts to *«εργος»*, and *«που και που»* to *«ου και που»*, *«ολοι» το «ολη», «τρελοί» το «τρελή», « η βραδιά» το «οι βραδιά»*, *«να σε δω να ξυπνας»* transcribed as *«νασαι εδω να ξυπνας».*

These are typically short edit-distance substitutions that preserve the syllabic scaffold yet alter lexical identity enough to count as substitutions in WER. The effect is most visible in sustained vowels or unstressed syllables, where musical timing downplays phonemic contrast and the decoder normalizes toward frequent grapheme sequences.

(iii) Function words and clitics constitute a third locus of instability, with frequent deletions and sporadic insertions that subtly alter propositional content. For example the insertion in *«αφηνω γραμμα στο τραπεζι»* to *«αφηνω ενα γραμμα στο τραπεζι»*. Yet many cases go the other way: particles like "*μη, πως, δε"* drop in quick sequences, as in the various renderings of *«μην πεις πως δε σου το πα»* where one or more function words vanish. This behavior is consistent with singing's prosodic hierarchy, where unstressed function words are easily masked by accompaniment or elongated neighboring vowels, leading to deletion-heavy edit profiles.

(iv) Closely related are semantic substitutions where a phonetically proximate but semantically divergent word replaces the target. The contrast between *«Εισαι αγαπη εισαι κριμα»* and *«Εισαι αγαπη εισαι χρημα»* illustrates a single-token substitution with small phonetic distance but large semantic shift. Likewise, *«Αυτο που ειμαι χανω»* becomes *«Αυτο που ειμαι κανω»*, *«ευθυνομαι για την αγανακτηση των πολιτων» becomes «δε φτυνομαι για την αγανακτηση των πολιτων», «γιατι τ αδικο το ζουμε»* becomes *«γιατι τ αντικοτωσουμε»* preserving rhythm and meter while flipping meaning.

(v) Although far rarer after fine-tuning than at zero-shot, lyric-prior hallucinations still surface in specific configurations and songs, most memorably the repeated substitution with *«διδυμοτειχο blues»* in both transcribe and translate runs. This substitution reflects the decoder's attraction to culturally salient n-grams when acoustic evidence is noisy or ambiguous.

(vi) Boundary drift is another singing-specific artifact: melisma and syncopation shift perceived word boundaries, resulting to resegmentations that preserve many characters but misplace spaces and morpheme edges. The line *«στην αμμουδια ποτε του»* becoming *«στην αμμου διαποτετου»* is indicative. Here «αμμουδια» is fractured and recomposed across a rhythmic boundary. Similar drift appears in *«ετσι περνουσα πριν φυγω»* where variants like *«απ τη φυγω/πυφιγο»,* or *«που την ειδα στεκοτανε»* transcribed in *«που την ειδαστε κοτανε», «κι οταν πια σε ξεπερασω»* transcribed in *«κι οταν πιασε ξεπερασω»* emerge in other songs, pointing to the same boundary ambiguity rather than simple orthographic error.

(vii) Beyond segmentation, the systems display morphology and inflection drift, where lemmas are preserved but person, number, or case shift under acoustic pressure. For example, *«πανακριβο»* to *«πανακριβα».* In *«τωρα στα ερημα χωρια μεινανε γεροι και παιδια»* the output *«μειναμε χερι και παιδια»* flips number and introduces a near-phonetic neighbor (*χερι* for *γεροι*), while *«διπλα ο ιουδας κλαιει σκυφτος»* mutates to *«κλαις κι εφτος»*, merging person change with cluster corruption. Such errors point to incomplete adaptation of the decoder's morphological priors to the elongated and coarticulated acoustics of singing.

(viii) Finally, fixed expressions and named entities remain obscure under musical conditions. The phrase *«Βαλεντίνο άγιο»* is rendered as *«μπαλεντινο αιγιο»*, conflating the saint's name with the Greek city Αίγιο. Similarly, *«πλατινένιος»* is deformed to *«πλατιμένος»*. These distortions reveal a gap between the model's lexicalized phrase

knowledge from pretraining and its alignment to singing acoustics, where slight phonetic perturbations trigger retrieval of the wrong lexical template.

Given this difficult transcription example: “ολοι γελουσαν σαν τρελοι δεν ημουνα μαζι τους” we see a clear, capacity-driven progression and the benefit of task focus. Whisper Small collapses into severe semantic distortion and morpho-syntactic instability (“ολη η γενουσα… φευγει μου…”), losing the core verb (*γελούσαν*) and flipping the pronoun from *“tous”* (them) to *“mou”* (me). Whisper Medium recovers some structure but still exhibits lexical substitution (“χαιρούσα” for “γελούσαν”) and person/number mismatches, while keeping the pronoun error (*“μαζί μου”*). Whisper Large (2-to-1) stabilizes the verbal spine (“γελούσαν”) and the overall rhythm, yet retains small insertions (“εκεί μου όταν”) and still fails to restore *“μαζί τους.”* Finally, Whisper Large (transcribe-only) produces an almost perfect transcription, correct verb, order, and prosody with only a residual pronoun shift (*“μαζί μου”* vs. *“μαζί τους”*). Overall, larger capacity sharply reduces semantic drift and spurious insertions, while single-task (transcribe-only) fine-tuning further locks down function words, leaving only subtle pronoun inaccuracies to target next.

IV: Conclusions & Future Work

6. Conclusions

ALT remains a challenging subfield of speech technology, situated between ASR and MIR. Unlike standard ASR, ALT must operate on highly variable and musically complex input, where melodic pitch, rhythm, and accompaniment interfere with the phonetic structure of words. Singing alters vowel duration, intensity, and prosody, while background instruments further blur spectral cues that ASR systems depend on. As a result, transferring knowledge from speech-trained models to singing requires careful adaptation, robust alignment, and domain-specific data. Within this context, the goal of this thesis was to develop and evaluate an end-to-end pipeline capable of handling these challenges for Greek singing voice.

This thesis presented a complete end-to-end pipeline for ALT of Greek singing voice, integrating source separation, forced alignment, dataset preparation, and Whisper-based fine-tuning. The experimental analysis revealed that model scale, training objective, and task composition significantly influence transcription accuracy. Larger Whisper checkpoints consistently achieved lower normalized WER, reflecting their stronger representational capacity and ability to generalize across domains. Among all setups, transcribe-only fine-tuning created the most accurate and stable results.
For smaller models, multitask configurations with a 2:1 transcription-to-translation ratio performed much better than balanced 1:1 mixtures, which indicates that the model gradually masters the transcription task and benefits from preserving its dominant objective within the multitask framework. However, these improvements remain marginal compared to pure transcription training, which continues to offer the most consistent performance across scales. Finally, the two-stage fine-tuning strategy with pre-adaptation on Greek speech followed by specialization on singing did not gift measurable gains. The limited size of the Greek Common Voice corpus and the robustness of Whisper’s multilingual pretraining likely restricted the benefits of speech adaptation.

Overall, this study establishes the first systematic benchmark for Greek ALT and demonstrates that Whisper's multilingual architecture can generalize effectively to singing voice when trained on well-aligned, segment-level data. The findings confirm that task-pure adaptation remains the most efficient and robust strategy for domain specialization in low-resource singing transcription.

By addressing the specific challenges of Greek ALT-data scarcity, prosodic variability, and domain mismatch-this thesis lays the groundwork for future multilingual and cross-modal research in singing recognition and provides a foundation for extending ALT systems toward more expressive, musically aware, and linguistically rich applications.

7. Limitations and Future Work

Despite the encouraging results, several limitations constrain the current work. The Greek singing dataset remains relatively small and stylistically narrow, limiting model adaptation. Although the inclusion of a forced-alignment stage improved the temporal accuracy and consistency of supervision, further gains will depend on expanding both the size and diversity of the Greek singing corpus. Future research should focus on scaling the dataset by incorporating additional publicly available Greek songs and semi-supervised annotations to strengthen model robustness. Another promising direction is to revisit the two-stage adaptation strategy using a larger and more acoustically varied Stage-1 corpus, such as Greek podcast or broadcast audio, to provide stronger speech representations before adaptation to singing. Extending this pipeline to English ALT would also enable broader comparative research under high-resource conditions.

A particularly promising direction involves integrating LLMs, or better Greek-specific language modeling, for instance models such as Llama Krikri. These models could provide stronger linguistic knowledge of Greek, helping the system handle rare words, correct spelling and morphological mistakes, thus improving Greek transcriptions. Moreover, exploring parameter-efficient fine-tuning methods could reduce computational costs while maintaining competitive performance. Finally, extending the framework to capture expressive dimensions of singing, such as pitch, rhythm, and emotion, represents a natural continuation of this work. Integrating these aspects could lead to richer, musically aware models capable of both transcription and expressive analysis.

ABBREVIATIONS – ACRONYMS

Acronym	Meaning
ASR	Automatic Speech Recognition
ALT	Automatic Lyrics Transcription
WER	Word Error Rate
ML	Machine Learning
NLP	Natural Language Processing
DL	Deep Learning
SGD	Stochastic Gradient Descent
MSE	Mean Squared Error
RL	Reinforcement Learning
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
BLSTM	Bidirectional Long Short-Term Memory
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
GMM	Gaussian Mixture Model
EM	Expectation–Maximization
DNN	Deep Neural Network
CTC	Connectionist Temporal Classification
CER	Character Error Rate
PPG	Phonetic PosteriorGram
GAD	Greek Audio Dataset

References

[1] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[2] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, “On the Properties of Neural Machine Translation: Encoder–Decoder Approaches,” in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, D. Wu, M. Carpuat, X. Carreras, and E. M. Vecchi, Eds., Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 103–111. doi: 10.3115/v1/W14-4012.

[3] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM and other neural network architectures,” *Neural Netw.*, vol. 18, no. 5, pp. 602–610, July 2005, doi: 10.1016/j.neunet.2005.06.042.

[4] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” May 19, 2016, arXiv.1409.0473.

[5] M.-T. Luong, H. Pham, and C. D. Manning, “Effective Approaches to Attention-based Neural Machine Translation,” Sept. 2015, arXiv:1508.04025 .

[6] A. Vaswani *et al.*, “Attention is All you Need,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017.

[7] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-Based Models for Speech Recognition,” June 24, 2015, arXiv.1506.07503.

[8] G. Hinton *et al.*, “Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups,” *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012, doi: 10.1109/MSP.2012.2205597.

[9] A. Graves, A. Mohamed, and G. Hinton, “Speech Recognition with Deep Recurrent Neural Networks,” Mar. 22, 2013, *arXiv*: arXiv:1303.5778. doi: 10.48550/arXiv.1303.5778.

[10] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural ’networks,” in *Proc. 23rd Int. Conf. on Machine Learning (ICML)*, 2006, pp. 369–376, doi: 10.1145/1143844.1143891.

[11] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, “Listen, Attend and Spell,” Aug. 20, 2015, *arXiv*: arXiv:1508.01211. doi: 10.48550

[13] A. Gulati *et al.*, “Conformer: Convolution-augmented Transformer for Speech Recognition,” May 16, 2020, *arXiv*: arXiv:2005.08100.

[14] A. Graves, “Sequence Transduction with Recurrent Neural Networks,” Nov. 14, 2012, *arXiv*: arXiv:1211.3711.

[15] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,” Oct. 22, 2020, *arXiv*: arXiv:2006.11477.

[16] A. Babu *et al.*, “XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale,” Dec. 16, 2021, *arXiv*: arXiv:2111.09296.

[17] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust Speech Recognition via Large-Scale Weak Supervision,” Dec. 06, 2022, *arXiv*: arXiv:2212.04356.

[18] P. D. Green, “From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition,” *Interspeech 2004*, Jan. 2004, doi: 10.21437/INTERSPEECH.2004-668.

[19] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 5206–5210. doi: 10.1109/ICASSP.2015.7178964.

[20] R. Ardila *et al.*, “Common Voice: A Massively-Multilingual Speech Corpus,” Mar. 05, 2020, arXiv:1912.06670.

[21] A. Kruspe, “More than words: Advancements and challenges in speech recognition for singing,” Mar. 14, 2024, *arXiv*: arXiv:2403.09298.

[22] H. Fujihara and M. Goto, “Lyrics-to-Audio Alignment and its Application,” in *Multimodal Music Processing*, vol. 3, M. Müller, M. Goto, and M. Schedl, Eds., in Dagstuhl Follow-Ups, vol. 3. , Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2012, pp. 23–36. doi: 10.4230/DFU.Vol3.11041.23.

[23] T. Virtanen, A. Mesaros, and M. Ryynänen, “Combining pitch-based inference and non-negative spectrogram factorization in separating vocals from polyphonic music,” in *Proc. SAPA 2008*, 2008, pp. 17–22.

[24] D. Stoller, S. Durand, and S. Ewert, “End-to-end Lyrics Alignment for Polyphonic Music Using an Audio-to-character Recognition Model,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK: IEEE, May 2019, pp. 181–185. doi: 10.1109/ICASSP.2019.8683470.

[25] A. Mesaros and T. Virtanen, “Automatic Recognition of Lyrics in Singing,” *EURASIP J. Audio Speech Music Process.*, vol. 2010, no. 1, pp. 1–11, Dec. 2010, doi: 10.1155/2010/546047.

[26] M. Mauch, H. Fujihara, and M. Goto, “Lyrics-to-audio alignment and phrase-level segmentation using incomplete Internet-style chord annotations”.

[27] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, “DALI: a large Dataset of synchronized Audio, LyrIcs and notes, automatically created using

teacher-student machine learning paradigm," Sept. 2018, doi: 10.5281/zenodo.1492443.

[28] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised Cross-lingual Representation Learning for Speech Recognition," Dec. 15, 2020, *arXiv*: arXiv:2006.13979.

[29] G. R. Dabike and J. Barker, "Automatic lyric transcription from karaoke vocal tracks: Resources and a baseline system," in *Proc. Interspeech*, 2019.

[30] X. Gao, C. Gupta, and H. Li, "Automatic Lyrics Transcription of Polyphonic Music With Lyrics-Chord Multi-Task Learning," *IEEEACM Trans. Audio Speech Lang. Process.*, vol. 30, pp. 2280–2294, 2022, doi: 10.1109/TASLP.2022.3190742.

[31] S. Wu *et al.*, "SongTrans: An unified song transcription and alignment method for lyrics and notes," Oct. 10, 2024, *arXiv*: arXiv:2409.14619.

[32] X. Gao, X. Yue, and H. Li, "Self-Transriber: Few-shot Lyrics Transcription with Self-training," Mar. 02, 2023, *arXiv*: arXiv:2211.10152. doi: 10.48550/arXiv.2211.10152.

[33] H. Pham, Z. Dai, Q. Xie, M.-T. Luong, and Q. V. Le, "Meta Pseudo Labels," Mar. 01, 2021, *arXiv*: arXiv:2003.10580.

[34] L. Zhuo *et al.*, "LyricWhiz: Robust Multilingual Zero-shot Lyrics Transcription by Whispering to ChatGPT," July 25, 2024, *arXiv*: arXiv:2306.17103.

[35] Z. Song, J. Zhuo, Y. Yang, Z. Ma, S. Zhang, and X. Chen, "LoRA-Whisper: Parameter-Efficient and Extensible Multilingual ASR," June 07, 2024, *arXiv*: arXiv:2406.06619.

[36] Z. Zhang *et al.*, "SpeechLM: Enhanced Speech Pre-Training with Unpaired Textual Data," June 15, 2023, *arXiv*: arXiv:2209.15329.

[37] H. Fujihara, M. Goto, J. Ogata, K. Komatani, T. Ogata, and H. G. Okuno, "Automatic Synchronization between Lyrics and Music CD Recordings Based on Viterbi Alignment of Segregated Vocal Signals," in *Proceedings of the Eighth IEEE International Symposium on Multimedia*, in ISM '06. USA: IEEE Computer Society, Dec. 2006, pp. 257–264. doi: 10.1109/ISM.2006.38.

[38] C. Gupta, R. Tong, H. Li, and Y. Wang, "SEMI-SUPERVISED LYRICS AND SOLO-SINGING ALIGNMENT".

[39] C. Gupta, E. Yılmaz, and H. Li, "Automatic Lyrics Alignment and Transcription in Polyphonic Music: Does Background Music Help?," Oct. 22, 2019, *arXiv*: arXiv:1909.10200.

[40] L. Kürzinger, D. Winkelbauer, L. Li, T. Watzel, and G. Rigoll, "CTC-Segmentation of Large Corpora for German End-to-end Speech Recognition," vol. 12335, 2020, pp. 267–278. doi: 10.1007/978-3-030-60276-5_27.

[41] E. Rastorgueva, V. Lavrukhin, and B. Ginsburg, “NeMo forced aligner and its application to word alignment for subtitle generation,” in *Proc. Interspeech 2023*, 2023, pp. 5257–5258.

[42] R. Huang *et al.*, “Less Peaky and More Accurate CTC Forced Alignment by Label Priors,” July 18, 2024, *arXiv*: arXiv:2406.02560.

[43] E. Demirel, S. Ahlbäck, and S. Dixon, “Low Resource Audio-to-Lyrics Alignment From Polyphonic Music Recordings,” Feb. 18, 2021, *arXiv*: arXiv:2102.09202.

[44] S. Durand, D. Stoller, and S. Ewert, “Contrastive Learning-Based Audio to Lyrics Alignment for Multiple Languages,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, June 2023, pp. 1–5. doi: 10.1109/ICASSP49357.2023.10096725.

[45] J. Park, S. Yong, T. Kwon, and J. Nam, “A Real-Time Lyrics Alignment System Using Chroma And Phonetic Features For Classical Vocal Performance,” Jan. 17, 2024, *arXiv*: arXiv:2401.09200.

[46] G. Paraskevopoulos, T. Kouzelis, G. Rouvalis, A. Katsamanis, V. Katsouros, and A. Potamianos, “Sample-Efficient Unsupervised Domain Adaptation of Speech Recognition Systems A case study for Modern Greek,” Dec. 31, 2022, *arXiv*: arXiv:2301.00304.

[47] D. Damianos, G. Paraskevopoulos, and A. Potamianos, “MSDA: Combining Pseudo-labeling and Self-Supervision for Unsupervised Domain Adaptation in ASR,” June 02, 2025, *arXiv*: arXiv:2505.24656.

[48] L. G. Pillai, K. Manohar, B. K. Raju, and E. Sherly, “Multistage Fine-tuning Strategies for Automatic Speech Recognition in Low-resource Languages,” Nov. 07, 2024, *arXiv*: arXiv:2411.04573.

[49] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, “Pre-training on high-resource speech recognition improves low-resource speech-to-text translation,” Feb. 27, 2019, *arXiv*: arXiv:1809.01431.

[50] “The Greek Music Dataset ” in *ResearchGate*, doi: 10.1145/2797143.2797175.

[51] C.C. Papaioannou, I. Valiantzas, T. Giannakopoulos, M. Kaliakatsos-Papakostas, and A. Potamianos, “A dataset for Greek traditional and folk music: Lyra,” 2022.

[52] G. Paraskevopoulos, C. Tsoukala, A. Katsamanis, and V. Katsouros, “The Greek podcast corpus: Competitive speech models for low-resourced languages with weakly supervised data,” June 21, 2024, *arXiv*: arXiv:2406.15284.

[53] S. Vakirtzian *et al.*, “Speech Recognition for Greek Dialects: A Challenging Benchmark,” in *Interspeech 2024*, ISCA, Sept. 2024, pp. 3974–3978. doi: 10.21437/Interspeech.2024-2443.

[54] M. Turisini, M. Cestari, and G. Amati, “LEONARDO: A Pan-European Pre-Exascale Supercomputer for HPC and AI applications,” *J. Large-Scale Res. Facil. JLSRF*, vol. 9, no. 1, Jan. 2024, doi: 10.17815/jlsrf-8-186.

[55] S. Rouard, F. Massa, and A. Défossez, “Hybrid Transformers for Music Source Separation,” Nov. 15, 2022, *arXiv*: arXiv:2211.08553. doi: 10.48550/arXiv.2211.08553.