

A Language-Independent Method for Lyrics-Based Cover Song Identification Using Phoneme Transcriptions

A.H. Rathnaweera

2024



A Language-Independent Method for Lyrics-Based Cover Song Identification Using Phoneme Transcriptions

A.H. Rathnaweera

Index No : 19001371

Supervisor: Dr. M.I.E. Wickramasinghe

Co-Supervisor: Mr. W.R.N.S. Abeyweera

May 2024

Submitted in partial fulfillment of the requirements of the

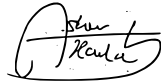
B.Sc in Computer Science Final Year Project (SCS4224)



Declaration

I certify that this dissertation does not incorporate, without acknowledgment, any material previously submitted for a degree or diploma in any university, and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organisations.

Candidate Name: A.H. Rathnaweera



.....
Signature of Candidate

Date: September 28, 2024

This is to certify that this dissertation is based on the work of A. H. Rathnaweera under my supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

Supervisor's Name: Dr. M.I.E. Wickramasinghe



.....
Signature of Supervisor

Date:

Co-Supervisor's Name: Mr. W.R.N.S. Abeyweera



.....
Signature of Co-Supervisor

Date: September 28, 2024

Abstract

Research in music information retrieval has traditionally focused on audio-based methods for cover song identification, analyzing elements like chords, melody, and harmony. However, these methods face scalability issues and struggle with covers that significantly alter the original music. Recent studies have shifted towards text-based approaches, using metadata and lyrics, since lyrics often remain consistent across different versions of a song, making these systems more robust. Nonetheless, these approaches are typically limited by language dependency in Singing Voice Recognition (SVR).

This thesis introduces a novel method for cover song identification that utilizes the phonetic transcriptions of lyrics. The approach is based on the premise that any spoken language can be transcribed into the phonetic transcriptions of International Phonetic Alphabet (IPA). We fine-tuned the XLS-R wav2vec 2.0 model using Connectionist Temporal Classification (CTC) to transcribe singing into IPA phonetic representations. Songs are then analyzed for similarity using the Levenshtein distance to identify cover versions.

The study achieved a 40.41% improvement in multilingual phoneme recognition in singing voices compared to the baseline. However, the results for English cover song identification were below those of the state-of-the-art lyrics-based cover song identification methods. Nonetheless, our proposed system achieved a Mean Average Precision (MAP) of 0.513 for identifying cover songs in Sinhala, a language not previously seen by the model during training or fine-tuning. This demonstrates the potential of using phonetic transcriptions for language-independent, lyrics-based cover song identification.

Acknowledgement

I would like to extend heartfelt appreciation to all the individuals and organizations that have provided invaluable support throughout the development of my research. In particular, I am grateful to Cognitive Systems and Time Series (COTS) Labs and its esteemed supervisors, Dr. Manjusri Wickramasinghe, Mr. Roshan Nadeesha, Mr. Pasindu Marasinghe, and Mr. Isuru Nanayakkara, for their unwavering commitment. Their expert guidance, technical support, and insightful feedback have been instrumental in shaping the direction and scope of my work.

I would also like to acknowledge the invaluable contributions of the senior research interns and past research interns of COTS Lab, whose dedication and expertise have laid a solid foundation for my endeavors.

Moreover, I express gratitude to my colleagues at COTS Lab, whose constructive criticism and feedback have helped improve the clarity, coherence, and organization of my research. Their diverse perspectives and expertise have been indispensable in refining arguments and enhancing research findings.

I am deeply grateful to all those who have contributed to this research study. Their support, encouragement, and expertise have been pivotal in enabling me to produce work that I hope will contribute meaningfully to the field.

List of Acronyms

MAP	Mean Average Precision
MSD	Million Song Dataset
MIREX	Music Information Retrieval Evaluation eXchange
TDNN	Time Delay Neural Network
MFCC	Mel-Frequency Cepstral Coefficients
SVR	Singing Voice Recognition
IPA	International Phonetic Alphabet
ASR	Automatic Speech Recognition
AP	Average Precision
DNN	Deep Neural Network
HMM	Hidden Markov Model
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
VQ	Vector Quantization
PER	Phone Error Rate
CER	Character Error Rate
WER	Word Error Rate
MLS	Multilingual LibriSpeech
CTC	Connectionist Temporal Classification

Table of Contents

1 Introduction	1
1.1 Background and Related Works	4
1.2 Problem	10
1.3 Proposed Solution	11
1.4 Significance	11
1.5 Research Aim	12
1.6 Research Questions	12
1.7 Objectives	12
2 Literature Review	13
2.1 Phoneme Recognition in Songs	13
2.2 Automatic Speech Recognition	13
2.2.1 The PyTorch-Kaldi Speech Recognition Toolkit	14
2.2.2 Wav2Vec	15
2.2.3 VQ-Wav2Vec	17
2.2.4 Wav2Vec 2.0	18
2.2.5 Cross-Lingual Automatic Speech Recognition (ASR)	20
2.3 Summary	22
3 Methodology	24
3.1 Research Design	24

3.1.1	Data Collection	25
3.1.2	Determining the Baseline	25
3.1.3	Fine-tuning the ASR Model	26
3.1.4	Evaluation of Phoneme Recognition Accuracy	27
3.1.5	Measuring the Similarity	28
3.1.6	Evaluation of Cover Song Identification	29
3.1.7	Comparative Analysis	31
3.2	Summary	32
4	Implementation	33
4.1	Baseline Model Implementation	33
4.1.1	Data Preprocessing	33
4.1.2	Phonemization the Lyrics	34
4.1.3	Running the Model	34
4.2	Fine-tuning XLS-R Model	34
4.2.1	Version Management	35
4.2.2	Data Preprocessing	35
4.2.3	Prepare the Tokenizer	37
4.2.4	Prepare the Feature Extractor	37
4.2.5	Prepare the Trainer	38
4.2.6	Training	39
4.3	Cover Song Identification	40
4.4	Summary	40

5 Results & Analysis	42
5.1 Results	42
5.1.1 Baseline	42
5.1.2 Fine-tuned XLS-R Model	43
5.1.3 Cover Song Identification	44
5.2 Analysis	44
5.2.1 Fine-tuned XLS-R Model	45
5.3 Cover Song Identification	47
5.4 Summary	49
6 Limitations & Future Works	50
7 Conclusions	52

List of Figures

1.1	Global recorded music industry revenues 1999 - 2023 (US\$ billions) [1].	1
1.2	Global recorded music revenues by segment 2023 [1].	2
1.3	Functional blocks in cover song identification by Serra <i>et al.</i> [2].	8
1.4	Overview of the fused system in [7].	9
2.1	An overview of the PyTorch-Kaldi architecture [17].	14
2.2	Illustration of pre-training from audio data X which is encoded with two convolutional neural networks that are stacked on top of each other. The model is optimized to solve a next time step prediction task [18].	15
2.3	(a) The vq-wav2vec encoder maps raw audio (X) to a dense representation (Z) which is quantized (q) to \hat{Z} and aggregated into context representations (C); training requires future time step prediction. (b) Acoustic models are trained by quantizing the raw audio with vq-wav2vec, then applying BERT to the discretized sequence and feeding the resulting representations into the acoustic model to output transcriptions [19].	18
2.4	Illustration of our framework which jointly learns contextualized speech representations and an inventory of discretized speech units [19].	19
2.5	XLSR approach [23].	21
2.6	Phone Error Rate (PER) of different ASR systems on TIMIT.	23
3.1	Overview of the proposed fine-tuning framework.	27
5.1	Phoneme recognition performance by music genres	45

List of Tables

1.1	Types of cover songs [2].	3
1.2	Musical facets [2].	5
1.3	Results of lyrics-recognition based and tonal-based cover detection system on Da-Tacos-voice. Da-Tacos-instr is the subset of the Da-Tacos test restricted to instrumental tracks. Standard errors are given in parenthesis [7].	10
1.4	Performances of lyrics-recognition and fused based cover detection system on Covers80 with various Singing Voice Recognition (SVR) framework. Lyrics-informed framework are informed by lyrics at test time [7].	10
2.1	PER(%) obtained on TIMIT [16] when progressively applying some techniques implemented within PyTorch-Kaldi [17].	15
2.2	Results for phoneme recognition on TIMIT in terms of PER [17].	16
2.3	TIMIT phoneme recognition in terms of PER [18].	18
2.4	TIMIT phoneme recognition accuracy in terms of PER [22].	20
4.1	Duration of each language in the DALI dataset (rounded to nearest minute).	36
5.1	Average PER for each language in baseline model.	43
5.2	Results of fine-tuned XLS-R model.	44
5.3	Cover song identification results.	44
5.4	Phoneme Error Rate (PER) improvement for each language, comparing baseline and fine-tuned models.	46

5.5 Comparison of English cover song identification results using the pro-	
posed method and two other methods.	47

Chapter 1 - Introduction

The Global Recorded Music Industry witnessed remarkable growth by 2023 (Figure 1.1), with its value surpassing 28 billion US dollars. This phenomenal expansion can be attributed to the increasing popularity of music streaming services, which accounted for a staggering 67.3% (Figure 1.2) of the industry's revenue [1]. With rapid technological advancements and growing accessibility to high-speed internet, music enthusiasts around the world have embraced streaming platforms as their primary source of music consumption. This paradigm shift has moved forward the music industry to new heights, revolutionizing the way people access and experience music.

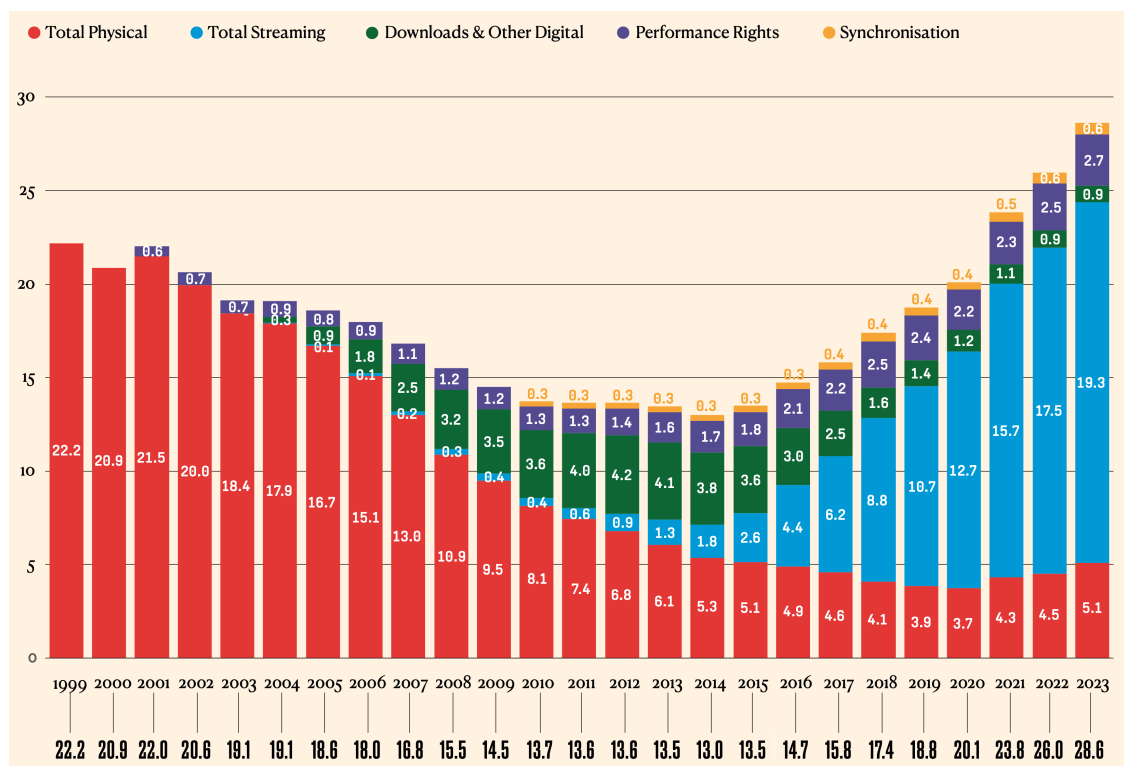


Figure 1.1: Global recorded music industry revenues 1999 - 2023 (US\$ billions) [1].

However, with these technological revolutions, the music industry faces a significant challenge from the unauthorized use of original musical works. Such illegal practices not only dissuade artists from creating new music but also impact their financial gains. One of the most common forms of copyright infringement in the music industry is copying an original song's composition, which includes the melody, harmony, and lyrics. Additionally, performing an original piece without exclusive permission is a prevalent

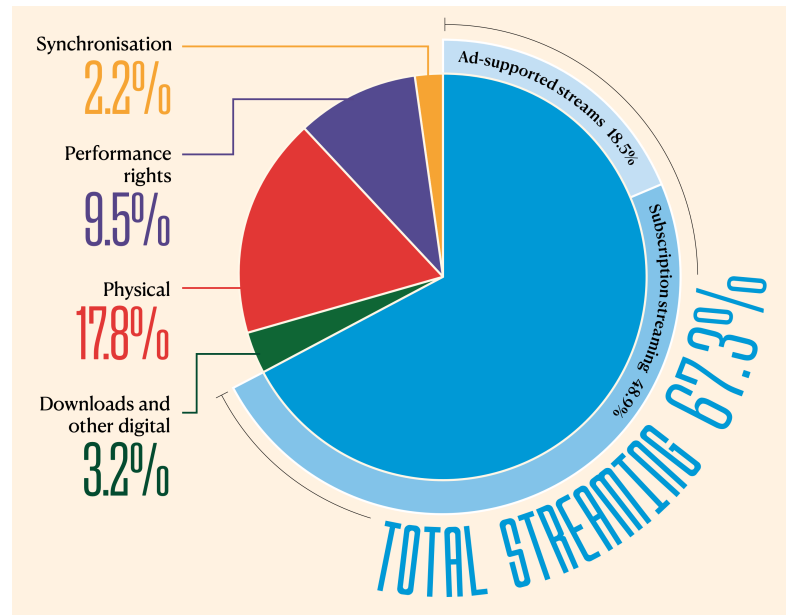


Figure 1.2: Global recorded music revenues by segment 2023 [1].

violation of copyright laws, particularly in regions such as Sri Lanka. As a result, copyright detection plays a crucial role in protecting the interests of the music industry.

The terms “copyright detection” and “cover song identification” will be used interchangeably in the later part of this thesis, as copied musical works can be viewed as cover versions of the original pieces. These copies and cover versions can be classified into various types, as outlined by Serrà, Gómez, and Herrera [2]. Each type serves as a label for a specific cover. In some cases, these labels can be observed in the titles of the covers themselves. Table I.1 provides a summary of some of these cover types along with their descriptions. Whenever a musical work falls into one of these types, it can be considered a copy or a cover of an original composition.

Table 1.1: Types of cover songs [2].

Type	Description
Remaster	Remastering is the process of creating a new master recording of a song or album, which often involves improving the sound quality through techniques like compression, equalization, and different endings or fade-outs. This is done to enhance the listening experience for the audience and to update the original recording for modern technology.
Instrumental	An instrumental is a type of cover song that does not include any sung lyrics. These versions may be released for different audiences, such as classical versions of pop songs or karaoke versions for people to sing or play along with.
Live Performance	A live performance is a recorded track of a song that was performed in front of an audience. This can be a recording of the original artist who previously released the song in a studio album or a recording of other performers.
Acoustic	In the context of music, an acoustic version of a song refers to a recording that features a different set of instruments and is often performed in a more intimate setting.
Demo	A demo is a rough recording made by musicians to capture their ideas and share them with others, such as record labels or bandmates. It can also be used as a simplified recording for publishing or copyright purposes. Demos are often made quickly and may not have the same quality as a fully produced studio recording.
Duet	A duet is a type of musical performance where two singers or instrumentalists perform together.

Continued on next page

Type	Description
Medley	A medley is a musical composition that combines several songs or tunes into a single piece. It is often performed live and involves seamlessly transitioning between different songs without stopping. The purpose of a medley is to create a unique and engaging performance that captures the listener's attention.
Remix	A remix is a type of cover song that involves altering the original recording by adding or subtracting elements, changing the equalization, dynamics, pitch, tempo, or other musical components. It can also involve substantial changes to the arrangement of the original work or a re-interpretation of the given work by combining fragments of two or more works. Remixes can be ambiguous and may not resemble the original work, making it a challenging task to identify cover songs.
Quotation	In the context of music, a quotation refers to the use of a brief segment of existing music in another work, similar to a quote in literature or speech. This can include borrowing a melody or incorporating a musical texture from another piece, but it is not considered a main part of the new work. This is one of the ways that musicians can create cover songs, which involve recording a different version of an existing song with changes to various musical facets, such as dynamics, tempo, and timbre.

1.1 Background and Related Works

As previously mentioned, music copyright detection can be considered as a task of cover song identification. Cover song identification involves the identification of alternative versions of existing musical compositions. These versions can exhibit significant differences from the original in terms of timbre, tempo, structure, and even fundamental

elements such as harmony and melody [3]. In recent years, the research community has shown a considerable interest in the field of cover song identification, owing to the remarkable growth of the music industry. Various musical aspects can be altered between cover songs, and Table 1.2 provides a description of some of the primary facets that can be modified.

Table 1.2: Musical facets [2].

Musical Facets	Description
Timbre	Timbre refers to the general color or texture of a sound, which can vary depending on different factors. These factors include production techniques, such as sound recording and processing, and instrumentation, which involves the use of different instruments, configurations, or recording procedures. Timbre variations can affect the overall sound of a cover version of a song, making it sound different from the original.
Tempo	Tempo refers to the speed or pace of a musical piece. In a cover version of a song, the tempo may change from the original version due to the performer's intention or feeling. This can result in small tempo fluctuations or even significant changes, which can affect the expressiveness and contextual feedback of the music.
Timing	Timing refers to the rhythmical structure of a piece of music and how it might change depending on the performer's intention or feeling. This can include changes in tempo, swing, syncopation, pauses, and other expressive deviations. Even in classical music, small tempo fluctuations are introduced for different renditions of the same piece, and tempo changes abound with different performers.

Continued on next page

Musical Facets	Description
Structure	In the context of music covers, structure refers to the arrangement of the song's sections, such as the intro, verse, chorus, bridge, and outro. Cover versions may modify the structure of the original song by adding or removing sections, repeating parts, or changing the order of the sections. These modifications can range from minor adjustments to radical transformations and can affect the overall feel and impact of the song.
Key	In music, the key refers to the tonality or pitch range of a piece. When performing a cover song, it is common to transpose the piece to a different key or tonality to suit the singer or instrument, or to create a different mood for the listener. This modification can significantly alter the sound of the song, but it is one of many changes that can be made to a cover version.
Harmonization	Harmonization is a musical characteristic that can change in cover versions of a song. It refers to chord progression, which can be modified by adding or deleting chords, substituting them with related chords, or changing the chord types or tensions. This is often done in the introduction and bridge passages, as well as in instrument solo parts, to create a unique interpretation of the original song.
Lyrics	In the context of cover songs, lyrics refer to the words of the original song that are sung by the new performer. In some cases, the lyrics may be translated into a different language to appeal to a wider audience. Changing the lyrics can also be a way for the new performer to put their own spin on the original work.

Continued on next page

Musical Facets	Description
Noise	Noise refers to any additional sounds that may be present in a song recording, such as audience reactions, speech, or compression artifacts. These noises can affect the overall sound quality and may be intentionally or unintentionally included in the cover version. In some cases, the original song may be difficult to recognize due to the presence of noise or other modifications.

Till recent days cover song identification is mainly based on audio-based systems. Audio-based cover song identification systems are built in a way that those are insensitive to the variations of musical attributes such as key, timbre, temp, and structure as those can vary between covers significantly. Those systems use tonal progression features such as chord, melody, and harmony since those features are commonly preserved between cover songs. Serrà, Gómez, and Herrera [2] extensively studied these cover song identification systems and found that we can group the functionality of those systems into four different functional blocks as shown in Figure 1.3. A summary of the several audio-based systems with techniques those system used in these four functional blocks is included in [2].

Even though audio-based cover song identification systems work well most of the time there are some critical scenarios that those systems fail to identify covers or copies in our context. In the introduction section, it was mentioned that there are different types of cover versions. Although the tonal progression features such as chord, melody, and harmony are mostly preserved in the cover songs there are situations in which these features also vary between covers where these audio-based systems fail. In a remix of a song, those features can be completely different from the original version of the song. For example, if the remix is done by converting the song into a different genre. The quotation of a music piece is another example of where audio-based systems may fail. If lyrics are copied and used in a new song with a completely different melody and chords audio-based systems fail to identify it as a copy or cover of the original work.

As a solution for those kinds of scenarios, researchers focused on leveraging textual data related to a song to identify covers in recent years. To the best of our knowledge, the

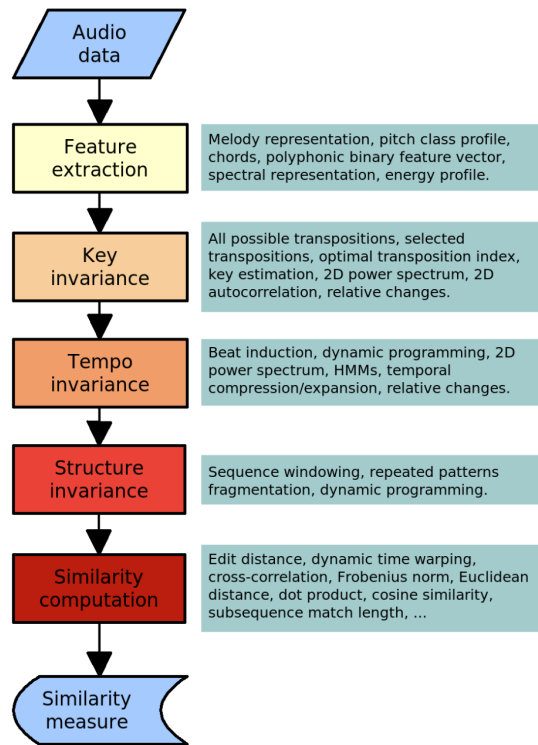


Figure 1.3: Functional blocks in cover song identification by Serra *et al.* [2].

research done by Correya, Hennequin, and Arcos [4] to investigate the usability of textual information such as metadata of a song and lyrics to identify covers in 2018 was the first approach to leverage textual information in the domain of cover song identification. They used a combination of text-based and audio-based approaches. As the text-based approaches a metadata-based approach and a lyrics-based approach were used. As the audio-based approach, they used the traditional cover song identification system proposed by Serrà, Gómez, and Herrera [2]. Then they compared the Mean Average Precision (MAP) [5] proposed text-based methods and combine methods against the state-of-the-art audio-based methods. The study achieved a 35.5% increase in MAP on the Million Song Dataset (MSD) [6] by using text-based methods. Also, this result proves that lyrics are often kept the same between the covers of a song.

There are some flaws in the previous approach. It was assumed that the lyrics of songs are available. However, this assumption may not hold for large musical collections. Also, the information carried by each modality is not optimally combined since each feature is only used in a separate part of a multi-layer database pruning method. As a solution, Vaglio, Hennequin, Moussallam, *et al.* [7] proposed a new approach for cover

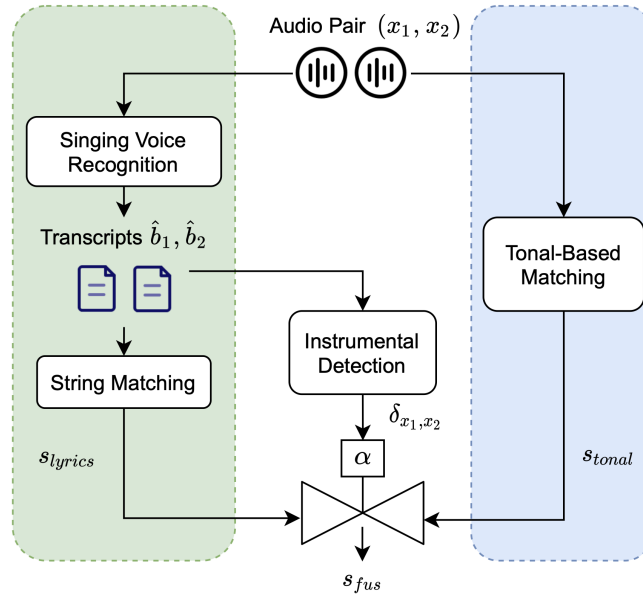


Figure 1.4: Overview of the fused system in [7].

detection that leverages lyrics information extracted from audio. This was the first time that lyrics transcripts from audio have been used explicitly for cover detection. Their approach combined a singing voice recognition framework with an audio-based cover detection method. The authors proposed a fused system for cover detection that uses transcription methods to obtain lyrics estimates for all songs. They suggest that a lyrics-recognition-based system is useful for covers with different tonal features but the same lyrics, such as Janis Joplin’s cover of Summertime. However, they also acknowledge that a pure lyrics-based system is insufficient for instrumental music, hence they use a tonal-based system as well and apply an instrumental detector to inform the fusion strategy. The overview of the system can be seen in the figure [1.4].

The authors used a framework that was considered the best in the 2020 lyrics transcription challenge organized by the Music Information Retrieval Evaluation eXchange (MIREX) [8] for transcribing lyrics. This framework employs a Time Delay Neural Network (TDNN) acoustic model trained on English tracks of the DALI dataset [9], along with an extended lexicon and a 3-gram word language model. The framework extracts Mel-Frequency Cepstral Coefficients (MFCC) from the input audio and outputs transcribed English words. As the tonal-based system, the authors used the Re-MOVE system which is the updated version of MOVE. This system achieved the second most accurate benchmark on the Da-Tacos dataset [10]. Re-MOVE has the advantage of be-

ing publicly available, unlike the best-performing system [11] reported in the dataset. The results for the fused system on the full Da-Tacos test and its Da-Tacos-voice subset are presented in Table 1.3, showing that the fused system outperforms the tonal-based system alone and validates the assumption that both methods are highly complementary. They found that the cover song detection accuracy directly depended on the transcription accuracy by running the experiment with Covers80 using a lyrics informed approach to simulate an ideal SVR framework. The results are presented in Table 1.4

Query	System	MAP (%)
Da-Tacos-voice	Lyrics	66.4 (0.4)
	Tonal	54.0 (0.4)
Da-Tacos-instr	Lyrics	0.45 (0.06)
	Tonal	47.8 (0.7)

Table 1.3: Results of lyrics-recognition based and tonal-based cover detection system on Da-Tacos-voice. Da-Tacos-instr is the subset of the Da-Tacos test restricted to instrumental tracks. Standard errors are given in parenthesis [7].

System	SVR	MAP (%)
Lyrics	Our	79.0 (0.6)
	Lyrics-informed	89.7 (0.4)
Fused	Our	88.5 (0.4)
	Lyrics-informed	93.6 (0.4)

Table 1.4: Performances of lyrics-recognition and fused based cover detection system on Covers80 with various SVR framework. Lyrics-informed framework are informed by lyrics at test time [7].

1.2 Problem

According to the literature, utilizing lyrics for cover song identification has proven to be highly effective in the development of accurate and scalable systems. However, a significant challenge faced by these lyrics-based systems is their reliance on specific

languages. Consequently, the primary research focus centers around addressing this dependency issue.

1.3 Proposed Solution

In the literature review, the text-based systems discussed involve the transcription of singing utterances into actual words. Once transcribed, text similarity is assessed using string-matching algorithms to identify covers. The meaning of the words themselves is deemed irrelevant for this task. Consequently, there is no necessity to transcribe the singing utterances into words of a specific language. Instead, the focus is on transcribing the singing utterances into a universal representation that can be applied to any language.

The study suggests transcribing the singing utterances into phonetic transcription by adapting a suitable [ASR](#) system. This phonetic transcription will enable the comparison of similarity between different phonetic transcriptions to identify covers. The International Phonetic Alphabet ([IPA](#)) will be utilized for this transcription, as it offers a language-independent approach. The [IPA](#) is a phonetic notation system based on the Latin script, which was developed by the International Phonetic Association in the late 19th century. Its purpose is to provide a standardized and consistent means of representing the sounds of human speech in written form [\[12\]](#).

1.4 Significance

This research has significant implications for the fields of computer science, music information retrieval, and the music industry. A phoneme-based approach to copyright detection has the potential to improve the accuracy and efficiency of current methods, which are limited in their ability to detect cover songs and versions across different languages. This research also has broader societal implications, as it contributes to the protection of intellectual property and supports fair compensation for music creators.

1.5 Research Aim

The aim of this research is to develop a lyrics-based language-independent system for cover song identification.

1.6 Research Questions

This research study attempts to answer the following research questions.

- RQ1.** How can available **ASR** models be adapted to transcribe the singing utterances of a song into a phoneme sequence in a language-independent manner?
- RQ2.** Which similarity measures are most effective for comparing phoneme sequences transcribed from songs to detect cover versions?
- RQ3.** What is the effectiveness of the proposed method in identifying cover songs, and how does it compare to existing methods?

1.7 Objectives

- RO1.** Select and investigate the most suitable **ASR** model for transcribing songs into phoneme sequences without language dependency.
- RO2.** Evaluate and optimize the selected **ASR** model for transcribing singing utterances in language independently into phonemes.
- RO3.** Identify a suitable similarity measures for comparing the phoneme sequences of songs for cover song identification.
- RO4.** Assess the efficacy of the proposed method in identifying cover songs and benchmark its performance against established methods.

Chapter 2 - Literature Review

Phoneme recognition within musical compositions is a niche and underexplored area of research, with scant prior studies and a noticeable lack of contemporary research. However, the field of [ASR](#) has seen significant progress, particularly in the identification of phonemes. This progress has led to breakthroughs in both cross-lingual speech and phoneme recognition capabilities.

2.1 Phoneme Recognition in Songs

The most recent study by Hansen [\[13\]](#) introduces a novel approach that integrates [MFCC](#) with temporal patterns to enhance phoneme recognition in vocal music. However, the outcomes do not yet reach a level of reliability that would allow for the method's application in cover song identification based on transcribed sequences of phonemes. Complementary research, such as Mesaros and Virtanen [\[14\]](#), investigates the employment of n-gram language models for the recognition of sung phonemes and words, while Gruhne, Dittmar, and Schmidt [\[15\]](#) delineates a system for phoneme detection using audio information retrieval and various classification strategies. Nonetheless, these studies also fall short in producing results that would fulfill the research objectives of this research satisfactorily.

2.2 Automatic Speech Recognition

This review is focused exclusively on [ASR](#) models with an emphasis on phoneme recognition capabilities. It centers on research that employs the TIMIT [\[16\]](#) dataset, which is renowned for providing transcriptions at both phonetic and word levels.

2.2.1 The PyTorch-Kaldi Speech Recognition Toolkit

In 2018, Ravanelli, Parcollet, and Bengio [17] spearheaded the PyTorch-Kaldi project, which aims to bridge the functionalities of the PyTorch and Kaldi toolkits, leveraging Kaldi's processing efficiency and PyTorch's flexible programming environment into a unified framework. This toolkit is adept at combining Kaldi's robust feature extraction, alignment, and decoding with PyTorch's dynamic acoustic model implementation, paving the way for sophisticated Deep Neural Network (DNN)-Hidden Markov Model (HMM) speech recognizers (Figure 2.1). It accommodates a spectrum of neural network models, including DNNs, Convolutional Neural Network (CNN)s, and Recurrent Neural Network (RNN)s, and endorses the creation of intricate architectures through various combinations of models, features, and labels. The platform facilitates the customization of neural networks and the exploration of a broad range of acoustic characteristics, activation functions, normalization techniques, cost functions, and optimization algorithms via editable configuration files. Trials on diverse datasets and tasks have validated PyTorch-Kaldi's potential in constructing state-of-the-art speech recognition models that stand up to current competitive standards. The condensed outcomes of these tests on the TIMIT [16] dataset are presented in Table 2.1.

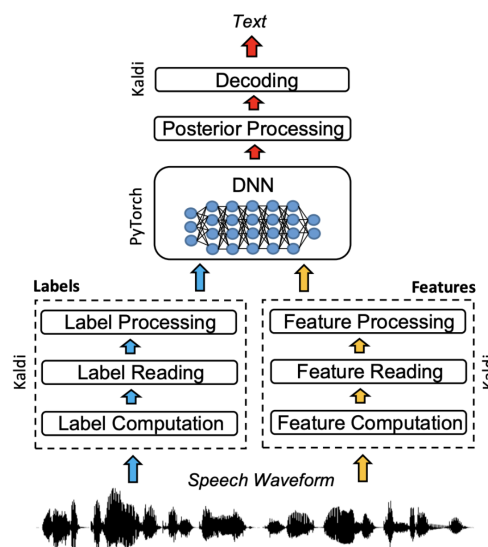


Figure 2.1: An overview of the PyTorch-Kaldi architecture [17].

	RNN	LSTM	GRU	Li-GRU
Baseline	16.5	16.0	16.6	16.3
+ Incr. Seq. length	16.6	15.3	16.1	15.4
+ Recurrent Dropout	16.4	15.1	15.4	14.5
+ Batch Normalization	16.0	14.8	15.3	14.4
+ Monophone Reg.	15.9	14.5	14.9	14.2

Table 2.1: PER(%) obtained on TIMIT [16] when progressively applying some techniques implemented within PyTorch-Kaldi [17].

2.2.2 Wav2Vec

The wav2vec paper by Schneider, Baevski, Collobert, *et al.* [18] presents an innovative approach to enhance ASR by utilizing unsupervised pre-training. This technique allows the model to learn from a vast corpus of unlabeled audio data, thereby developing generalizable speech representations without relying on transcriptions.

At the heart of the wav2vec framework there are two key components. The first is a convolutional feature encoder that processes raw audio waveforms and outputs latent representations of the speech. The second component is a context network that takes these latent representations and aggregates them to understand longer contextual dependencies within the speech input (Figure 2.2).

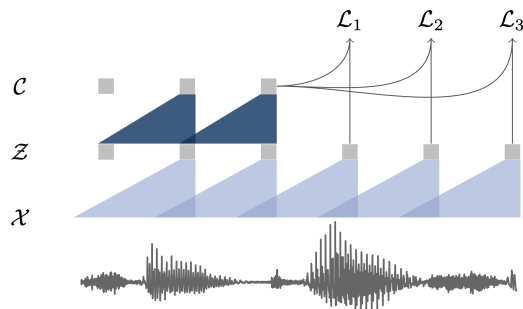


Figure 2.2: Illustration of pre-training from audio data X which is encoded with two convolutional neural networks that are stacked on top of each other. The model is optimized to solve a next time step prediction task [18].

The learning process involves a contrastive objective, where the model is trained to pre-

	dev	test
Li-GRU + MFCC [17]	-	16.7 ± 0.26
Li-GRU + FBANK [17]	-	15.8 ± 0.10
Li-GRU + FMLLR [17]	-	14.9 ± 0.27
Baseline	16.9 ± 0.15	17.6 ± 0.11
wav2vec (Librispeech 80h)	15.5 ± 0.03	17.6 ± 0.12
wav2vec (Librispeech 960h)	13.6 ± 0.20	15.6 ± 0.23
wav2vec (Librispeech + WSJ)	12.9 ± 0.18	14.7 ± 0.42

Table 2.2: Results for phoneme recognition on TIMIT in terms of **PER** [17].

dict the correct future audio samples from a set of possible options. This enables the model to finely tune its understanding of speech patterns and characteristics during the unsupervised pre-training phase. Once the model has been pre-trained on the unsupervised data, it undergoes a fine-tuning stage using a smaller labeled dataset. This stage introduces a new output layer, mapping the pre-trained representations to actual phonetic or character-based transcriptions used in speech recognition tasks.

The benefits of the wav2vec method are particularly notable because it can tap into the abundance of unlabeled audio data, which is often more readily available than labeled datasets. This characteristic is especially advantageous for languages with limited labeled resources.

In terms of performance, the wav2vec pre-trained models have shown remarkable improvements over traditional methods when fine-tuned, setting new benchmarks in the field as evidenced by the results detailed in Table 2.2 for the TIMIT [16] dataset. This approach not only advances the state-of-the-art in **ASR** but also opens up new avenues for speech recognition research, especially in the context of low-resource languages and dialects. The paper’s findings underscore the potential of unsupervised learning as a means to significantly reduce the dependency on costly labeled datasets while still achieving high levels of accuracy in **ASR**.

2.2.3 VQ-Wav2Vec

Baevski, Schneider, and Auli [19] present an evolution of the original wav2vec framework by integrating a quantization step that transforms continuous speech representations into discrete codes. This is similar to Vector Quantization (VQ), hence the name vq-wav2vec. The approach retains the self-supervised learning paradigm, leveraging unlabelled audio data to learn useful features for speech recognition tasks without the need for annotated data.

The architecture of vq-wav2vec builds upon a convolutional feature encoder that processes raw audio input into latent representations. The new quantization module then discretizes these latent representations into a finite set of tokens (Figure 2.3a). The quantization part is done by either using a Gumbel-Softmax [20] or online k-means clustering. These discrete tokens are similar to words in a text corpus, which enables the application of methods traditionally used in natural language processing to speech.

To effectively train this architecture, vq-wav2vec adopts a contrastive task from the BERT [21] model, another revolutionary model in natural language processing. In this setup, a certain percentage of the audio frames are masked, and the model must predict the correct quantized representation of these masked frames (Figure 2.3b). This self-supervised task compels the model to understand the context in which sounds occur, fostering the learning of rich, contextualized speech representations. By predicting the quantized representations of the masked audio frames, the model effectively learns to understand speech in context. This is a significant advance over the original wav2vec [18] model, which did not include a quantization step and dealt with continuous representations of audio.

The combination of self-supervised learning with discrete representation learning and BERT-like pretraining makes vq-wav2vec an efficient and powerful framework for speech processing. The discrete nature of the learned representations reduces the complexity and size of the model's output space, which can be particularly beneficial for downstream tasks such as ASR. By leveraging the compact and discrete output of the model, ASR models can be trained with much less labeled data than traditional models, which require extensive annotations. The BERT-inspired pretraining technique further enhances

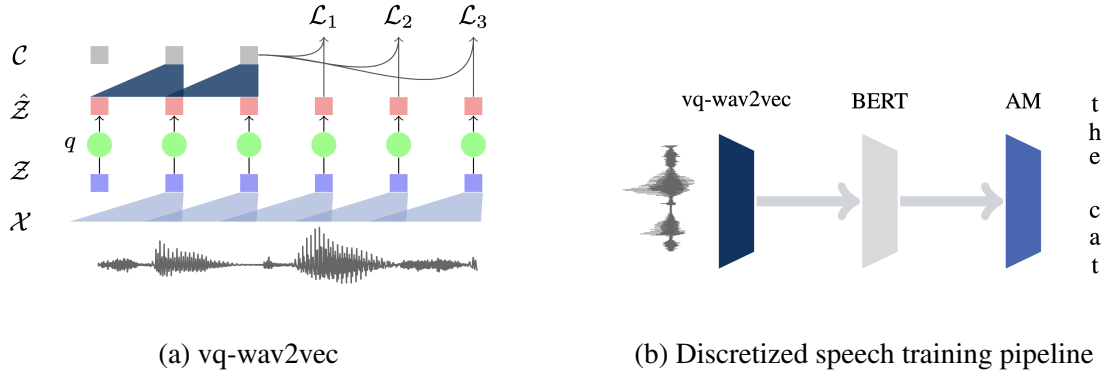


Figure 2.3: (a) The vq-wav2vec encoder maps raw audio (X) to a dense representation (Z) which is quantized (q) to \hat{Z} and aggregated into context representations (C); training requires future time step prediction. (b) Acoustic models are trained by quantizing the raw audio with vq-wav2vec, then applying BERT to the discretized sequence and feeding the resulting representations into the acoustic model to output transcriptions [19].

the model’s performance (Table 2.3) , as it enables the vq-wav2vec to capture the nuanced context of speech, an essential aspect of understanding and processing spoken language.

2.2.4 Wav2Vec 2.0

The wav2vec 2.0 proposed by Baevski, Zhou, Mohamed, *et al.* [22] details an advanced method in the field of speech processing, proposing a self-supervised framework for

	dev PER	test PER
Li-GRU + fMLLR [17]	–	14.9
wav2vec [18]	12.9	14.7
Baseline (log-mel)	16.9	17.6
vq-wav2vec, Gumbel	15.34	17.78
+ BERT small	9.64	11.64
vq-wav2vec, k-means	15.65	18.73
+ BERT small	9.80	11.40

Table 2.3: TIMIT phoneme recognition in terms of PER [18].

learning speech representations from raw audio data. This framework also aims to alleviate the heavy reliance on large sets of labeled data, which is a common bottleneck in the development of speech recognition systems.

The innovation of wav2vec 2.0 lies in its two-step approach, starting with pre-training. During this phase, the model is fed a vast corpus of unlabeled audio. The model uses a multi-layer CNN to transform the raw waveform into latent representations. Subsequently, these representations are partially masked (similar to masked language modeling in BERT) and passed through a Transformer network, which is tasked with predicting the masked audio segments. This prediction task forces the model to understand the nuances of speech in different contexts, effectively learning useful speech features without any labeled data.

The next phase is fine-tuning, where the pre-trained model is refined with a smaller set of labeled data tailored to a specific task like speech recognition. This phase customizes the model to the nuances and requirements of the particular task, ensuring the general features learned previously are optimally applied. The architecture of wav2vec 2.0 is particularly noteworthy (Figure 2.4). The convolutional feature encoder takes raw audio input and generates latent representations that capture the essence of the speech signal. These representations are then input into the context network, built on the Transformer architecture, known for its effectiveness in capturing long-range dependencies and contextual information.

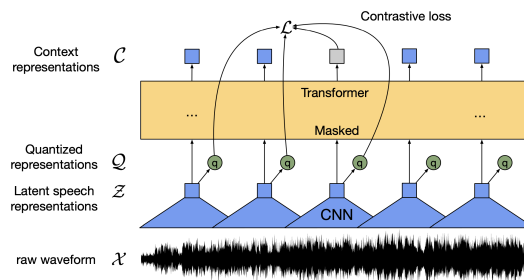


Figure 2.4: Illustration of our framework which jointly learns contextualized speech representations and an inventory of discretized speech units [19].

In training the model, a contrastive loss function is utilized during the pre-training phase. This loss function encourages the model to correctly identify the true speech representation from among a set of incorrect ones, improving its discriminative power. Addition-

	dev PER	test PER
Li-GRU + fMLLR [17]	-	14.9
wav2vec [18]	12.9	14.7
vq-wav2vec [19]	9.6	11.6
wav2vec 2.0 (no LM)		
LARGE (LS-960)	7.4	8.3

Table 2.4: TIMIT phoneme recognition accuracy in terms of PER [22].

ally, a quantization module is introduced, discretizing the continuous latent representations into a set of distinct categories, facilitating the contrastive learning process.

The model’s performance was rigorously evaluated on standard speech recognition benchmarks. The results were impressive, showcasing the ability of wav2vec 2.0 to achieve state-of-the-art results in phoneme recognition and other speech-related tasks without even using a language model (Table 2.4). These results were particularly striking given that wav2vec 2.0 requires much less labeled data than traditional speech recognition approaches.

2.2.5 Cross-Lingual ASR

The XLSR framework introduced by Conneau, Baevski, Collobert, *et al.* [23] enhances the wav2vec 2.0 model to understand multiple languages through a novel pre-training technique on raw audio waveforms from 53 different languages. Crucially, the model incorporates a shared quantization module that processes the outputs of the feature encoder, yielding multilingual quantized speech units. These discrete units facilitate the learning of a universal set of speech representations that are used across different languages (Figure 2.5).

The essence of the XLSR approach lies in how these quantized units serve as targets for a Transformer architecture, which is trained using contrastive learning. By encouraging the model to align similar sounds and distinguish dissimilar ones, the system learns to associate discrete tokens with similar acoustic patterns across languages. This effectively

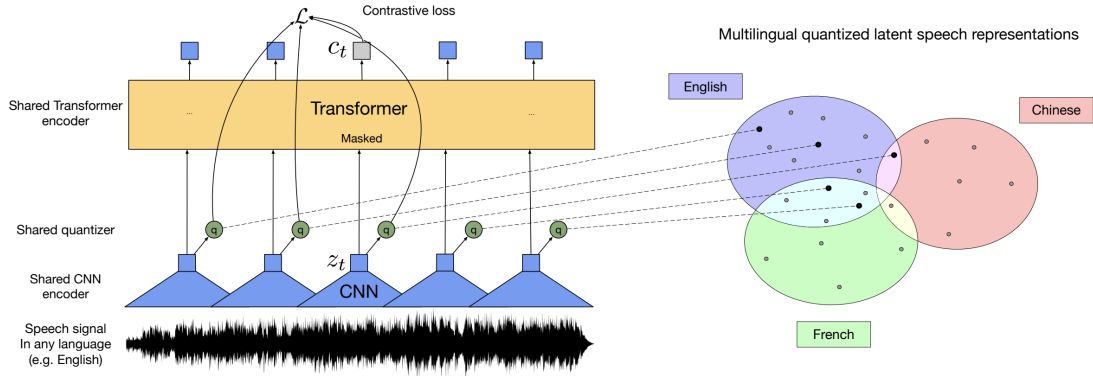


Figure 2.5: XLSR approach [23].

builds “bridges” between languages, enabling the model to transfer knowledge learned from one language to another.

In testing, the XLSR model achieved superior performance in speech recognition tasks across a multitude of languages, notably improving on languages with sparse training data. It excelled in comparison to monolingual baselines and even showed potential in zero-shot learning scenarios, where it could recognize speech from languages it was not explicitly trained on. This demonstrates the XLSR model’s ability to capture universal speech features, making it a significant stride towards robust, language-agnostic speech recognition technology.

Xu, Baevski, and Auli [24] fine tuned the above wav2vec 2.0 XLSR-53 model to transcribe unseen languages. The approach involves mapping phonemes from the training languages to the target language using articulatory features. This mapping helps address the issue of out-of-vocabulary phonemes in the target languages during testing. Experimental results demonstrate that this straightforward method outperforms previous approaches that utilized task-specific architectures and only utilized a portion of a monolingually pretrained model.

The XLS-R model introduced by Babu, Wang, Tjandra, *et al.* [25] extends the wav2vec 2.0 framework to achieve self-supervised cross-lingual speech representation learning at a large scale. This model leverages a convolutional feature encoder that transforms raw audio into latent speech representations. These are then processed by a Transformer to produce contextualized outputs. A novel aspect of the XLS-R training methodology

involves a contrastive task where spans of feature encoder outputs are masked and the model must identify the correct quantized latent representations among several distractors.

The training of the XLS-R model was conducted on an extensive and diverse corpus totaling 436,000 hours of publicly available speech data, sourced from multiple datasets. These include VoxPopuli, with 372,000 hours of parliamentary speech in 23 European languages; Multilingual LibriSpeech (MLS), comprising 50,000 hours primarily in English; CommonVoice, featuring 7,000 hours of read speech across 60 languages; VoxLingua107, which adds 6,600 hours of YouTube content in 107 languages; and BABEL, consisting of about 1,000 hours of conversational telephone speech in 17 languages.

Significant improvements were observed with the deployment of the XLS-R model across various benchmarks. In speech translation tasks on the CoVoST-2 benchmark, the model improved the previous state-of-the-art by an average of 7.4 BLEU points over 21 translation directions into English. For speech recognition, XLS-R reduced error rates by 14-34% on average relative to the best known prior works on datasets such as BABEL, MLS, CommonVoice, and VoxPopuli. Furthermore, it set a new state-of-the-art in language identification on the VoxLingua107 benchmark. These results demonstrate the model's capability to leverage large-scale, cross-lingual pretraining to significantly enhance speech processing tasks across a broad spectrum of languages and applications.

2.3 Summary

This chapter comprehensively explores the field of Automatic Speech Recognition (ASR), with a specific focus on phoneme recognition within musical compositions and broader multilingual contexts. It highlights several key studies and innovations that have significantly advanced the capabilities of ASR systems. The review details the integration of Mel-frequency cepstral coefficients (MFCC) with temporal patterns to enhance phoneme recognition in music, although these methods have not yet achieved a reliability suitable for cover song identification. It also discusses the PyTorch-Kaldi toolkit, which merges the strengths of PyTorch and Kaldi to create sophisticated DNN-HMM speech recognizers. Innovations like Wav2Vec and its derivatives (VQ-Wav2Vec, Wav2Vec 2.0) are

noted for their use of unsupervised pre-training from large unlabeled datasets, reducing the reliance on labeled data and expanding the technology’s applicability to under-resourced languages. The XLS-R framework extends Wav2Vec 2.0’s methodology to understand multiple languages through shared quantization, achieving impressive cross-lingual speech recognition performance. These technological advancements underscore a shift towards models that can learn from vast amounts of unlabeled data, enhancing phoneme recognition and pushing the boundaries of what modern ASR systems can achieve. In summary, the literature review, as illustrated in Figure 2.6, indicates that wav2vec 2.0 models represent the current state-of-the-art in phoneme recognition.

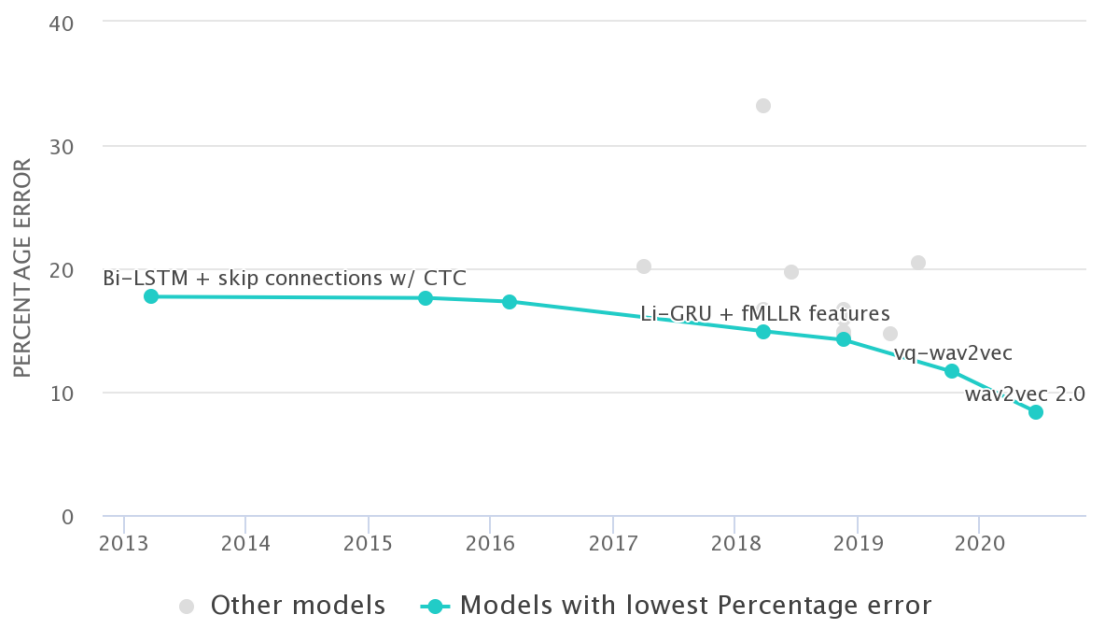


Figure 2.6: PER of different ASR systems on TIMIT.

Chapter 3 - Methodology

Hair, Money, Samouel, *et al.* [26]’s Research Onion model has been adapted for planning the research methodology in this study, as highlighted by Melnikovas [27]. The research approach employed in this study is deductive, with a hypothesis stating that an ASR model, when appropriately trained, can transcribe song lyrics into phonemes across different languages. These phoneme transcriptions can then be utilized to identify songs with identical lyrics. The research strategy is experimental in nature, involving the creation of an intervention in the form of a trained ASR model, followed by the observation of outcomes.

A monomethod research design is adopted, focusing on quantitative measurements of phoneme transcription accuracy and copyright infringement detection. The time horizon for the study is cross-sectional, as data analysis will be conducted at a single point in time rather than tracking changes over an extended period.

3.1 Research Design

The research design for this study encompasses a comprehensive methodology aimed at developing and evaluating a phoneme recognition system tailored for cover song identification. It incorporates a series of structured sub-sections, each detailing critical components of the research process: data collection involves using the DALI and specific cover song datasets; determining the baseline employs the wav2vec 2.0 model; fine-tuning the ASR model leverages the advanced XLS-R framework; phoneme recognition accuracy is meticulously measured; similarity in cover songs is quantified through Levenshtein distance; cover song identification efficacy is assessed using a precise metric, and comparative analysis benchmarks the system against existing technologies. This multi-faceted approach ensures a robust evaluation of the system’s capabilities in recognizing and processing phonemes across diverse musical pieces and languages.

3.1.1 Data Collection

For the development and evaluation of the Phoneme Recognition System, the DALI[28] dataset is utilized as the primary source of training and testing data. This dataset is a comprehensive resource that provides synchronized audio files along with their corresponding full-duration lyrics, phonemes, and vocal melody notes. The DALI dataset uniquely organizes lyrics into four distinct hierarchical levels: individual notes linked with specific textual content, words, lines, and paragraphs. Each audio track in the DALI dataset is further enriched with multimodal data including genre, language, artist details, album artwork, and URLs to related music videos. The audio files are accessible for download using YouTube URLs provided within the dataset.

To assess the system’s performance in cover song identification, two datasets are employed: Covers80[29] and Sinhala Cover Songs[30]. The Covers80 dataset is a recognized benchmark in the domain of cover song identification and comprises 160 songs, with each original track paired with its cover version, totaling 80 pairs. The Sinhala Cover Songs dataset, specifically curated for the cover song identification task within the Sinhala music context, includes 14 original songs along with their respective cover versions. This dataset has been previously utilized in studies focusing on cover song identification tasks for Sinhala music, providing a specialized resource for evaluating system performance in a linguistically and culturally specific context.

3.1.2 Determining the Baseline

The literature review identifies wav2vec 2.0 as the current state-of-the-art in phoneme transcription, as indicated by the enhancements reported by Xu, Baevski, and Auli [24] in transcribing languages not encountered during training. This capability is particularly relevant to the objectives of this research. Therefore, Xu, Baevski, and Auli [24]’s model is adopted as the baseline phoneme recognition system for this study.

A key challenge in using the DALI dataset for evaluating the baseline model is the difference in phonetic transcription formats. While both the DALI dataset and Xu, Baevski, and Auli [24]’s model employ the IPA for phonetic transcriptions, the specific alpha-

bets or labeling systems used differ. This discrepancy makes it impractical to directly measure the transcription accuracy of the baseline model using the DALI dataset’s transcriptions, as manual conversion of lyrics into phonemes is time-consuming and not feasible for this research.

During the literature review, two tools were identified for potentially automating the phonemization of lyrics: ESpeak¹ and Phonetisaurus². For the purposes of this study, ESpeak is selected to handle phoneme transcriptions due to its compatibility with the baseline model and its support for over 100 languages. The use of ESpeak ensures consistency between the model’s output vocabulary and the phonemic transcriptions of the dataset’s lyrics, which is crucial for accurate evaluation and comparison.

3.1.3 Fine-tuning the ASR Model

The XLS-R^[25] model, recognized as the state of the art in large-scale, cross-lingual speech representation learning, is selected as the pre-trained model for fine-tuning to specialize in phoneme recognition of singing voices. This choice is based on the analysis conducted in literature review, which identifies XLS-R as superior to the previously utilized XLSR-53^[23] model, which was considered state of the art at the time of Xu, Baevski, and Auli^[24]’s research.

For this specific task, phoneme transcriptions provided by the DALI dataset are employed instead of those generated by the ESpeak phonemization tool. The DALI dataset uses fewer phoneme labels compared to ESpeak, which is advantageous as fewer labels can lead to improved classification results in this context. Additionally, only a subset of the DALI dataset is used to maintain a balanced representation across languages.

The fine-tuning of the XLS-R model is conducted using the Connectionist Temporal Classification (CTC) algorithm^[31]. CTC is particularly suited for this application as it is an alignment-free method, eliminating the need for manual alignment of audio inputs to their corresponding labels—a process that becomes impractical given the size of the dataset. The alignment-free nature of CTC is due to its loss function, which marginalizes

¹<https://github.com/espeak-ng/espeak-ng>

²<https://github.com/AdolfVonKleist/Phonetisaurus>

over all possible alignments, thereby simplifying the training process. A **CTC** layer is integrated above the transformer structure, and the entire model undergoes fine-tuning to minimize the **CTC** loss. The feature extractor remains frozen during this fine-tuning phase because it has already been adequately trained during the pretraining stage (Figure 3.1).

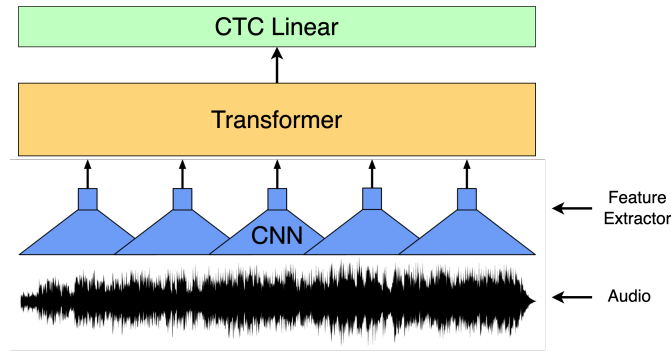


Figure 3.1: Overview of the proposed fine-tuning framework.

In this study, two versions of the XLS-R model are employed: one with 300 million parameters and another with 2 billion parameters. Different learning rates are tested to determine the optimal rate for achieving the best results in phoneme recognition tasks. This approach allows for a thorough exploration of the model’s capabilities and the effects of scale on performance.

3.1.4 Evaluation of Phoneme Recognition Accuracy

The phoneme transcription accuracy of the fine-tuned **ASR** model will be rigorously evaluated using test subsets derived from the DALI [28] dataset.

The efficacy of phoneme transcription will be quantitatively measured using the **PER**, a standard metric in speech recognition that reflects the model’s precision in capturing phonemic details. **PER** is analogous to the Word Error Rate (**WER**) but focuses on the more granular phoneme level.

The computation of PER involves the following steps:

1. Alignment: Initially, the **ASR** model’s predicted phoneme sequence is aligned with the reference sequence to enable direct comparison. Dynamic programming is typically

employed to optimize this alignment by identifying the most cost-effective sequence of edits.

2. Error Categorization: Following alignment, transcription discrepancies are classified as: - Substitutions: Incorrect recognition of a phoneme as another. - Insertions: Unwarranted addition of a phoneme not present in the reference. - Deletions: Failure to recognize a phoneme that exists in the reference.

3. Calculation Process: The total **PER** is calculated by summing the substitution, insertion, and deletion errors and dividing by the number of phonemes in the reference. The formula is expressed as:

$$\text{PER} = \frac{S + I + D}{N}$$

Here, S represents substitutions, I denotes insertions, D stands for deletions, and N is the count of reference phonemes.

4. Normalization: The **PER** is typically presented as a percentage. A lower percentage signifies higher transcription accuracy.

5. Analytical Insight: The **PER** offers nuanced insights into the **ASR** model's phonemic recognition capabilities, which is particularly vital for understanding its performance across various languages and dialects, where precise phoneme differentiation is critical.

This thorough approach to evaluating phoneme transcription ensures that the **ASR** model's capabilities are accurately understood and benchmarked, paving the way for targeted improvements where necessary.

3.1.5 Measuring the Similarity

In the process of identifying cover songs, it is essential to measure the similarity between phoneme transcriptions of the query song and those already stored. For this purpose, Levenshtein distance[32] is employed as the primary metric. Levenshtein distance, also known as edit distance, quantifies the similarity between two sequences by calculating the minimum number of single-character edits required to change one sequence into the

other. These edits can include insertions, deletions, or substitutions.

This metric is particularly relevant in the context of phoneme transcriptions because it allows for a detailed comparison of the sequences at a granular level. For example, if one transcription includes a phoneme that the other does not, Levenshtein distance will count this as a single edit—either an insertion or a deletion, depending on the sequence being modified. Similarly, if a phoneme needs to be changed to another to match the second sequence, this is counted as a substitution.

Levenshtein distance is akin to the **PER**, a common measure in speech recognition that also counts the number of edits needed to correct a transcription error. By using Levenshtein distance, the methodology directly assesses the extent of difference or similarity between the phoneme sequences of the original songs and their respective covers. This approach is critical for the task at hand, as the accurate detection of cover songs relies on identifying even subtle phonetic variations that distinguish different performances of the same musical piece.

3.1.6 Evaluation of Cover Song Identification

The evaluation of cover song identification utilizes the Covers80[29] and Sinhala Cover Songs[30] datasets, as previously outlined. The Covers80 dataset serves a dual purpose: it measures the accuracy of English cover song identification and provides a benchmark against other systems in the field, being the most commonly used dataset for such comparisons. Its widespread adoption in research makes it an ideal standard for assessing system performance in recognizing English cover songs.

In contrast, the Sinhala Cover Songs dataset is employed to evaluate the accuracy of identifying Sinhala cover songs and to assess the system’s performance in a multilingual context with a focus on an unseen language. This dataset offers a unique perspective on how well the phoneme recognition system adapates to languages that are less commonly represented in global datasets.

However, the challenge arises in fully evaluating multilingual cover song identification. While the Sinhala Cover Songs dataset supports testing in a less common linguistic

context, it is insufficient for a comprehensive assessment across multiple languages. Although there have been studies in other languages, such as Chinese cover song identification, access to corresponding datasets is often restricted or non-existent. Creating a new multilingual dataset to fill this gap would be akin to undertaking a separate research project, given the extensive time and resources required.

Due to these limitations, this study focuses solely on evaluating the English and Sinhala cover song identification accuracy. This decision is driven by the practical constraints of dataset availability and the scope of the current research project. The methodology, therefore, emphasizes detailed analysis within the accessible datasets to yield insights into the system’s performance in recognized and less commonly studied linguistic settings.

The key metric for this evaluation, **MAP**, is a critical indicator that combines precision and recall elements, prioritizing the rank-order of predictions. In the context of cover song identification, **MAP** measures how well a system can recognize different performances of the same underlying musical composition.

The computation for **MAP** involved following steps:

1. Query and Retrieval Process: A ‘query’ refers to an original track which the system uses to scour a database for all potential ‘cover’ versions. The identified covers are then ranked according to the likelihood of them being true covers.
2. Precision at K (P@k): For any given query, the precision at a certain rank k signifies the proportion of retrieved tracks within the top k that are verified covers.
3. Average Precision (Average Precision (**AP**)): This figure is the mean of precision scores recorded at every instance a new correct cover is found within the retrieval list.
4. Mean Average Precision (**MAP**): **MAP** represents the average of these AP scores across all queries in a test set. It is expressed as:

$$\mathbf{MAP} = \frac{1}{Q} \sum_{q=1}^Q AP_q$$

where AP_q denotes the Average Precision for each individual query q .

A system with a high MAP score is not only adept at identifying cover songs but also excels in ranking the most relevant covers at the top of its results, providing an efficient search outcome.

MAP is particularly critical in the evaluation of cover song identification systems because it accounts for both the correctness of the matches and their ranked order, providing insight into the practical performance of the system, where users would benefit from quickly finding the most fitting covers.

3.1.7 Comparative Analysis

The comparative analysis in this study focuses on two main aspects: phoneme recognition accuracy and cover song identification accuracy. The performance of the developed system is benchmarked against existing systems using standard metrics and datasets within the domain, facilitating a direct comparison without the need for reproducing previous studies.

For phoneme recognition, accuracy is assessed using the DALI dataset, which is well-recognized for its comprehensive annotation of phonemes, making it suitable for a reliable accuracy assessment. In the case of cover song identification, the accuracy is also compared using established benchmarks. The Covers80 and Sinhala Cover Songs datasets provide a basis for these comparisons. Covers80, being a widely used benchmark in the field, enables the assessment of English cover song identification capabilities against other systems. Similarly, the Sinhala Cover Songs dataset allows for evaluation in a less common linguistic context, offering insights into the system's performance in identifying covers in a multilingual setting.

The comparative analysis leverages these standard datasets to ensure that the results are both valid and comparable to existing research. By using these recognized benchmarks, the study ensures that the evaluations are consistent with industry standards, thus providing a credible comparison of the developed system's performance against the current state of the art in both phoneme recognition and cover song identification. This approach not only enhances the reliability of the comparisons but also contributes to the broader understanding of system capabilities in a multi-lingual context.

3.2 Summary

The methodology chapter of the research thesis outlines a structured approach to developing and evaluating a phoneme recognition system for cover song identification. It incorporates a detailed plan based on the Research Onion model, employing a deductive research approach and an experimental strategy. The chapter is divided into several key sections: Data Collection, which leverages the DALI and Covers80 datasets; Determining the Baseline, utilizing the state-of-the-art wav2vec 2.0 model; fine-tuning the ASR Model, applying the advanced XLS-R model; Evaluation of Phoneme Recognition Accuracy, using Phoneme Error Rate (PER); Measuring the Similarity, employing Levenshtein distance; Evaluation of Cover Song Identification, focusing on the MAP metric; and Comparative Analysis, benchmarking against existing systems. Each section is meticulously designed to ensure a comprehensive assessment of the system's capabilities across diverse musical pieces and languages, supporting a robust evaluation and comparison framework for phoneme recognition and cover song identification.

Chapter 4 - Implementation

This chapter outlines the implementation of the baseline model and fine-tuning the XLS-R model. Initially, the chapter details the preprocessing steps for the DALI dataset to prepare it for use with the model, including the retrieval of audio files and conversion to a compatible format. It then describes the phonemization process to align dataset phoneme labels with the model's output, followed by the implementation steps for running the model using the Huggingface platform. The fine-tuning of the XLS-R model variants with different parameter sizes is then discussed, including specific modifications and preparations for the training environment, data handling, and tokenizer setup to ensure optimal model performance.

4.1 Baseline Model Implementation

Xu, Baevski, and Auli [24]'s fine-tuned XLSR-53 model is utilized as the baseline model for this study as mentioned above. As the first step DALI dataset is preprocessed to use with the selected model.

4.1.1 Data Preprocessing

To evaluate the phoneme recognition accuracy of the baseline model, a representative subset of songs from the DALI dataset is selected, covering all languages present in the dataset. The DALI dataset itself does not include audio files; thus, audio files are retrieved from YouTube using the URLs provided within the dataset. Some songs could not be accessed due to removal from YouTube or regional restrictions and were consequently excluded from the dataset. The Python library 'pytube' is employed to download these video files.

Once downloaded, the video files, which are in MP4 format, need to be converted to a compatible audio format for the model. The baseline model requires audio files with a 16 kHz sampling rate. Using the Python library 'pydub', the MP4 files are converted

into Waveform Audio File Format (WAV) at the required 16 kHz sampling rate.

4.1.2 Phonemization the Lyrics

The phoneme output from the baseline model is in International Phonetic Alphabet (IPA) symbols, whereas the DALI dataset provides phoneme labels in ARPABET symbols. To align the phoneme transcriptions with the model's output, the ESpeak phonemization tool is used to convert the provided lyrics in DALI dataset to IPA symbols. This tool was also used in the original work by Xu, Baevski, and Auli [24], ensuring consistency in phoneme transcription.

4.1.3 Running the Model

Access to the Xu, Baevski, and Auli [24]'s model is provided through Huggingface¹, an open-source deep learning framework that offers convenient APIs and tools. The Huggingface Transformers library is utilized to implement and run the baseline model, facilitating the evaluation of phoneme recognition across the processed audio files. This setup ensures that the model runs efficiently and the phoneme recognition results can be accurately compared against the DALI dataset's labels.

4.2 Fine-tuning XLS-R Model

The XLS-R model is available in several variants based on the number of parameters, specifically the 300 million, 1 billion, and 2 billion parameters models². For this study, the 300 million and 2 billion parameters models are selected to evaluate the impact of model size on phoneme recognition accuracy.

¹<https://huggingface.co/>

²<https://github.com/facebookresearch/fairseq/blob/main/examples/wav2vec/xlsr/README.md>

4.2.1 Version Management

During the fine-tuning process, training checkpoints are systematically uploaded to the Hugging Face Hub. This platform integrates version control, which provides a robust safeguard against potential issues during training. If a problem arises, the process can be resumed from the last saved checkpoint, thus avoiding the need to restart the entire training from scratch.

Due to the substantial size of the model checkpoint files, Git Large File Storage (Git-LFS)³ is employed. This extension of Git is designed to handle large files more efficiently, allowing for smoother uploads and management of extensive data within the version-controlled environment provided by the Hugging Face Hub. This approach ensures that all model data remains accessible and securely stored, facilitating ongoing research and development activities.

4.2.2 Data Preprocessing

The DALI dataset, comprising a total of 7,756 songs, provides URLs to corresponding YouTube videos for each song. However, some of these videos are either removed or have restricted access on YouTube. Such inaccessible songs are excluded from the dataset. The remaining songs are downloaded using the Python 'pytube' library, which retrieves the video files in MP4 format.

The XLS-R model, utilized in this study, requires audio files to have a 16 kHz sampling rate. Therefore, the downloaded MP4 files are converted into the Waveform Audio File Format (WAV) at a 16 kHz sampling rate using the Python 'pydub' library.

DALI dataset annotations include timestamps marking the start and end of each sentence within a song's lyrics. Utilizing these timestamps, each full-length audio file is segmented into shorter clips, each containing a single sentence. This segmentation is crucial because the full-length songs, typically lasting 3-4 minutes, are too lengthy for the XLS-R model to process efficiently. The model's self-attention mechanism demands a significant amount of memory, especially for longer input sequences, making the han-

³<https://git-lfs.com/>

ding of such large files impractical.

An analysis (detailed in Table 4.1) reveals a substantial imbalance in the total duration of songs across different languages in the dataset. While it is not feasible to perfectly balance the duration across all languages due to limitations in the available audio files, an effort is made to select approximately 480 minutes of audio per language for model fine-tuning. In cases where a language does not have up to 480 minutes of audio, all available files are used. The dataset is then split into training and testing sets, with 80% of the audio files allocated for training and the remaining 20% for testing from each language.

The prepared datasets are subsequently loaded into the research environment using the Hugging Face Datasets library, ensuring they are ready for the subsequent stages of model training and evaluation. This step facilitates efficient data management and accessibility during the machine learning workflow.

Language	Duration (Minutes)	Language	Duration (Minutes)
English	11171	Hungarian	20
German	1105	Swahili	17
French	513	Croatian	14
Spanish	482	Slovak	14
Italian	477	Danish	10
Polish	159	Slovene	10
Dutch	139	Welsh	7
Finnish	136	Romanian	3
Portuguese	64	Czech	2
Swedish	63	Estonian	2
Norwegian	30	Latin	5
Turkish	22		

Table 4.1: Duration of each language in the DALI dataset (rounded to nearest minute).

4.2.3 Prepare the Tokenizer

The pre-trained XLS-R model outputs a sequence of context representations, which are then processed into actual transcriptions by a tokenizer. For this process, the `Wav2Vec2CTCTokenizer` from the Hugging Face Transformers library is utilized. This tokenizer acts as a linear layer added on top of the XLS-R model's transformer.

The tokenizer's function is to classify the model's output into labels or tokens corresponding to the vocabulary of the transcription dataset. Special characters are removed from the transcriptions, and the text is transformed into lowercase. Additionally, two special tokens, "padding token" and "blank token," are introduced to accommodate the requirements of the `CTC` algorithm.

Every unique character from the dataset's transcriptions is extracted and assigned a unique number. This mapping of numbers to characters or tokens constitutes the vocabulary that is provided to the tokenizer.

4.2.4 Prepare the Feature Extractor

Audio files are continuous signals and need to be sampled before they can be fed into a model. To ensure compatibility with the model's input requirements, a feature extractor is used. Specifically, the `Wav2Vec2FeatureExtractor` from the Hugging Face Transformers library is configured for this task.

The feature size is set to 1 because the inputs are raw audio files. The sampling rate is maintained at 16kHz to match the model's training conditions, as changing the sampling rate can significantly alter the distribution of the audio data. Therefore, it is crucial to use audio input at 16kHz.

For training, batch inference is employed due to the large size of the dataset. This necessitates padding the shorter inputs within a batch to align with the length of the longest input, using a padding value of 0.0.

In the `Wav2Vec2FeatureExtractor`, it is necessary to specify whether the input should undergo zero-mean-unit-variance normalization. This normalization process adjusts the

data so that its mean is zero and its variance is one, a condition that generally improves the performance of speech models. Thus, this normalization is activated.

Additionally, it is essential to determine whether to use an attention mask for batched inference. Given that XLS-R models require the use of an attention mask to perform optimally, this feature is also enabled.

4.2.5 Prepare the Trainer

The Trainer from the Hugging Face Transformers library is also utilized and configured for this task. A specialized data collator is used to dynamically pad the inputs within a batch to match the length of the longest input. This data collator is unique because it processes ‘input_values’ and ‘labels’ differently, applying distinct padding functions to each by leveraging the XLS-R processor’s context management capabilities.

This method is crucial because it acknowledges that the inputs and outputs in speech models represent different types of data, each requiring specific handling. Unlike standard data collators, this approach does not apply uniform padding across different data types. For the labels, padding tokens are assigned a value of -100, ensuring that these tokens are ignored during loss calculations, which is essential for accurate model training.

This method is crucial because it acknowledges that the inputs and outputs in speech models represent different types of data, each requiring specific handling. Unlike standard data collators, this approach does not apply uniform padding across different data types. For the labels, padding tokens are assigned a value of -100, ensuring that these tokens are ignored during loss calculations, which is essential for accurate model training.

The evaluation metric for this phoneme recognition task is defined as **PER**, and the Character Error Rate (**CER**) the ‘jiwer’ Python library, which is similar to **PER**, is utilized for this purpose. The model architecture generates a sequence of logit vectors, labeled as Y_1, Y_2, \dots, Y_m . These vectors result from the function f_θ acting on the input sequence (x_1, \dots, x_n) , where n exceeds m . Each logit vector Y_i contains the log-odds for each term in the predetermined vocabulary, and thus the dimension of Y_i is equal to the size of the vocabulary specified in the configuration.

The focus is primarily on the model’s most accurate predictions, which are identified by applying the argmax function to the logit vectors. The process also includes converting the encoded labels back to their original string representation. This conversion involves replacing instances of the `pad_token_id` with -100, which is critical to ensure that consecutive tokens are not mistakenly merged into a single token, adhering to the principles of `CTC` decoding.

The appropriate pre-trained model checkpoint is specified in the trainer configuration. Since this study utilizes two versions of the XLS-R model, this parameter is adjusted accordingly for each version. The trainer also includes several adjustable parameters such as attention dropout, hidden dropout, and learning rate.

Optimizing these parameters typically requires hyper-parameter tuning. However, due to limited computing resources, conducting extensive tuning with the large datasets and models used in this study is not feasible. Therefore, the parameter values were adopted from a similar previous study. Only three different learning rates were experimentally tested in this study, as detailed in the results section.

The initial part of the XLS-R architecture consists of several `CNN` layers. These layers are designed to extract acoustically significant and contextually unique features from raw speech signals. According to the relevant literature, this segment of the architecture has already undergone extensive pretraining. Consequently, further fine-tuning of these layers is considered unnecessary. Therefore, the `requires_grad` setting for the parameters associated with this feature extraction segment can be set to `False`, indicating that these parameters do not need to be updated during training.

4.2.6 Training

The model with 300 million parameters is fine-tuned using three different learning rates to determine the most suitable one. Once the optimal learning rate is established from the experiments with the 300 million parameters model, it is then applied to fine-tune the larger model with 2 billion parameters. In both instances, the training is conducted over 30 epochs.

The results of these fine-tuning processes are detailed in the results section. The model with 300 million parameters was fine-tuned on a free Kaggle instance which has P100 GPU with 16 GB memory, while the 2 billion parameters model was fine-tuned on a paid instance equipped with two A100 GPUs, each having 45GB of memory.

4.3 Cover Song Identification

All songs across two datasets are transcribed into their phonetic representations using a fine-tuned model, which is readily accessible through the Hugging Face Hub since the model is uploaded there during the fine-tuning process. Due to the limitations of the wav2vec 2.0 model, which cannot process long-duration audio files, each song is divided into 10-second segments. These segments are individually transcribed, and the resulting phonetic transcriptions are then concatenated to form the complete transcription of each song.

To identify cover versions of original songs, each original song transcription is compared to all potential cover song transcriptions using the Levenshtein distance metric. This metric measures the minimum number of edits required to transform one transcription into another, effectively quantifying their similarity. Cover songs are then ranked for each original song based on their Levenshtein distances; a lower distance indicates a closer match and thus a higher rank.

The accuracy of these matches is evaluated using the **MAP** metric, focusing only on the top 10 results. This approach is justified because each dataset contains only one cover song per original song, making it sufficient to consider a limited number of top-ranked matches to assess performance accurately.

4.4 Summary

The chapter provides a comprehensive overview of the procedures and technical configurations used to fine-tune and evaluate the XLS-R model across multiple experiments aimed at improving phoneme recognition. Various datasets, including the DALI dataset,

were processed and prepared for these experiments. Significant attention was given to the management of large audio files and the handling of data through tools like pytube, pydub, and the Hugging Face Transformers library. The fine-tuning process was methodically documented, highlighting the importance of maintaining consistent training conditions, such as the sampling rate and batch processing norms. The latter part of the chapter focuses on the implementation of the cover song identification process, detailing how the phonetic transcriptions of songs are used to compare and rank potential cover versions using the Levenshtein distance metric.

Chapter 5 - Results & Analysis

This chapter presents the outcomes of a study focused on phoneme recognition in singing voices using various models and data sizes, alongside an exploration of cover song identification effectiveness. It details the methodologies, baseline configurations, incremental enhancements, and the comparative performance of different models. Additionally, the analysis delves into the challenges and potential improvements within the context of phoneme recognition and cover song identification across diverse languages and musical genres. Results are systematically documented in tables and figures throughout the chapter.

5.1 Results

This section details the results of a study on phoneme recognition in singing voices using different models and data scales. The baseline is established using the fine-tuned XLSR-53 model, followed by experiments on learning rates and dataset sizes with 300 million and 2 billion parameter models. Additionally, the performance of these models in cover song identification across two datasets is assessed.

5.1.1 Baseline

Xu, Baevski, and Auli [24]’s fine-tuned XLSR-53 model is utilized as the baseline model for this study. Transcriptions generated by the ESpeak phonemicization tool are compared with the predictions from the fine-tuned XLSR-53 model. The evaluation metric used is the **PER**. The overall PER across the DALI test set stands at 0.537 and the Table 5.1 displays the average **PER** for each language, representing the baseline accuracy of multilingual phoneme recognition in singing voice for this study.

Language	PER	Language	PER
English	0.494	Hungarian	0.695
German	0.439	Swahili	0.599
French	0.461	Croatian	0.425
Spanish	0.436	Slovak	0.420
Italian	0.439	Danish	0.496
Polish	0.504	Slovene	0.529
Dutch	0.422	Welsh	0.6
Finnish	0.469	Romanian	0.634
Portuguese	0.473	Czech	0.676
Swedish	0.617	Estonian	0.582
Norwegian	0.589	Latin	0.772
Turkish	0.586		

Table 5.1: Average **PER** for each language in baseline model.

5.1.2 Fine-tuned XLS-R Model

In an initial phase of the study, three different learning rates are tested to fine-tune a model with 300 million parameters, as the larger 2 billion parameters model required more computing power than is available on the free tier. The learning rates tested are 5×10^{-3} , 5×10^{-4} , and 5×10^{-5} . The lowest learning rate, 5×10^{-5} , produced the best results and is subsequently chosen for all further experiments.

Following this, the 300 million parameters model underwent further training using varying amounts of data. In the initial training session, the model is fine-tuned using 100 minutes of audio from each language, incorporating all available data from languages that had less than 100 minutes of audio. The phoneme **PER** for this setup is 0.7088. In a subsequent run, the amount of data is increased to 240 minutes of audio per language, again including all available data from languages with fewer resources. This led to a notable improvement in **PER**, which dropped to 0.3870.

For the final experiment, the 2 billion parameters model is fine-tuned using an even larger dataset consisting of 480 minutes of audio from each language, including all data from

languages with limited resources. This adjustment resulted in a further reduction in the **PER**, achieving a rate of 0.32 (Table 5.2).

Model Size (parameters)	Data per Language (minutes)	PER
300 million	100	0.7088
300 million	240	0.3870
2 billion	480	0.32

Table 5.2: Results of fine-tuned XLS-R model.

5.1.3 Cover Song Identification

For the Covers80 dataset, which comprises exclusively English songs, the **MAP** achieved is 0.745. In contrast, the **MAP** for the Sinhala Cover Songs dataset is 0.513 (Table 5.3).

Dataset	MAP
Covers80 (English songs)	0.745
Sinhala Cover Songs	0.513

Table 5.3: Cover song identification results.

5.2 Analysis

The Analysis section delves deeper into the nuances of the study, examining the factors contributing to the model’s performance. This section discusses the challenges in phoneme recognition, particularly the distinction between vowels and consonants in a singing context, and the impact of background music. The section also explores the implications of using global versus local similarity measures for cover song identification and addresses the limitations posed by the model’s handling of long audio files. The discussion highlights the potential for future improvements in model accuracy through enhanced training strategies and resource allocation.

5.2.1 Fine-tuned XLS-R Model

The fine-tuned model demonstrates a notable enhancement in transcription accuracy compared to the baseline. The overall **PER** across the DALI test set improved significantly from 0.537 to 0.32, representing a 40.41% improvement in multilingual phoneme recognition of singing voice. Table 5.4 provides a detailed comparison, listing the average **PER** for each language using both the baseline and the fine-tuned models, alongside the percentage improvement achieved over the baseline model.

In Table 5.4, it's clear that low-resource languages have seen greater improvements than high-resource languages. This difference can be attributed to the music genres associated with these languages. Most songs in low-resource languages fall under genres like instrumental music, where the background sound is less intense. Genre-based analysis further supports this observation: accuracy tends to be lower in genres with louder music, such as rock and hard rock (Figure 5.1). Despite this, there is still noticeable improvement over the baseline model.

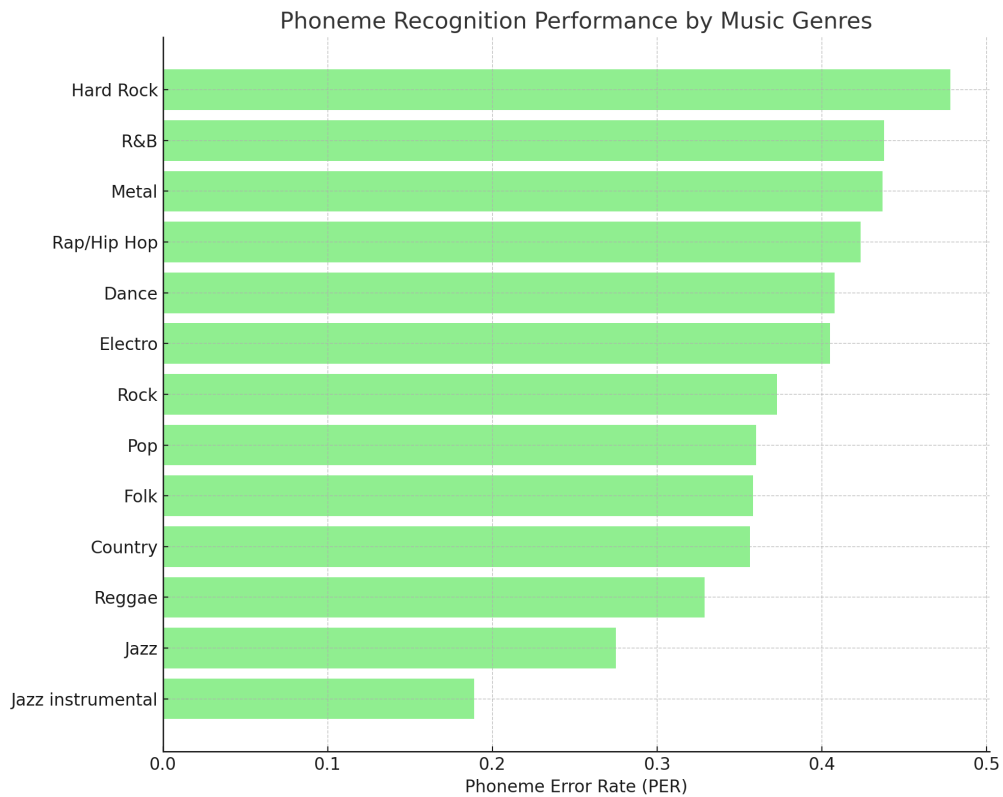


Figure 5.1: Phoneme recognition performance by music genres

Language	Baseline PER	Fine-tuned PER	Improvement (%)
English	0.494	0.334	32.39
German	0.439	0.344	21.64
French	0.461	0.39	15.40
Spanish	0.436	0.352	19.27
Italian	0.439	0.313	28.70
Polish	0.504	0.286	43.25
Dutch	0.422	0.207	50.95
Finnish	0.469	0.192	59.06
Portuguese	0.473	0.217	54.12
Swedish	0.617	0.303	50.89
Norwegian	0.589	0.270	54.16
Turkish	0.586	0.217	62.97
Hungarian	0.695	0.33	52.52
Swahili	0.599	0.285	52.42
Croatian	0.425	0.302	28.94
Slovak	0.420	0.314	25.24
Slovene	0.529	0.34	35.73
Welsh	0.600	0.252	58.00
Romanian	0.634	0.194	69.40
Czech	0.676	0.431	36.24
Estonian	0.582	0.416	28.52
Latin	0.772	0.593	23.19

Table 5.4: Phoneme Error Rate (PER) improvement for each language, comparing baseline and fine-tuned models.

In an analysis of the DALI test set, comparing predicted transcripts against reference transcripts, it was observed that the model accurately identifies consonants more often than vowels. This distinction is significant in phonetics, where speech sounds are categorized primarily into vowels and consonants. Vowels are vocal sounds produced without any significant closure of the air passage and are crucial for syllable formation; they include sounds like 'AH,' 'AA,' 'AE,' 'EY,' 'IH.' Consonants, by contrast, involve some degree of closure or constriction within the vocal tract, such as 'K,' 'D,' 'T,' 'P,' 'B.'

The model’s poorer performance in identifying vowels can be attributed to the complex variations in the singing voice, which differ significantly from normal speech. In singing, the rate of speech—especially the elongation and modulation of vowel sounds—is markedly different, which poses a challenge for accurate recognition. Additionally, loud background music often accompanies singing, further complicating the accurate discernment of vowel sounds.

The results of this study indicate that phoneme recognition accuracy in singing voices has improved significantly when compared to a baseline speech recognition model. The speech recognition model used was somewhat adapted to handle singing voices, though there remains substantial potential for enhancement.

In this research, hyperparameter tuning was not conducted during the training phase due to resource constraints. However, with adequate resources, conducting extensive hyperparameter tuning could likely lead to further improvements in model accuracy.

5.3 Cover Song Identification

The performance of our cover song identification system is subpar when compared with state-of-the-art lyric bases cover song identification system. Specifically, on the Cover80 dataset, our system achieved a **MAP** of only 0.745. This result is lower than those obtained by state-of-the-art cover song identification systems, as detailed in Table 5.5.

System	MAP
Correya, Hennequin, and Arcos [4]	0.926
Vaglio, Hennequin, Moussallam, <i>et al.</i> [7]	0.790
Proposed Method	0.745

Table 5.5: Comparison of English cover song identification results using the proposed method and two other methods.

The results from the Correya, Hennequin, and Arcos [4]’s study can be disregarded since it relies on a lyrics-informed system, where lyrics are manually inputted into the system. In contrast, the Vaglio, Hennequin, Moussallam, *et al.* [7]’s study employs a

SVR framework to automatically transcribe lyrics, and it demonstrated better results for English songs compared to the proposed method in this study. However, their system is specifically tailored for English, indicating a dependency on language.

The inaccuracies were primarily due to the accuracy of phoneme recognition. Another contributing factor was the presence of instrumental sections in songs, which do not contain any vocals. The model attempted to transcribe these purely musical parts as well, resulting in some nonsensical transcriptions. Implementing a singing voice identification method could resolve this issue by ensuring the model only transcribes sections where a singing voice is detected.

The chosen similarity measure, despite being the most suitable one available, has some limitations. The Levenshtein distance is a global similarity metric, which assesses the entire string to determine similarity, rather than focusing on localized or substring similarities. This approach posed problems when comparing songs whose cover versions are significantly shorter than the original versions, often because the original versions include longer instrumental sections. Consequently, when the model transcribes these instrumental parts, the original version ends up with a much longer phonetic transcription, which increases the similarity distance between the two versions. A potential solution is to use local similarity metrics that assess only portions of the strings. These metrics, similar to those used in local sequence alignment in bioinformatics, could provide a more accurate measure of similarity in such cases.

The model struggled with handling longer audio files, leading to the transcription of songs into phonemes in a chunk-wise manner, with the audio split into 10-second long chunks. This approach resulted in transcription inaccuracies at the points where the audio was split, contributing to lower accuracy in cover song identification. To improve accuracy, a singing voice identification method could be used to strategically split the song into chunks at points where only instrumental music is present, avoiding the disruption of vocal segments.

The proposed method performs poorly compared to other state-of-the-art lyrics-based cover song identification methods. However, it is important to note that these other systems are language-dependent. The proposed model, evaluated using a dataset of Sin-

hala songs, still managed to achieve reasonable results despite Sinhala being an unseen language during both the training and fine-tuning phases. A comprehensive analysis of multilingual lyrics-based cover song identification is currently limited due to the scarcity of cover song datasets in various languages. Nonetheless, the success achieved with an unseen language highlights the potential for multilingual cover song identification using the proposed phoneme-based method, which still has considerable room for improvement.

5.4 Summary

This chapter meticulously evaluates the performance of a phoneme-based model for cover song identification, comparing it against other state-of-the-art methods. It presents detailed results from various experiments, showing the model's capabilities and limitations in handling multilingual datasets and different audio complexities. The analysis further investigates the specific issues affecting accuracy, such as phoneme recognition and audio segmentation, and suggests potential solutions for these challenges. Overall, the chapter emphasizes the model's potential in multilingual settings and outlines areas for further research and development to enhance its effectiveness in real-world applications.

Chapter 6 - Limitations & Future Works

This study introduces a novel method for cover song identification using phoneme transcriptions of singing voices, but several limitations have emerged that require future exploration. One significant issue is the model's inherent limitation in processing long-duration audio files. This constraint necessitates dividing the audio into shorter segments, leading to inaccuracies in the final transcriptions, as detailed in Chapter 5. Future research could focus on developing models capable of handling extensive audio files or devising robust methods to mitigate inaccuracies in existing models.

Another major limitation is the absence of multilingual datasets to assess the accuracy of multilingual cover song identification. To address this, there is a need to create new datasets that include multilingual cover song data. Such datasets would significantly enhance the model's training and testing phases, providing a broader base for evaluating the effectiveness of the phoneme recognition system across different languages.

While the Levenshtein distance is currently the most suitable method for measuring similarities in this context, it has limitations that affect its efficacy, particularly in cover song identification where nuanced differences are crucial. A potential advancement would be the development of a Weighted Levenshtein Distance. This metric would extend the standard Levenshtein Distance by assigning variable costs to different operations—such as insertions, deletions, and substitutions—based on the phonetic similarity between phonemes. Implementing such a customization would improve the sensitivity and accuracy of the metric, especially in scenarios where phonetic nuances significantly influence the outcome.

Furthermore, the study did not engage in extensive hyperparameter tuning of the XLS-R model due to resource constraints. Future efforts could focus on comprehensive hyperparameter optimization to enhance the model's accuracy in multilingual phoneme recognition. Optimizing parameters such as learning rate, batch size, and model architecture could lead to substantial improvements in performance. This approach would not only refine the model's ability to handle diverse linguistic nuances but also extend its applicability to a wider range of audio processing tasks.

In summary, while this study lays the groundwork for advanced phoneme recognition in singing voices, it also highlights several avenues for further research and development. By addressing these limitations, future studies can significantly enhance the robustness and accuracy of phoneme recognition systems, particularly in complex and multilingual audio environments.

Chapter 7 - Conclusions

This research commenced with the hypothesis that utilizing phoneme transcriptions of singing voices could facilitate multilingual lyrics-based cover song identification, thereby eliminating language dependency from the identification process. The primary motivation for focusing on lyrics-based identification is its scalability and the ability to recognize covers even when the music has been significantly altered but the lyrics remain unchanged.

The first step was to identify a suitable [ASR](#) system that could be adapted to transcribe singing into phoneme transcriptions. A comprehensive literature review revealed that transformer-based wav2vec2 models exhibit state-of-the-art phoneme recognition performance in natural speech. Among several wav2vec2 models evaluated, Xu et al.'s fine-tuned XLSR-53 model was initially considered as the baseline for this study due to its superior performance in multilingual phoneme recognition. However, this model did not perform satisfactorily on singing voices for the purposes of cover song identification. Consequently, the more advanced XLS-R model, which outperforms XLSR-53, was selected for this study. The DALI dataset was used to fine-tune the XLS-R model, significantly enhancing its phoneme recognition capabilities in singing voices.

For cover song identification using phoneme transcriptions, a robust similarity measure that considers the order of tokens or characters is essential. The Levenshtein distance was chosen for this purpose due to its suitability for the task. This metric was then applied to assess cover song identification accuracy using two datasets: Covers80 and Sinhala Cover Songs.

Despite these efforts, the cover song identification accuracy did not reach the expected levels compared to state-of-the-art methods. The proposed method performed well with the Covers80 dataset and reasonably well with the Sinhala Songs dataset. The modest success with the Sinhala dataset is particularly notable because Sinhala was an unseen language for the model both during its pretraining and fine-tuning phases. This outcome demonstrates that the research has effectively addressed the initial hypothesis by removing language dependency. However, significant improvements are needed to enhance the

efficacy of the proposed method for practical cover song identification applications.

In conclusion, this research has made a significant contribution to the field of lyrics-based cover song identification by demonstrating the potential of phoneme transcription to mitigate language dependencies. By utilizing advanced ASR technology such as the XLS-R model and integrating phoneme recognition in singing voices, this study has laid the groundwork for more language-agnostic approaches in music identification. While the results with the Sinhala dataset underscore the challenges of applying this technology across diverse linguistic contexts, they also highlight the innovative direction of using phoneme transcriptions for music recognition. Future work will focus on refining the phoneme transcription techniques and improving the robustness of the similarity measures used, aiming to enhance the practical applicability of this method for cover song identification across a broader range of languages and musical styles.

Bibliography

- [1] *IFPI GLOBAL MUSIC REPORT 2023*, Online. [Online]. Available: <https://globalmusicreport.ifpi.org/>.
- [2] J. Serrà, E. Gómez, and P. Herrera, “Audio cover song identification and similarity: Background, approaches, evaluation, and beyond,” in *Advances in Music Information Retrieval*, Z. W. Raś and A. A. Wiczorkowska, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 307–332, ISBN: 978-3-642-11674-2. DOI: [10.1007/978-3-642-11674-2_14](https://doi.org/10.1007/978-3-642-11674-2_14). [Online]. Available: https://doi.org/10.1007/978-3-642-11674-2_14.
- [3] E. Liebman and P. Stone, *Artificial musical intelligence: A survey*, 2020. arXiv: [2006.10553](https://arxiv.org/abs/2006.10553) [cs.SD].
- [4] A. A. Correya, R. Hennequin, and M. Arcos, “Large-scale cover song detection in digital music libraries using metadata, lyrics and audio features,” *ArXiv*, vol. abs/1808.10351, 2018.
- [5] H. Schutze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*. Cambridge University Press, 2008.
- [6] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.
- [7] A. Vaglio, R. Hennequin, M. Moussallam, and G. Richard, “The words remain the same: Cover detection with lyrics transcription,” in *22nd International Society for Music Information Retrieval Conference ISMIR 2021*, 2021.
- [8] C. Gupta, E. Yilmaz, and H. Li, *Automatic lyrics alignment and transcription in polyphonic music: Does background music help?* 2019. arXiv: [1909.10200](https://arxiv.org/abs/1909.10200) [eess.AS].
- [9] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, “Creating dali, a large dataset of synchronized audio, lyrics, and notes,” *Trans. Int. Soc. Music. Inf. Retr.*, vol. 3, pp. 55–67, 2020.
- [10] F. Yesiler, C. J. Tralie, A. A. Correya, *et al.*, “Da-tacos: A dataset for cover song identification and understanding,” in *International Society for Music Information Retrieval Conference*, 2019.
- [11] X. Du, Z. Yu, B. Zhu, X. Chen, and Z. Ma, “Bytecover: Cover song identification via multi-loss training,” *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 551–555, 2020.
- [12] I. P. Association, *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.
- [13] J. K. Hansen, “Recognition of phonemes in a-cappella recordings using temporal patterns and mel frequency cepstral coefficients,” 2012. [Online]. Available: <https://api.semanticscholar.org/CorpusID:37419482>.

- [14] A. Mesaros and T. Virtanen, “Recognition of phonemes and words in singing,” *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2146–2149, 2010. [Online]. Available: <https://api.semanticscholar.org/CorpusID:2809611>.
- [15] M. Grühne, C. Dittmar, and K. Schmidt, “Phoneme recognition in popular music,” in *International Society for Music Information Retrieval Conference*, 2007. [Online]. Available: <https://api.semanticscholar.org/CorpusID:15841983>.
- [16] J. S. Garofolo, L. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, “Darpa timit acoustic-phonetic continuous speech corpus cd-rom {timit} — nist,” 1993. [Online]. Available: <https://api.semanticscholar.org/CorpusID:65148724>.
- [17] M. Ravanelli, T. Parcollet, and Y. Bengio, “The pytorch-kaldi speech recognition toolkit,” *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6465–6469, 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:53790579>.
- [18] S. Schneider, A. Baevski, R. Collobert, and M. Auli, “Wav2vec: Unsupervised pre-training for speech recognition,” in *Interspeech*, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:119188316>.
- [19] A. Baevski, S. Schneider, and M. Auli, “Vq-wav2vec: Self-supervised learning of discrete speech representations,” *ArXiv*, vol. abs/1910.05453, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:204512445>.
- [20] E. Jang, S. S. Gu, and B. Poole, “Categorical reparameterization with gumbel-softmax,” *ArXiv*, vol. abs/1611.01144, 2016. [Online]. Available: <https://api.semanticscholar.org/CorpusID:2428314>.
- [21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *North American Chapter of the Association for Computational Linguistics*, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:52967399>.
- [22] A. Baevski, H. Zhou, A.-r. Mohamed, and M. Auli, “Wav2vec 2.0: A framework for self-supervised learning of speech representations,” *ArXiv*, vol. abs/2006.11477, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:219966759>.
- [23] A. Conneau, A. Baevski, R. Collobert, A.-r. Mohamed, and M. Auli, “Unsupervised cross-lingual representation learning for speech recognition,” in *Interspeech*, 2020. [Online]. Available: <https://api.semanticscholar.org/CorpusID:220055837>.
- [24] Q. Xu, A. Baevski, and M. Auli, “Simple and effective zero-shot cross-lingual phoneme recognition,” *ArXiv*, vol. abs/2109.11680, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:237635125>.
- [25] A. Babu, C. Wang, A. Tjandra, *et al.*, “Xls-r: Self-supervised cross-lingual speech representation learning at scale,” in *Interspeech*, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:244270531>.
- [26] J. F. Hair, A. H. Money, P. Samouel, and M. Page, “Research methods for business,” *Education+ Training*, vol. 49, no. 4, pp. 336–337, 2007.

- [27] A. Melnikovas, “Towards an explicit research methodology: Adapting research onion model for futures studies,” *Journal of futures Studies*, vol. 23, no. 2, pp. 29–44, 2018.
- [28] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, “Creating dali, a large dataset of synchronized audio, lyrics, and notes,” *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, 2020.
- [29] D. P. W. Ellis, *The ”covers80” cover song data set*, <http://labrosa.ee.columbia.edu/projects/coversongs/covers80/>, Accessed: [insert date here], 2007.
- [30] P. Wijesena, L. Jayaratne, M. Wickramasinghe, S. Abeytunge, and P. Marasinghe, “Metric learning with sequence-to-sequence autoencoder for content-based music identification,” *ITM Web of Conferences*, vol. 60, Jan. 2024. DOI: [10.1051/itmconf/20246000007](https://doi.org/10.1051/itmconf/20246000007).
- [31] A. Hannun, “Sequence modeling with ctc,” *Distill*, 2017, <https://distill.pub/2017/ctc>. DOI: [10.23915/distill.00008](https://doi.org/10.23915/distill.00008).
- [32] W. Heeringa, “Measuring dialect pronunciation differences using levenshtein distance,” 2004. [Online]. Available: <https://api.semanticscholar.org/CorpusID:61144415>.