Birzeit University

Faculty of Engineering and Technology

Department of Electrical and Computer Engineering

Language Code-Switching Identification

Prepared by:

1200814 - Masa Itmaiza

1201619 - Hamza Awashra

1201467 - Jehad Halahla

Supervised by:

Dr. Abualseoud Hanani

A graduation project submitted to the Department of Electrical and Computer Engineering in partial fulfillment of the requirements for the degree of B.Sc. in Computer Engineering.

Birzeit

Jan - 2025

Abstract

The widespread prevalence of multilingualism in today's globalized world has increased the need for advanced speech processing systems capable of handling complex linguistic phenomena like code-switching. Code-switching, the practice of alternating between languages within a single conversation or utterance, poses significant challenges for Automated Speech Recognition systems, which are traditionally designed for monolingual speech. This project addresses these challenges by developing an AI-based language diarization model tailored for English-Arabic code-switched speech. By integrating state-of-the-art technologies, such as Whisper and WhisperX models for transcription and alignment and the speechbrain/lang-id-voxlingua107-ecapa model for language identification, this system aims to accurately segment and identify language boundaries within speech data. The methodology consists of audio preprocessing, transcription and word-level alignment, language identification for each segment, and generation of Rich Transcription Time Marked (RTTM) outputs. Potential applications of this work include enhancing ASR systems, virtual assistants, and subtitle generation tools, particularly in code-switched contexts. The outcomes will be benchmarked against established metrics to ensure performance and robustness, aiming to contribute to the broader field of multilingual speech technologies. This system achieves a Diarization Error Rate (DER) of **37.85%** using the pretrained VoxLingua107-ECAPA model and **40.09%** with a custom-trained model, demonstrating competitive performance in English-Arabic code-switching scenarios.

Abstract (Arabic)

أدى الانتشار الواسع للتعدد اللغوي في عالمنا اليوم إلى تزايد الحاجة إلى أنظمة متقدمة لمعالجة الكلام، تكون قادرة على التعامل مع الظواهر اللغوية المعقدة مثل التناوب بين اللغات، وهو التبديل بين لغتين أثناء الحديث. يشكل هذا التحدي عائقًا كبيرًا أمام أنظمة التعرف التلقائي على الكلام التي صُممت في الأصل للتعامل مع لغة واحدة فقط.

يهدف هذا المشروع إلى معالجة هذه التحديات من خلال تطوير نموذج ذكي لتحديد اللغة في الكلام المتناوب بين العربية والإنجليزية. يعتمد النظام على تقنيات حديثة مثل نماذج ويسبر و ويسبر إكس لعمليات النسخ والمحاذاة الزمنية، بالإضافة إلى نموذج سبيتش برين المخصص لتحديد اللغة، بهدف تقسيم الكلام بدقة وتحديد الحدود اللغوية بين المقاطع المختلفة.

تتضمن المنهجية خطوات تشمل: معالجة الصوت الأولية، التفريغ الصوتي، المحاذاة على مستوى الكلمات، تحديد اللغة لكل مقطع، وإنتاج ملفات التوصيف الزمني للنص.

تشمل التطبيقات المحتملة لهذا العمل تحسين أنظمة التعرف على الكلام، والمساعدات الصوتية الذكية، وأدوات توليد الترجمة النصية، خصوصًا في البيئات التي تتضمن تناوبًا لغويًا. تم تقييم النظام باستخدام مقاييس أداء معروفة لضمان الكفاءة والموثوقية.

، في حين سجل النموذج 37.85% حقق النموذج القائم على فوكس لينغوا معدل خطأ بلغ
، مما يُظهر قدرة النظام على التعامل مع التناوب 40.09% المُدرّب محليًا معدل خطأ قدره
اللغوي بين العربية والإنجليزية، مع إمكانية تحسينه مستقبلاً.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation and Overview

Languages have always been an integral part of human cultures and civilizations. Language is not only a means of communication, but it is also one of the important elements that gives people their sense of identity and authenticity.

Even though languages are defined by rules, this does not mean they are static, rather, they evolve over time. Some languages are very old, and have hardly changed at all, other languages evolve rapidly by incorporating elements of other languages. Still, many languages have died out, many dying languages live on in the vocabularies and dialects of prominent languages around the world.

Since the old times, different people spoke different languages and/or dialects, and when people with differing languages needed to communicate with each other, they needed to learn a different language, or at least, they needed to learn sufficiently enough of that language in order to understand the other. Therefore, this language contact between populations have brought to us what is know as multilingualism, which is defined as the act of using more than one language in communication between individuals or groups [1].

Multilingualism is a natural outcome that developed in human societies for many reasons: discontinuous or prolonged language contact between populations, such as trade, conquest, travel, intermarriage, or mediated means, like an interest in studying written books. Such situations made it possible for speakers or entire groups to be able to understand, speak, or write, several languages, in varying degrees of proficiency [2].

Throughout history, certain languages have held dominant roles in fields like science, philosophy, and technology. This dominance has often shifted due to geopolitical and cultural shifts, alongside practical needs. For example, Latin was once the prevailing language of scientific knowledge in medieval Europe. As the Roman Empire spread, Latin became essential for scholars, who sought to access and contribute to its wealth of accumulated knowledge. Centuries later, Arabic rose to prominence, largely due to the Islamic Golden Age, when scholars in the Arab world translated, preserved, and expanded on scientific knowledge from Greek, Persian, and Indian texts, making Arabic essential for scientific and scholarly pursuits.

Today, English holds a similar status, having become the global lingua franca* in major domains such as science, technology, business, and entertainment. English's dominance is driven by historical colonial influences and modern economic and technological leadership by English-speaking nations. Accessing and contributing to the most current resources often requires proficiency in English, making it essential for professionals and scholars worldwide.

However, multilingualism today presents a new layer of complexity. Individuals in multilingual societies often speak multiple languages with varying degrees of fluency, based on personal background, educational experiences, and practical needs. This variation shapes how people communicate and use their languages. For example, in conversations, many multilinguals might alternate between languages within a single utterance—a phenomenon known as code-switching. Unlike simply translating, code-switching involves seamlessly shifting between languages within the same discourse, sentence, or even word.

Code-switching does not necessarily imply full fluency in both languages but is often used to convey nuanced meanings, express cultural identity, or even bridge linguistic gaps in mixed-language conversations. It's a flexible, adaptive strategy that many bilinguals and multilinguals employ in daily interactions, reflecting their complex, hybrid linguistic environments. This behavior is increasingly prevalent in globalized societies, as more people have regular exposure to multiple languages in their social, academic, and professional lives.

The phenomenon of multilingualism exists in abundance, especially in places where there exists more than one culture in a single geographical locations, or in places that experienced colonialism over an extended period of time, or it is practiced normally by people who work/live in a foreign country [1]. There exists many examples of multilingual environments, such as Algerian Arabic-French multilingualism, and Palestinian Arabic-Hebrew multilingualism, which existed as a natural outcome of prolonged colonialism.

Today, multilingualism drives modern technology and academics to globalize their industries. The rise of the phenomenon drives countries to prepare for such globalization by teaching them a wide repertoire of languages. It is also well known that learning new languages provides a considerable improvement in cognitive abilities and can help hone logical thinking by providing a new perspective in the thinking process [3].

This project aims to develop an AI-based language diarization model that can accurately segment code-switched speech, focusing primarily on English-Arabic conversations.this project may also enhance the performance of ASR systems by improving their ability to process multilingual speech. The research will benchmark the model's results against baseline metrics from the literature to ensure acceptable performance and usability.

This work addresses an urgent need for better multilingual speech processing tools, especially in regions where code-switching is prevalent, to enable seamless communication and bridge the gap between monolingual and multilingual speech technologies.

1.2 Code Switching (CS)

Code switching (CS from now on) is the ability to alternate between languages in an unchanged setting, often within the same utterance. It is a natural result of multilingualism and regular language contact. This linguistic manifestation might range from the insertion of single words to language alternation for broader parts of discourse. Bilinguals with varying levels of ability and language contact contexts may develop non-uniform communication styles. CS can be used for several purposes, including filling linguistic gaps, expressing ethnic identity, and achieving specific discursive goals.

Example of Code Switching in an Arabic-Hebrew Context

One specific example of code switching from an Arabic-Hebrew CS context:

- بدك تيجي معي عالسوق؟
- أُني صريك لقنوت دبريم لبيت.
- طيب، بس بسرعة عشان عندي شغل.

tranlsation:

- "Do you want to come with me to the market?"
- "I need to buy some things for the house."
- "Okay, but quickly because I have work."

also in a single utterance:

١- بدي أروح ألعب كرة القدم، بأل همشخق متخيل مؤخر.

٢- خلص، أنا رح أدرس هلا، وبعدين نلك لبيت قپه.

٣- شو هالبلقان.

translation:

1. "I want to go play football, but the game starts late."
2. "Okay, I'll study now, and then we'll go to a café."
3. what is this mess [4].

Types of Code Switching

There is a debate in the literature on the characterization of code-switching types [4] since it is not a trivial task. However, there are four major types:

1. **Inter-sentential switching**: also known as "extra-sentential" switching, happens between sentences or clauses at their boundaries.
2. **Intra-sentential switching**: occurs within a sentence or clause.
3. **Tag-switching**: involves switching between a tag phrase, a word, or both.
4. **Intra-word switching**: occurs at morpheme boundaries within a single word.

1.3 Impact of CS on Automatic Speech Recognition

Most existing ASR systems are designed primarily for monolingual speech, struggling to deliver acceptable performance when faced with multilingual conversations. This creates communication barriers and limits the usability of ASR technology in such contexts.

The current lack of effective language diarization models, which can accurately segment audio into different language regions, presents a significant challenge for processing code-switched speech. In particular, there is a need for a diarization model that can identify and handle the complexities of English-Arabic speech and potentially other language pairs like Arabic-Hebrew, depending on available datasets.

1.4 Problem Statement

The existence of code-switching presents challenges not only for the normal everyday exchange of speech between individuals, but also for modern systems that are concerned with spoken languages.

usually in any exchange between people, there is a certain level of ambiguity in understanding presented by the linguistic gap that code-switching introduces, for example, a couple of individuals can have different levels of comprehension of the languages that are present in the exchange, and it is not necessary that they both share the same linguistic background, which implies that one of them at least may not even have one of the languages that are present in the code-switch pair of languages that the other person uses, and even if that is not the case, the couple of individuals will have a varying level of comprehension for both languages even if they share them.

As for most automated speech processing systems today, they also seem to face their own problems regarding code-switched speech, since most of them are trained and designed to work with speech consisting of one language only, and face a degradation in performance when dealing with speech involving more than one language in the same utterance.

Because of the above reasons and more, the need for a model that can determine change in language in a speech segment becomes more and more essential to help foster communication between individuals from different linguistic backgrounds and variable levels of fluency in the languages being used in such social interactions. Such a model can help ASR models to deal with code-switched speech and boost their performance in such contexts.

1.5 Importance and possible applications

The significance of this project lies in the growing demand for language diarization models, which can enhance communication across various domains, including education, entertainment, and general interactions. Potential applications include virtual assistants, subtitle generation tools, and ASR systems integrated with diarization capabilities. Integrating a diarization system into such models can help making these models more powerful and flexible by being able to identify the language in each segment and generating a response accordingly.

1.6 Project Objectives

The main objectives of this project are listed below:

- Provide a comprehensive and up-to-date review of the available research and literature concerning code-switching and Language Diarization.
- Evaluate methodologies, challenges, and advancements in Language Diarization.
- Systematically analyze available datasets to decide which serves as a reliable training and evaluation set for the proposed system, or create a dataset that satisfies the said criteria.
- Investigate the most recent spoken language processing (SLP) and machine learning (ML) approaches for language diarization challenges. and Adapt these strategies to satisfy the needs of Arabic-English code-switching diarization.
- Create a Language Diarization system for Arabic-English code-switched speech with acceptable accuracy, based on the techniques discussed in the literature review.
- Assess the performance of the developed system using conventional metrics such as Equal Error Rate (EER) and Balanced Accuracy (BAC) that are used by the majority of similar systems to give a realistic result.
- Refine and optimize the developed system based on the evaluation results.

2 Related Work

This section outlines various studies and advancements in *LD*, a field that has garnered significant interest in recent years. Researchers have explored several approaches to automatically diarize code-switched speech, including leveraging pre-trained deep learning models, extracting bottleneck features, and ensembling neural networks with similar architectures. These innovative techniques aim to capture the phonotactic and acoustic variations inherent in code-switched speech, driving progress in spoken language processing and enhancing the overall effectiveness of language diarization systems.

2.1 Integration of Overlap Awareness in Speaker Diarization

Speaker diarization has traditionally faced challenges in handling overlapping speech, a common scenario in multi-speaker environments such as meetings and conversations. Conventional systems primarily operated under the assumption of single-speaker dominance, which often led to degraded performance in overlap-heavy datasets.

Recent innovations, such as the **DOVER-Lap** algorithm [5], [6], have shifted focus to overlap-aware methodologies. Building on the DOVER algorithm, DOVER-Lap introduces global optimization in label mapping through weighted k-partite graph matching, thereby improving the alignment of diarization hypotheses from multiple systems. This approach demonstrated consistent improvements in Diarization Error Rate on AMI and LibriCSS datasets, particularly in overlapping regions [6].

Key experiments revealed the effectiveness of this method when integrating outputs from clustering-based, region proposal networks, and target-speaker voice activity detection systems. Despite its promise, DOVER-Lap relies on heuristics that could benefit from further refinement, such as improving label voting mechanisms for mixed single-speaker and overlapping scenarios.

Beyond the algorithmic adjustments, the inclusion of multi-resolution analysis techniques has further refined overlap-aware diarization. For instance, the Whisper-based system evaluated by Vachhani et al [5] integrated accent detection as a mechanism to refine diarization outputs. This technique narrowed potential language hypotheses, reducing false positives and improving overall performance. Such advancements not only enhance overlap handling but also emphasize the importance of tailoring diarization systems for real-world scenarios involving varying speaker densities and accents.

2.2 Leveraging Self-Supervised Learning and Feature Representations

Advances in Self-Supervised Learning (SSL) frameworks have played a pivotal role in enhancing Language Identification (LID) and diarization systems. The adoption of wav2vec2.0 [7] , exemplifies this shift. By fine-tuning wav2vec2.0 for phonological feature detection—such as manner and place of articulation—researchers achieved significant gains in LID accuracy for code-switched English-Mandarin speech. The approach leverages speech attributes as a low-level representation, serving as an effective initialization for downstream classification tasks. This methodology performed exceptionally well on short-duration utterances, with balanced accuracy reaching 81.3% and an equal error rate of 10.6% in the MERLIon CCS challenge [8].

Similarly, Gupta et al [9] proposed a lightweight architecture optimized for compute-constrained environments. This two-stage Encoder-Decoder-based model incorporated depth-wise separable convolutions and squeeze-and-excitation layers, offering a compact solution without compromising accuracy. The model, designed for streaming inference, achieved competitive EERs on closed (15.6%) and open (11.1%) tracks, underscoring the adaptability of SSL-based models for real-time applications. These findings highlight the dual benefits of leveraging SSL: reducing model complexity while retaining robustness against challenging speech patterns like accent variability and code-switching.

2.3 Fusion Strategies and Multi-Resolution Approaches

Fusion techniques have emerged as a cornerstone in improving diarization and identification systems. Ensemble methods, such as those used in the DOVER-Lap algorithm, combine outputs from diverse systems, leveraging complementary strengths to enhance overall performance. Vachhani et al [5] extended this idea by integrating multi-resolution diarization hypotheses, achieving an absolute improvement of 11.66% in DER over baseline systems. The use of varying temporal resolutions enabled finer granularity in capturing speaker transitions and overlaps, addressing the limitations of single-resolution approaches.

The integration of pre-trained models into these frameworks has further amplified their effectiveness. For example, Shahin et al utilized phonological features alongside wav2vec2.0 embeddings to improve performance, while Gupta et al combined models trained on multilingual datasets with curated domain-specific data to adapt to regional variations in speech. This hybrid approach, blending pre-training and fine-tuning, aligns with broader trends in machine learning, where general-purpose models are adapted to niche applications through targeted training [7]gupta2023Spoken.

2.4 Challenges in Code-Switched and Multilingual Speech

Code-switching presents unique challenges for diarization and identification systems, especially in multilingual contexts. The MERLIon CCS challenge dataset [8], featuring English-Mandarin child-directed speech, exemplifies these difficulties. Spontaneous and short-duration utterances, coupled with non-standard accents, necessitate systems capable of precise segmentation and language detection. The dataset revealed a critical gap: existing models struggled with short utterances and rapid language transitions, prompting researchers to explore novel architectures.

Spoorthy et al [10] addressed these challenges with an SVM-based language diarizer tailored for bilingual Indian speech. By leveraging bottleneck features derived from neural networks trained on monolingual English speech, their system effectively identified code-switch points and segmented speech into homogeneous language segments. This approach underscores the potential of feature-based methods in environments where traditional acoustic models fail due to dataset limitations.

2.5 Practical Implications and Future Directions

The reviewed advancements highlight significant progress in diarization and language identification, yet challenges remain. Many systems exhibit limitations when transitioning from controlled datasets to real-world environments. For example, DOVER-Lap performs well in structured datasets but may struggle with high speaker densities and complex acoustic conditions. Similarly, systems optimized for short utterances, like those in the MERLIon CCS challenge, need further testing in less constrained scenarios.

Future research should prioritize:

- **Enhanced Overlap Handling**: Improving algorithms like DOVER-Lap to manage high overlap density and mixed single-speaker regions.
- **Domain Adaptation**: Expanding pre-training strategies to include underrepresented accents and dialects.
- **Scalability**: Developing lightweight yet robust systems for deployment in compute-constrained environments.

By addressing these areas, diarization and identification systems can move closer to achieving universal applicability, bridging gaps between research and real-world usability.

3 Methodology

3.1 Data Gathering

The data that will be used in this project comprises Egyptian Arabic and English speech, meticulously curated by Mohammed Rashad [11] and sourced from various platforms, including YouTube and contributions from bilingual Egyptian speakers. This dataset includes 12,480 audio-text pairs with audio durations of up to approximately 24.9 seconds per pair. Initially designed for Automatic Speech Recognition systems, this dataset now serves as a foundation for building and testing the diarization models. Additionally, another dataset tailored for the ArzEn-LLM system was collected through advanced tools such as LLME and Gemma. This dataset consists of 25,557 pairs of audio-text segments, contributed by 38 bilingual speakers, and includes 12 hours of diverse audio content sampled at 16 kHz to ensure compatibility with transcription models such as Whisper [12].

Since ground truth segmentation was not available, Praat was used to extract initial speaker segmentation information. Then the output was saved in RTTM format to determine each segment's start time, duration, and speaker ID. These RTTM entries were manually validated and modified by annotators to ensure precision. After annotation, the RTTM files were converted into metadata.json files as follows:

```
SPEAKER <file> 1 <start> <duration> <NA> <NA> <speaker_id> <NA> <NA>
```

Figure 3.1 illustrates the initial approach of this project to perform the language diarization task, it can be simplified into 5 main steps as shown above, each step has its own level of complexity that is going to be discussed in subsequent sections.This dataset setup supports training and testing of a simple model that identifies whether the input audio is in Arabic or English.

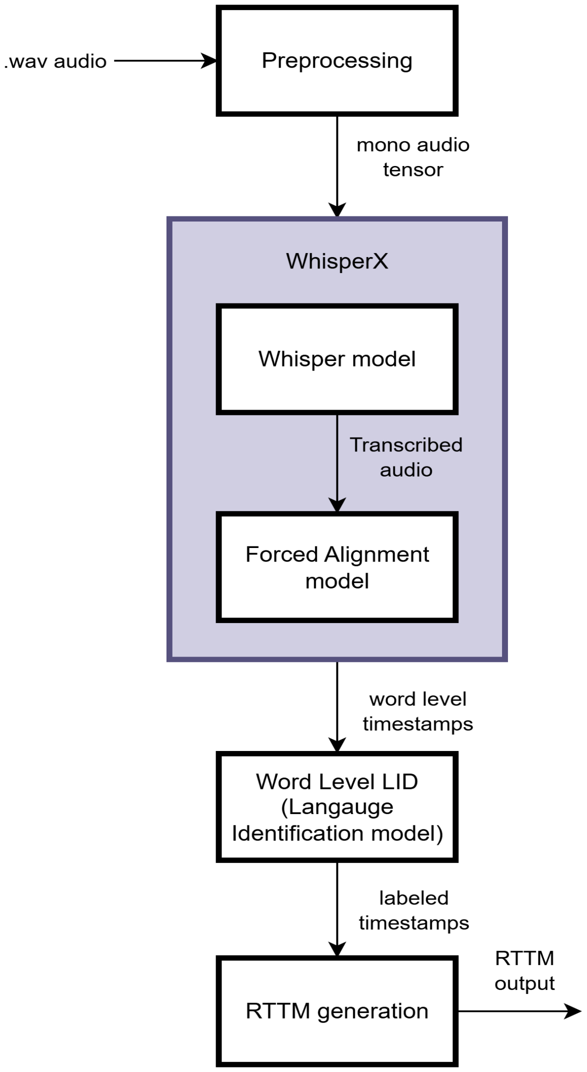

Figure 3.1: Proposed System's Full Pipeline

3.2 Preprocessing

Preprocessing is an essential step in the pipeline, ensuring consistency and quality across all subsequent stages. Audio data was already normalized to a 16 kHz sampling rate. All audio recordings were resampled to mono to have the same channel layout.

As part of preprocessing, the audio data was filtered to include only files that are more than 5 seconds in length. The audio file first starts as a mono conversion, followed by normalization to adjust audio levels. After that, the signal is resampled, converted into tensors, and finally stored as preprocessed audio for the next steps in the pipeline.

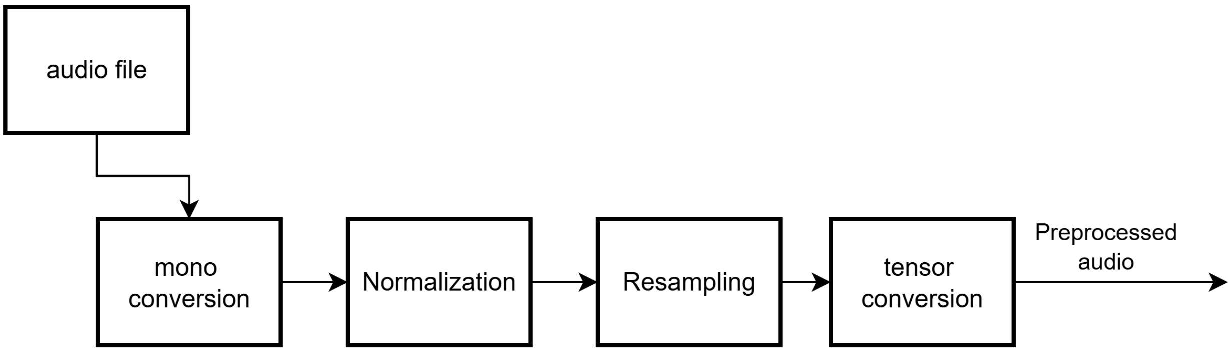

Figure 3.2: steps for audio pre-processing

3.3 Whisper Model

What is Whisper

Whisper is a pretrained ASR system developed by OpenAI [13], trained on approximately 680,000 hours of multilingual and multitask weakly supervised data. The dataset, comprising audio paired with transcripts collected from the web, emphasizes diversity, which significantly contributes to the model's robustness. This includes handling accents, background noise, and technical language. Whisper has demonstrated competitive performance compared to large open-source unsupervised speech recognition models, such as Wav2Vec.

WhisperX

WhisperX is an extension of the Whisper ASR model, specifically designed to enhance temporal alignment of transcribed speech [14]. It builds on Whisper's transformer-based architecture, focusing on generating precise word-level timestamps by aligning transcriptions to audio signal. It uses forced alignment techniques, leveraging acoustic and linguistic features to map each transcribed word to its corresponding time interval, while accounting for variations in speech rate, pauses, and prosodic patterns. This alignment process produces a structured output of word segments, each containing the word, its start time, end time, and additional metadata, which is very useful for applications such as language diarization.

Why Whisper

Whisper was selected for this language diarization project due to its exceptional robustness and versatility in processing multilingual and multitask speech data. Unlike conventional models requiring fine-tuning for specific datasets or languages, Whisper performs effectively in zero-shot settings. This capability makes it ideal for projects involving diverse audio sources. Whisper supports both English and non-English language processing and integrates language identification and transcription within a unified architecture. Its strong performance in out-of-distribution scenarios ensures reliable language detection even in noisy or challenging environments. Furthermore, Whisper's scalability, state-of-the-art robustness, and open-source availability make it a natural choice for building accurate and adaptable language diarization systems [13].

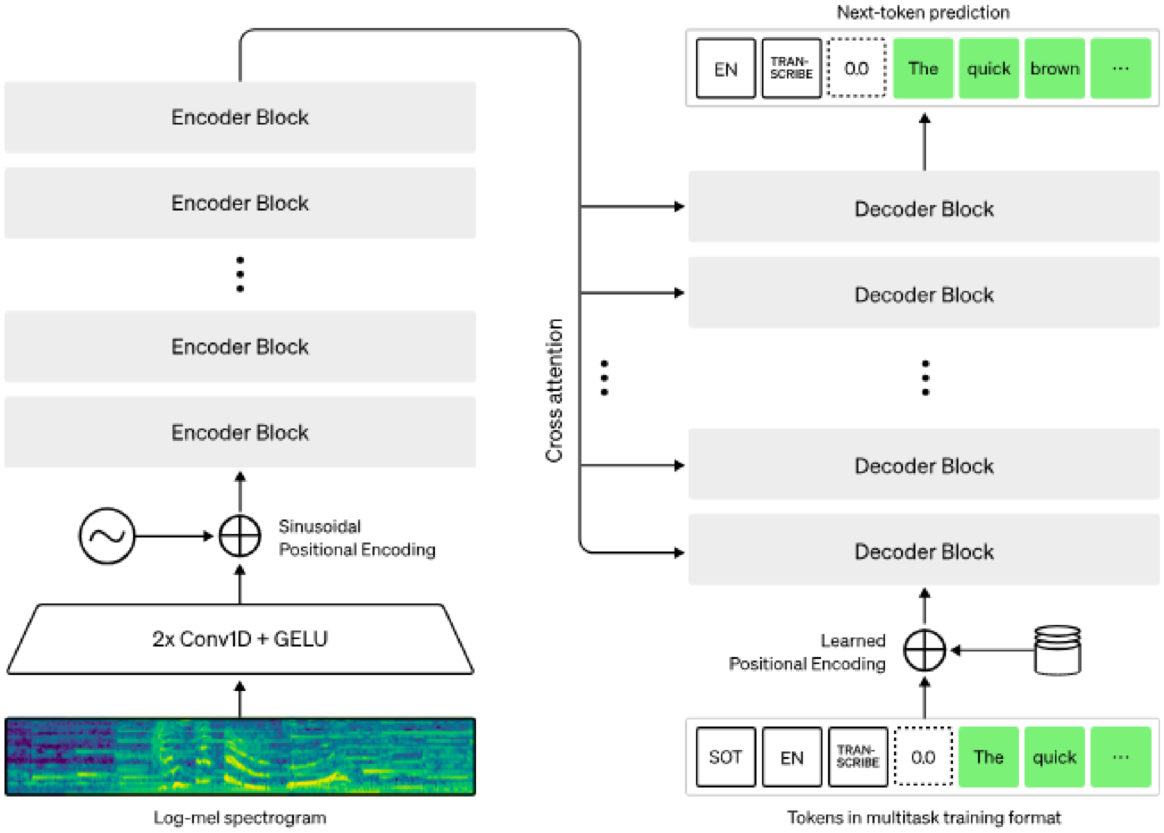

Figure 3.3: Whisper's architecture [13].

Whisper's Architecture

Whisper employs an encoder-decoder Transformer architecture tailored for various tasks, including transcription, translation, voice activity detection, and language identification. The input audio is resampled to 16 kHz, and 80-channel log-MEL spectrograms are computed using 25ms windows with a stride of 10 seconds. Whisper offers model variants ranging from small

(39M parameters) to large (1.55B parameters), allowing flexibility depending on application requirements. Figure 3.3 shows a simplified block diagram for the whisper model:

WhisperX Step

Once audio preprocessing had been completed, all the files were processed through WhisperX to generate transcriptions and word-level time stamps. Transcription was first performed using WhisperX's `medium` model, followed by alignment using WhisperX's internal phoneme-level model. This alignment enhances the timing precision of each word in the transcript.

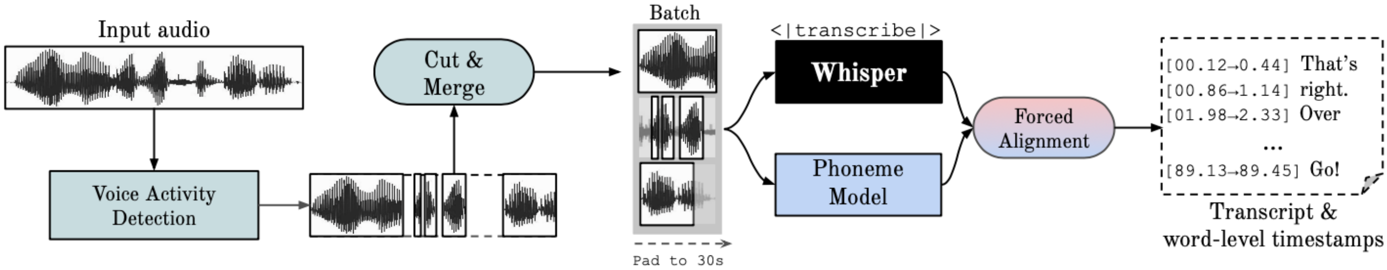

Figure 3.4: WhisperX architecture [14]

Word segments that were aligned were then entered into the SpeechBrain language identification model (`lang-id-voxlingua107-ecapa`, or the CNN based LID that we built) [15], which predicts probabilities for Arabic and English. For every word, a language label was obtained from these results. Audio segments were skipped to avoid inaccurate results when there was insufficient audio or silence. Moreover, when two sequential segments shared the same language label, they were combined into longer, continuous segments representing contiguous speech in the same language. For each combined segment, an RTTM line was constructed, containing the start time, duration, allocated language, and a normalized confidence score based on combined language probabilities.

Also, it is important to note how we dealt with silence periods between words; we simply put a silence threshold of 0.2 seconds between words. If the gap between words is larger than that, then we label this period as non-speech, which saves time by not needing to identify the language in that segment. But if the gap was smaller than said threshold, then we extend the end time of the first word to include it.

3.4 Language Identification Model

Background on the LID used

The ‘speechbrain/lang-id-voxlingua107-ecapa‘ model, developed by the SpeechBrain team, is a robust language identification system trained on the VoxLingua107 dataset, which includes speech samples from 107 languages [15]. It employs an ECAPA-TDNN (Embedding Calibration and Aggregation with Temporal Deep Neural Networks) architecture, which processes acoustic features, such as spectral and prosodic patterns, to classify spoken languages. This model is designed to accurately distinguish languages, including Arabic and English, by analyzing short audio segments, making it suitable for multilingual speech processing tasks.

The LID step

After WhisperX gives us the word timings and dealing with the silence gaps utilizing the silence threshold, we use the LID model to check the language of each segment. Since we are only interested in the codes-switching between Arabic and English, we compare the probabilities of these two languages, and we label the segment with the language that has a higher probability. This lets us catch cases where speakers switch between Arabic and English within the same audio. The language labels are then added to an RTTM file, which organizes all the information neatly. Each line in the RTTM file lists a word’s start time, duration, and language, making it easy to see when and how languages change.

It is important to mention how we dealt with very short words, as they don’t necessarily have enough information for the LID for it to be able to label it accurately; we simply extend segments shorter than 0.1 second to 0.5 by duplicating the samples in the utterance.

3.5 Audio Features

The use of audio features, such as acoustic and phonetic features, arises from the need for automation in spoken language tasks. These features capture key characteristics of speech, such as the manner and place of articulation, and provide meaningful insights into the signal. Some features, like Mel-Frequency Cepstral Coefficients (MFCCs) and its first and second derivatives [16] are especially effective in representing the spectral properties of audio, extracting harmonics and sidebands, which help in accurately capturing the signal’s structure.

However, the challenge with these features lies in the separate feature extraction process that is required. MFCCs and similar features rely on pre-processing steps that involve transforming the raw audio into a more manageable form, which can be computationally intensive and may not always capture higher-level abstractions, those type of features are more useful for data that doesn't necessarily contain temporal information (due to an audio for example being too short), so we mainly utilized MFCCs in extracting features for word level embeddings in the language identification step.

On the other hand, end-to-end feature extraction methods, such as those that generate embeddings, are more suitable for deep learning approaches. These methods extract meaningful representations of audio directly from raw waveforms, allowing for more efficient processing in complex pipelines. They eliminate the need for manual feature engineering and enable more flexible, scalable systems that can handle a wide variety of spoken language tasks.

3.6 RTTM File Generation

The final step in the pipeline involves generating the RTTM (Rich Transcription Time Marked) file. This file serves as the diarization output, containing segments formatted with details such as audio ID, start time, end time, and language label, we also made sure that no consecutive lines in the RTTM have the same label by combining them. The RTTM file provides a structured representation of the diarization results, ready for downstream applications.

The output of the Whisper model is converted into RTTM format, enabling further processing in the diarization pipeline. The general RTTM format is as follows:

```
SPEAKER <Audio_ID> <Channel> <Start_Time> <Duration> <Language_Label> <Confidence>
```

Where:

- $< Audio\_ID >$: Identifier for the audio file.
- $< Channel >$: Usually 1 for single-channel audio.
- $< Start\_Time >$: Start time of the segment in seconds.
- $< Duration >$: Duration of the segment in seconds.
- $< Language\_Label >$: "Arabic" for AR or "English" for EN.

- $< Confidence >$: Confidence score of the prediction (e.g., 1.000).

Below is a possible snippet of an RTTM file:

```
SPEAKER file1.wav 1 0.000 0.800 <NA> <NA> Arabic <NA> 1.000
SPEAKER file1.wav 1 0.800 0.400 <NA> <NA> English <NA> 1.000
SPEAKER file1.wav 1 1.200 0.400 <NA> <NA> Arabic <NA> 1.000
SPEAKER file1.wav 1 1.600 0.400 <NA> <NA> English <NA> 1.000
```

4 Experiments and Results

This section presents the experimental evaluation of our language diarization system. We assessed the performance of a custom-trained Language Identification (LID) model and a diarization pipeline using both the custom LID and the pretrained speechbrain/lang-id-voxlingua107-ecapa model. The evaluation leverages the Egyptian-English code-switching dataset, with results measured using classification accuracy for the LID model and Diarization Error Rate for the diarization system.

4.1 Premature Attempts

Early in the project, we explored the use of Wav2Vec 2.0, a self-supervised speech representation model, to generate embeddings for language diarization in English-Arabic code-switched speech. The goal was to extract high-dimensional embeddings capturing acoustic and linguistic features, which could then be clustered to identify language boundaries. To manage the high dimensionality of Wav2Vec 2.0 embeddings, we applied dimensionality reduction techniques, such as Principal Component Analysis (PCA), to obtain uniform feature representations. Visualizations of the reduced embeddings were analyzed to detect distinct clusters corresponding to Arabic and English segments. However, the results revealed no clear separation between language classes, likely due to the complex interplay of acoustic features in code-switched speech and the model's broad pretraining on multilingual data, which diluted its specificity for our binary classification task. Additionally, integrating these embeddings into a diarization pipeline proved challenging, as the lack of temporal alignment hindered accurate segmentation. Recognizing these limitations, we transitioned to WhisperX, which offers robust transcription and forced alignment capabilities tailored for language diarization, enabling precise word-level segmentation and language identification in our pipeline.

4.2 Dataset Handling and Ground Truth Annotation

To evaluate the language diarization system, we utilized the Egyptian-English code-switching dataset curated by Mohammed Rashad, supplemented by data from the ArzEn-LLM collection [11]. The dataset comprises 12,480 audio-text pairs with durations ranging from 5 seconds to approximately 90 seconds, capturing diverse conversational scenarios with frequent language switches between English and Arabic. The dataset was already sampled at 16 kHz, ensuring compatibility with our models. Given the absence of pre-annotated language segments, we created a manual ground truth using Praat, a phonetic analysis tool widely used for speech segmentation. We manually reviewed each audio file to identify language-homogeneous regions and code-switching points based on auditory and contextual cues. The annotations were exported to Praats' TextGrid format, which were later converted into RTTM format using a Python library. Each line in the RTTM specifies the audio file identifier, start time, duration, and language label (Arabic: "AR"; English: "EN"). This ground truth served as the reference for evaluating the accuracy of our diarization system, ensuring robust benchmarking against manually verified language boundaries.

4.3 Custom Language Identification Model

To develop a specialized LID model for English-Arabic code-switching, we trained a lightweight convolutional neural network (CNN) using data from parts of the Mozilla Common Voice project, specifically the `Common Voice Corpus 6.1` [17] and `Common Voice Delta Segment 17.0` datasets [18]. These datasets provide diverse speech samples in Arabic and English, suitable for binary classification tasks. Each audio segment was preprocessed into uniform feature representations with a shape of $(200, 120)$,where 200 represents the number of time frames and 120 corresponds to the number of features per frame.These features include 40 MFCC coefficients, 40 delta (first derivative) values, and 40 delta-delta (second derivative) values.

The dataset was split as follows:

- **Training Set:** 14,702 samples (50% Arabic, 50% English).

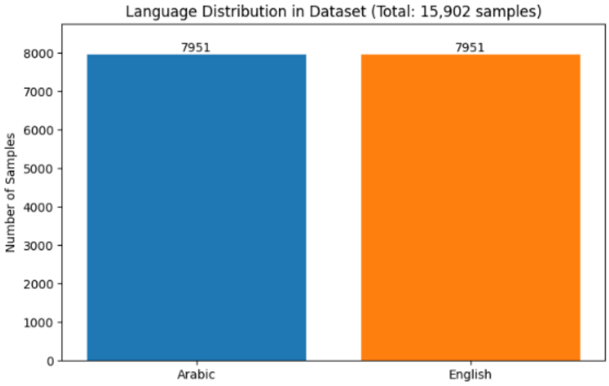

Figure 4.1: Data distribution of LID dataset.

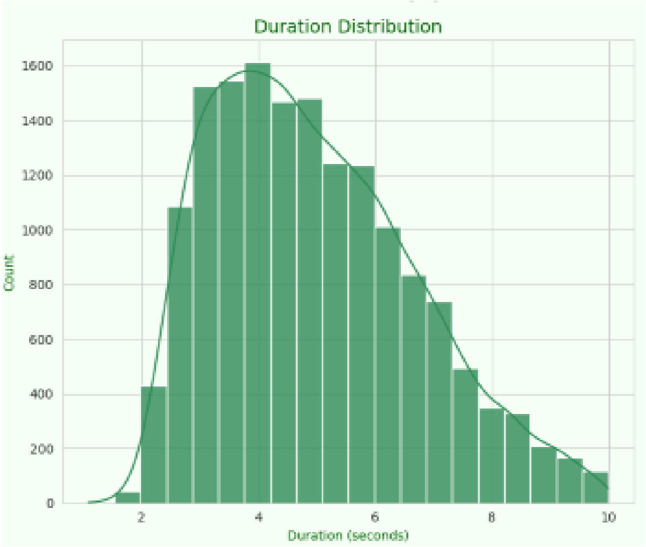

Figure 4.2: time distribution of LID dataset.

- **Validation Set:** 600 samples (balanced across languages).
- **Test Set:** 600 samples (balanced across languages).

Figure 4.1 shows the data distribution across the two languages. Equal numbers of samples were taken for the two languages to prevent bias in the LID classification.

Figure 4.2 shows the duration distribution of the audio samples in the taken samples for the LID. It shows that most audio samples lie in the 3 to 6 seconds range.

Layer (type)	**Output Shape**	**Param #**
conv2d (Conv2D)	(None, 200, 120, 64)	640
batch_normalization (BatchNormalization)	(None, 200, 120, 64)	256
max_pooling2d (MaxPooling2D)	(None, 100, 60, 64)	0
dropout (Dropout)	(None, 100, 60, 64)	0
conv2d_1 (Conv2D)	(None, 100, 60, 128)	73,856
batch_normalization_1 (BatchNormalization)	(None, 100, 60, 128)	512
max_pooling2d_1 (MaxPooling2D)	(None, 50, 30, 128)	0
dropout_1 (Dropout)	(None, 50, 30, 128)	0
conv2d_2 (Conv2D)	(None, 50, 30, 256)	295,168
batch_normalization_2 (BatchNormalization)	(None, 50, 30, 256)	1,024
max_pooling2d_2 (MaxPooling2D)	(None, 25, 15, 256)	0
dropout_2 (Dropout)	(None, 25, 15, 256)	0
flatten (Flatten)	(None, 96000)	0
dense (Dense)	(None, 256)	24,576,256
batch_normalization_3 (BatchNormalization)	(None, 256)	1,024
dropout_3 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32,896
batch_normalization_4 (BatchNormalization)	(None, 128)	512
dropout_4 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 2)	258

Table 4.1: Summary of CNN Architecture

Table 4.1 shows the full architecture of the model. It has approximately 25 million parameters, the majority of which are trainable. After the training process, the final model size was around 350MB due to the additional information stored for optimization (e.g., weights, optimizer states, and buffers). The architecture was designed to balance depth and computational efficiency, enabling the model to learn both temporal and spectral patterns effectively from the input features. Batch normalization and dropout were incorporated to improve generalization and reduce overfitting. The model was trained using a categorical cross-entropy loss function with the Adam optimizer, and training was conducted over multiple epochs with early stopping based on validation loss to ensure optimal performance without overtraining.

The CNN model was trained to classify each segment as either Arabic or English, optimizing for binary classification accuracy. The evaluation results, summarized in Table 4.2, demonstrate the model's effectiveness in distinguishing between the two languages. The model shows strong performance across precision, recall, and F1-score metrics, indicating its robustness in handling varied audio durations and speaker accents within controlled settings.

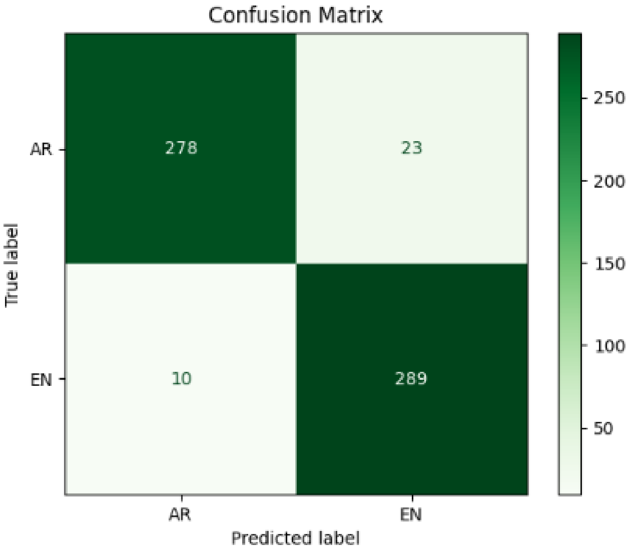

Figure 4.3: Confusion matrix of the CNN-based LID

Figure 4.3 shows the confusion matrix of the LID that we trained. It shows that the model is slightly better at spotting English than Arabic - English segments are misclassified as Arabic only 10 times, wheres Arabic segments are misclassified as English 23 times. This might be due to the diversity of the dialects in the Arabic dataset.

Class	**Precision**	**Recall**	**F1-score**	**Support**
0 (Arabic)	0.97	0.92	0.94	301
1 (English)	0.93	0.97	0.95	299
Accuracy			0.94	600
Macro avg	0.95	0.95	0.94	600
Weighted avg	0.95	0.94	0.94	600

Table 4.2: CNN LID results

4.4 Diarization Approaches and Evaluation

4.4.1 Evaluation Metric: Diarization Error Rate

The primary metric for evaluating our language diarization system is the Diarization Error Rate, a widely adopted measure in speech processing to assess the accuracy of segmenting and labeling speech regions [19]. DER quantifies the proportion of time in which the system's output does not match the ground truth annotations, capturing errors in language boundary detection for English-Arabic code-switched speech. It comprises three components: missed detection (time assigned to no language when a language is present), false alarm (time assigned to a language when none is present), and confusion error (time assigned to the wrong language) as shown in equation 1.

$$\text{DER} = \frac{\text{Missed} + \text{False} + \text{Confusion}}{\text{Total Duration}} * 100\% \tag{1}$$

where:

- Missed: Duration of segments where the system fails to detect a language present in the ground truth.
- False: Duration of segments where the system incorrectly assigns a language to non-speech or incorrect regions.
- Confusion: Duration of segments where the system assigns an incorrect language label (e.g., Arabic instead of English).
- Total Duration: Total duration of the reference audio.

DER was chosen as the primary metric due to its comprehensive evaluation of temporal and labeling accuracy, critical for language diarization tasks where precise identification of language switches is essential. Its ability to break down errors into specific categories allows for targeted analysis of system performance, particularly in challenging code-switching scenarios with rapid transitions or dialectal variations. Additionally, DER's standardization in speech diarization research facilitates comparison with baseline systems, ensuring robust benchmarking of our model's effectiveness [19].

We evaluated two diarization models: one using the pretrained `speechbrain/lang-id-voxlingua107-ecapa` LID model and another using our custom-trained LID model, both integrated with WhisperX for transcription and word-level alignment. The evaluation was conducted on the Egyptian-English dataset, focusing on the model's ability to accurately segment and label language boundaries in code-switched speech.

4.4.2 Pretrained Model: VoxLingua107-ECAPA

This model utilized `WhisperX` to generate word-level transcriptions and timestamps, followed by the `speechbrain/lang-id-voxlingua107-ecapa` model for language prediction on each word segment [15]. `WhisperX`'s medium model performed transcription, and its phoneme-level forced alignment enhanced temporal precision, segmenting audio into separate words. The LID model assigned language labels (Arabic or English) based on probability scores for each word segment in the audio. Consecutive segments with the same language were merged to form continuous segments, and the results were exported as `RTTM` files.

Performance metrics for this approach include:

- **Diarization Error Rate:** [37.85%], comprising:
  - Average Missed detection:[0.2609]
  - False alarms: [0.0068]
  - Language confusion errors: [3.4608]

The pretrained model performed reliably on longer segments with clear articulation but showed higher errors on short utterances and dialect-heavy speech, likely due to its broad training on 107 languages diluting focus on Arabic-English nuances.

4.4.3 Custom LID Model

The second approach replaced the pretrained LID with our custom-trained CNN-based LID model, integrated after `WhisperX`'s transcription and alignment. The custom LID focused on binary classification (Arabic vs. English), leveraging its specialized training to improve detection of language switches in Arabic-English code-switched contexts. The same segment-merging logic was applied, producing `RTTM` files for evaluation.

Performance metrics for this approach include:

- **Diarization Error Rate:** [40.09%], comprising:
  - Average Missed detection: [0.2581]
  - False alarms: [0.0068]
  - Language confusion errors: [3.8127]

The model that uses the Voxlingua pretrained LID model showed a slightly improved performance compared to the model that uses the LID that we trained, this might be because the pretrained model is trained on a more diverse set of languages (107), which helped the model capturing subtle differences between languages and dialects more robustly and in more different settings, contrary to the LID model that we trained which was trained only on Arabic and English datasets.

LID Approach	DER (%)	Missed Det.	False Alarms	Confusion Errors
VoxLingua107-ECAPA	37.85	0.2609	0.0068	3.46
CNN-based LID	40.09	0.2581	0.0068	3.81

Table 4.3: Summary of Performance for Pretrained and Custom LID Approaches

4.5 Limitations and Considerations

This methodology integrates advanced models and techniques into a cohesive pipeline, aiming to produce accurate and reliable language diarization results. By leveraging the capabilities of Whisper for transcription and WhisperX for precise word-level alignment, along with a language identification stage using either a pretrained or custom LID model, the system is designed to handle the intricacies of English-Arabic code-switched speech. Figure 3.1 illustrates the overall workflow used to generate the final RTTM files from segmented and labeled speech.

However, the proposed system is still under active development and has not yet undergone extensive testing. It is expected to evolve significantly as experimentation continues. Several key challenges have been identified throughout the design and implementation process:

- **Computational Overhead:** Both Whisper and WhisperX are computationally intensive, leading to a noticeable bottleneck during transcription and alignment, especially when processing large datasets or long-duration recordings.

- **Short Segment Limitations:** Language identification models often perform better on longer utterances. Short word-level segments may lack sufficient acoustic information, making accurate language classification more difficult and prone to error.

- **Dialectal Variations:** The Arabic language encompasses a wide range of dialects. Differences in pronunciation, vocabulary, and prosody between dialects may negatively affect the accuracy of LID models that are not trained to generalize across such variability.

- **Lack of Annotated Datasets:** There is a shortage of publicly available datasets containing Arabic-English code-switched speech with ground-truth language segmentation in RTTM format. This necessitated manual annotation using tools like Praat, which is labor-intensive and time-consuming.

- **Task Complexity:** Language diarization remains a non-trivial problem. Even state-of-the-art systems report high Diarization Error Rates, especially in spontaneous conversations with frequent and rapid code-switching. This makes the evaluation and refinement of diarization models particularly challenging.

These challenges highlight the experimental nature of the project and motivate the need for further research, optimization, and validation of the proposed system.

5 Conclusion

This project explores the challenges of Arabic-English code-switching in speech processing, a phenomenon that poses significant difficulties for traditional Automatic Speech Recognition systems. The primary focus is on language diarization—accurately segmenting code-switched speech into distinct language regions to enhance the performance of multilingual speech processing systems.

Two datasets were used in this project. The Egyptian-English code-switched dataset was employed for the development and evaluation of the complete diarization system. Alongside this, the Mozilla Common Voice Arabic and English datasets were used to train a lightweight CNN-based language identification model. Although the primary dataset focused on the Egyptian dialect, the Whisper model was found to perform effectively in these cases, showing that the proposed approach is capable of generalizing across various Arabic dialects.

The proposed framework uses WhisperX for transcribing speech and aligning words with their timestamps, and an LID model for identifying whether each segment is in Arabic or English. The final diarization output is generated in RTTM format, which enabled us to assess the model's accuracy by comparing it with the RTTM of the ground truth - numerically representing its accuracy by the DER.

The project followed a multi-step approach that involved segmenting the audio, detecting language switches between Arabic and English. The performance evaluation revealed that the pretrained VoxLingua107-ECAPA model achieved a DER of 37.85%, while the custom CNN-based LID model achieved 40.09%.

6 Future work

This project establishes a foundation for English-Arabic code-switching diarization, but several avenues remain for further development. To address the computational intensity of Whisper and WhisperX, alternative word-level forced alignment methods, such as those based on Hidden Markov Models (HMMs) or lightweight neural networks, could be explored to reduce processing demands while maintaining accuracy. Testing the model on diverse datasets, including those with other Arabic dialects (e.g., Palestinian, Levantine) or additional code-switching pairs (e.g., Arabic-Hebrew), would enhance its generalizability across multilingual contexts. Investigating alternative language diarization approaches, such as end-to-end neural architectures or ensemble methods, could improve segmentation precision and robustness to rapid language switches. Additionally, optimizing the model's efficiency through techniques like model pruning or quantization would facilitate deployment in resource-constrained environments, such as mobile devices or real-time applications like live subtitling. Incorporating speaker diarization to handle multi-speaker scenarios and improving robustness to noisy audio or varied accents would further strengthen the system. Finally, evaluating the model with additional metrics, such as Word Error Rate (WER), would provide a more comprehensive assessment of its performance, ensuring its applicability to real-world multilingual speech processing tasks.

References

[1] *The Cambridge Handbook of Linguistic Code-switching* (Cambridge Handbooks in Language and Linguistics). Cambridge University Press, 2009.

[2] R. Franceschini, “History of multilingualism,” 2012.

[3] C. Spence. “How learning a new language changes your brain.” (2022), [Online]. Available: `https://www.cambridge.org/elt/blog/2022/04/29/learning-language-changes-your-brain/`.

[4] M. Abdel-Fattah, “Arabic-hebrew language-switching and cultural identity,” vol. 12, pp. 183–195, 2011. DOI: `10.33806/ijaes2000.12.1.11`.

[5] B. Vachhani, D. Singh, and R. Lawyer, “Multi-resolution approach to identification of spoken languages and to improve overall language diarization system using whisper model,” 2023. DOI: `10.21437/Interspeech.2023-1354`.

[6] D. Raj, L. P. Garcia-Perera, Z. Huang, S. Watanabe, D. Povey, A. Stolcke, and S. Khudanpur, *Dover-lap: A method for combining overlap-aware diarization outputs*, 2020. [Online]. Available: `https://arxiv.org/abs/2011.01997`.

[7] M. Shahin, Z. Nan, V. Sethu, and B. Ahmed, “Improving wav2vec2-based spoken language identification by learning phonological features,” *INTERSPEECH 2023*, 2023, pp. 4119–4123. DOI: `10.21437/Interspeech.2023-2533`.

[8] V. Y. H. Chua, H. Liu, L. P. G. Perera, F. T. Woon, J. Wong, X. Zhang, S. Khudanpur, A. W. H. Khong, J. Dauwels, and S. J. Styles, *Merlion ccs challenge: A english-mandarin code-switching child-directed speech corpus for language identification and diarization*, 2023. [Online]. Available: `https://arxiv.org/abs/2305.18881`.

[9] S. Gupta, S. Hiray, and P. Kukde, “Spoken language identification system for english-mandarin code-switching child-directed speech,” 2023, pp. 4114–4118. DOI: `10.21437/Interspeech.2023-1335`.

[10] S. V, V. Thenkanidiyoor, and D. Dileep, “Svm based language diarization for code-switched bilingual indian speech using bottleneck features,” 2018. DOI: `10.21437/SLTU.2018-28`.

[11] M. Rashad, *Arabic-english-code-switching*, 2024. [Online]. Available: `https://huggingface.co/datasets/MohamedRashad/arabic-english-code-switching/edit/main/README.md`.

[12] A. Heakl, Y. Zaghloul, M. Ali, R. Hossam, and W. Gomaa, *Arzen-llm: Code-switched egyptian arabic-english translation and speech recognition using llms*, arXiv preprint arXiv:2406.18120, 2024. [Online]. Available: `https://arxiv.org/pdf/2406.18120`.

[13] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, *Robust speech recognition via large-scale weak supervision*, 2022. arXiv: `2212.04356 [eess.AS]`. [Online]. Available: `https://arxiv.org/abs/2212.04356`.

[14] M. Bain, J. Huh, T. Han, and A. Zisserman, *Whisperx: Time-accurate speech transcription of long-form audio*, 2023. arXiv: `2303.00747 [cs.SD]`. [Online]. Available: `https://arxiv.org/abs/2303.00747`.

[15] M. Ravanelli, T. Parcollet, P. Plantinga, *et al.*, “Speechbrain: A general-purpose speech toolkit,” *arXiv preprint arXiv:2106.04624*, 2021. [Online]. Available: `https://arxiv.org/abs/2106.04624`.

[16] Z. K. Abdul and A. K. Al-Talabani, “Mel frequency cepstral coefficient and its applications: A review,” *IEEE Access*, vol. 10, pp. 122 136–122 158, 2022. DOI: `10.1109/ACCESS.2022.3223444`.

[17] Mozilla Foundation, *Common voice corpus 6.1*, `https://commonvoice.mozilla.org/en/datasets`, Accessed: July 2025, 2020.

[18] Mozilla Foundation, *Common voice delta segment 17.0*, `https://commonvoice.mozilla.org/en/datasets`, Accessed: July 2025, 2024.

[19] T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, “A review of speaker diarization: Recent advances with deep learning,” *Computer Speech & Language*, vol. 72, p. 101 317, 2022. DOI: `10.1016/j.csl.2021.101317`.