**Table of Contents**

Abstract

1. Introduction
   1. Introduction to Language
   2. The Problem Statement
   3. Challenges of identifying language from speech
2. Literature Survey: Unveiling Languages from Speech Samples with Embeddings
   1. Purpose
   2. Key concepts
   3. State-of-Art pre Trained Models: Revolutionizing Various Tasks
3. Metholody
   1. Data Acquisition and Preprocessing
   2. Pre Trained ECAPA – TDNN Model
   3. Anomaly Detection: OC-SVM A Deep Dive
      1. Core Concepts
      2. Key Components
      3. How it Works
   4. Support Vector Machine(SVM) for Classification: A Deep Dive
      1. Core Concepts
      2. Key Components
      3. How it Works
4. System Development
   1. SpeechBrain
   2. Librosa
   3. Scikit-learn: A Powerful Machine Learning Library for Python
5. Result Analysis
6. Conclusion and Future Work

**Abstract**

Language Identification refers to the process of detecting the language(s) of the speech or in text in the document based on the script used for writing and observing the diacritics particular to a language. This research area has always fascinated researchers as early as 1970 and till now due to varied applications and increased demands of this field. In this project we basically focus on identifying language from spoken audio samples. The method leverages a two-stage classification process to achieve accurate language recognition while mitigating the impact of outliers. Key stages include: Embedding extraction, Outlier Detection, Language Classification.

We utilize the ECAPA-TDNN architecture from SpeechBrain, a pre-trained deep learning model, to generate audio embeddings from the speech samples. These embeddings represent the speech data in a high-dimensional space, capturing essential characteristics relevant to language identity. An OCSVM is employed as the first stage. It is trained solely on speech data from known languages. New speech samples are mapped into the embedding space. The OCSVM identifies and discards outliers (potentially containing noise or non-speech elements) that fall outside the expected boundary of known languages. The remaining speech samples, deemed likely to represent actual speech from known languages, are passed to a standard SVM classifier. This SVM, trained on labeled speech data from multiple languages, classifies the samples into their respective languages based on the informative embeddings.

The project will evaluate the effectiveness of this multi-stage approach using appropriate metrics like accuracy, precision, recall. We will analyze the impact of the OCSVM stage in improving the accuracy of the final language classification by the SVM.

**Introduction**

* 1. **Introduction to Language**

Language, a system of conventional spoken, manual, or written symbols by means of which human beings, as members of a social group and participants in its culture, express themselves. The functions of language include communication, the expression of identity, play, imaginative expression, and emotional release.

The scientific study of language is called **linguistics**. It explores various aspects like:

* **Phonology:** The sound system of a language, analyzing how sounds are produced and combined.
* **Morphology:** The structure of words, including prefixes, suffixes, and how words are formed.
* **Syntax:** The rules for arranging words into sentences and phrases.
* **Semantics:** The meaning of words and sentences, including how context affects meaning.
* **Pragmatics:** How language is used in specific contexts to achieve communication goals.

There are thousands of languages spoken worldwide, each with unique features and histories. Languages can be categorized into families based on shared origins, like the Indo-European family which includes English, Spanish, and Hindi. Languages are constantly evolving, adapting and changing over time.

* 1. **Problem Statement**

Language identification (LID) from speech samples presents a captivating challenge at the intersection of computer science and linguistics. This technology aims to automatically determine the language spoken in a given audio clip.

Different systems of communication constitute different languages; the degree of difference needed to establish a different language cannot be stated exactly. Generally, systems of communication are recognized as different languages if they cannot be understood without specific learning by both parties, though the precise limits of mutual intelligibility are hard to draw and belong on a scale rather than on either side of a definite dividing line. So, here comes the role of Language Translation which implies the role and importance of Language Identification.

Language identification from speech samples is a rapidly evolving field with immense potential. As AI and linguistics continue to converge, we can expect even more innovative solutions that bridge the gap between languages and enhance communication across the globe.

**1.3 Challenges of Identifying Language from Speech**

Despite its potential, speech-based LID faces certain challenges:

* **Similarities Between Languages:** Languages often share characteristics, making it difficult for computers to differentiate between them definitively. Loanwords (words borrowed from another language) and dialect variations within the same language can further complicate the task.
* **Non-verbal Cues and Background Noise:** Speech samples can be influenced by factors beyond language itself. Accents, speaker characteristics, and background noise can pose hurdles for LID systems, requiring them to be robust to such variations.
* **Code-switching:** The practice of mixing languages within a single conversation adds another layer of complexity. LID systems need to be able to handle these situations and accurately identify the underlying languages used.

Despite significant advancements, several challenges remain. Developing LID systems that work effectively with limited data for less common languages is a critical area of research. Data augmentation techniques and transfer learning from high-resource languages are promising approaches to address this issue. Additionally, handling code-switching and recognizing specific dialects or accents remain ongoing challenges.

**2. Literature Survey: Unveiling Languages from Speech Samples with Embeddings**

**2.1 Purpose**

Language identification (LID) from speech samples plays a crucial role in bridging communication gaps across languages. This survey delves into the current landscape of speech-based LID, exploring the feature extraction techniques, classification models, and emerging trends that shape this domain.It aims to explore the field of language identification (LID) using speech samples. The goal is to understand the current state-of-the-art techniques, identify key challenges, and explore potential future directions for this technology.

**2.2 Key Concepts:**

* **Language Identification (LID):** The automatic process of determining the language spoken in a speech sample.
* **Speech Features:** Informative characteristics extracted from speech audio, such as Mel-Frequency Cepstral Coefficients (MFCCs) or embeddings learned by deep learning models.
* **Classification Models:** Algorithms that learn to distinguish between different languages based on extracted features. Examples include Support Vector Machines (SVMs) and deep neural networks.
* **Deep Learning:** A subfield of machine learning using artificial neural networks with multiple layers to learn complex representations from data.
* **Embeddings:** Compressed representations of speech data learned by deep learning models, capturing essential features for language identification.
* **Code-switching:** The practice of mixing languages within a single utterance.
* **Dialect Recognition:** Identifying the specific regional variation of a language spoken.

**2.3** **State-of-the-Art Pre-trained Models: Revolutionizing Various Tasks**

Pre-trained models have revolutionized various fields within machine learning, including natural language processing (NLP), computer vision, and speech processing. These models are trained on massive datasets and capture general-purpose knowledge that can be fine-tuned for specific tasks, significantly improving performance compared to training models from scratch. Here's an overview of the current landscape

**Speech Models:**

* **wav2vec 2.0:** This pre-trained model, based on a transformer encoder, learns general audio representations from raw waveforms. It can be fine-tuned for various speech processing tasks, including speech recognition, speaker identification, and language identification.
* **ECAPA TDNN (Emphasized Channel Attention, Propagation and Aggregation Time Delay Neural Network):** This model family leverages pre-trained embeddings and a time delay neural network architecture to capture temporal information in speech. ECAPA TDNN models excel in tasks like language identification and speaker identification, particularly when computational efficiency is a concern.

**3. Methodology**

Language identification (LID) systems aim to automatically determine the spoken language from a speech sample. Here's a breakdown of the typical methodology involved:

**3.1 Data Acquisition and Preprocessing:**

VoxLingua107 is a valuable open-source dataset for training language identification (LID) models. Here's how you can collect data from it. The VoxLingua107 dataset is publicly available at <https://bark.phon.ioc.ee/voxlingua107/>. There you'll find information about the dataset and download links. VoxLingua107 consists of speech segments in WAV format, each labeled with the language of the video from which it was extracted. The dataset also includes a separate evaluation set with verified language labels.

**3.2 Pre-trained ECAPA TDNN Model:**

ECAPA TDNN models have emerged as a powerful tool for various speech processing tasks, including language identification (LID). These models can learn informative representations of speech data, and extracting embeddings from them forms the foundation for LID applications. Repositories on Hugging Face provide pre-trained ECAPA TDNN models for speaker verification, along with tools for inference.

Understanding the ECAPA TDNN Architecture:

ECAPA TDNN models have a specific architecture that processes speech data to learn speaker or language representations. Here's a simplified breakdown of the key components:

* **Convolutional Layers:** These layers extract local features from the speech waveform, capturing patterns within short time windows.
* **Time Delay Neural Network (TDNN) Blocks:** TDNN layers introduce time delays, allowing the model to learn features across longer temporal contexts in the speech signal, crucial for LID as language-specific information can unfold over time.
* **Squeeze-and-Excitation (SE) Layers:** These layers enhance the model's ability to focus on informative features by dynamically adjusting channel-wise importance.
* **Residual Connections:** These connections help the model learn from deeper layers by adding the output of earlier layers to the output of later layers, promoting efficient learning.

The embeddings we aim to extract reside in the activations of the penultimate layer, typically a fully-connected layer, before the final classification layer. This layer represents the compressed representation of the speech data, capturing essential features that can be used to distinguish between languages.

**3.3 Anomaly Detection: OC-SVM A Deep Dive**

In the context of open-set language identification (LID), where you want to identify known languages and also reject speech samples from unseen languages, OC-SVM can be a valuable tool.

One-Class SVM (OC-SVM) is a powerful machine learning algorithm that excels at outlier detection and novelty identification. Unlike traditional SVMs designed for classification tasks with multiple classes, OC-SVM focuses on learning a boundary around the "normal" data distribution. This makes it ideal for scenarios where you have a good understanding of the typical data but limited information about anomalies or outliers.

**3.3.1 Core Concept:**

Imagine a dataset representing normal data points in a high-dimensional feature space. OC-SVM aims to create a boundary (often a hyperplane or a more complex surface) that encloses this "normal" region. Any data point falling outside this boundary is considered an outlier or anomaly.

**3.3.2 Key Components:**

* **Feature Space:** Data points are transformed into a higher-dimensional space using feature extraction or engineering techniques. This allows for more complex separation between normal data and outliers.
* **Kernel Function (Optional):** OC-SVM can utilize kernel functions to project data points into a higher-dimensional space, enabling the learning of non-linear decision boundaries in the original feature space. Common kernels include Gaussian RBF and polynomial kernels.
* **Decision Boundary:** The algorithm optimizes the decision boundary to maximize the distance between the boundary and most of the data points (normal data) while allowing for a small number of errors (data points inside the boundary). This trade-off is controlled by a hyperparameter called nu (v).
* **Nu (v) Parameter:** This crucial parameter controls the trade-off between the volume of the enclosed normal data region and the number of training errors (data points within the boundary). A higher nu leads to a tighter boundary, potentially excluding more inliers as outliers. Conversely, a lower nu creates a looser boundary, potentially including some outliers within the normal data region.

**3.3.3 How it Works:**

1. **Data Preprocessing:** Data is preprocessed by cleaning, normalization, and potentially feature extraction or engineering to create informative features for outlier detection.
2. **Kernel Selection (Optional):** If a non-linear decision boundary is necessary, a suitable kernel function is chosen.
3. **Model Training:** The OC-SVM algorithm iteratively optimizes the decision boundary to maximize the distance between the boundary and most of the data points, considering the nu parameter for error tolerance. OC-SVM excels at learning a decision boundary that encompasses the "normal" data distribution in the feature space. In open-set LID, the normal data represents the speech features of languages you've included in the training set.
4. **Outlier Detection:** New data points are projected into the feature space using the same transformation as the training data. Points falling outside the learned decision boundary are classified as outliers. Any speech sample that falls outside the learned decision boundary is considered an outlier. In open-set LID, these outliers represent languages not present in the training data.

## 3.4 Support Vector Machines (SVM) for Classification: A Deep Dive

SVMs (Support Vector Machines) are a cornerstone algorithm in machine learning, particularly for classification tasks. They excel at finding the optimal decision boundary to separate different classes in a high-dimensional feature space. Here's a detailed explanation of how SVMs work for classification:

**3.4.1 Core Concept:**

Imagine a dataset with data points belonging to different classes, like emails categorized as spam or not spam. SVMs aim to find a hyperplane (a line in 2D or a plane in 3D) that separates these classes with the **maximum margin**. This margin refers to the distance between the hyperplane and the closest data points from each class, called **support vectors**. These support vectors play a crucial role in defining the best hyperplane for classification.

**3.4.2 Key Components:**

* **Feature Space:** Data points are represented by features (numerical attributes) that capture their essential characteristics. These features can be directly used or transformed into a higher-dimensional feature space using techniques like **kernel functions** (explained later).
* **Hyperplane:** This is the decision boundary that separates the data points of different classes. The goal of the SVM is to find the hyperplane with the maximum margin, ensuring a clear distinction between classes.
* **Support Vectors:** These are the data points closest to the hyperplane from each class. They define the margin and significantly influence the classification process.
* **Kernel Functions (Optional):** While SVMs can work in lower dimensions, they can leverage kernel functions to project data points into a higher-dimensional space. This allows for learning non-linear decision boundaries in the original feature space, separating classes that wouldn't be separable with a straight line. Popular kernel functions include Gaussian RBF and polynomial kernels.

**3.4.3 How it Works (Step-by-Step):**

1. **Data Preprocessing:** The data is cleaned, normalized, and potentially transformed using feature extraction or engineering techniques to create informative features for classification.
2. **Kernel Selection (Optional):** If the data is not linearly separable (meaning a straight line cannot perfectly divide the classes), a suitable kernel function is chosen to project data points into a higher-dimensional space.
3. **Model Training:** The SVM algorithm finds the optimal hyperplane that maximizes the margin between the hyperplane and the support vectors from each class. This optimization process involves solving a mathematical equation.
4. **Classification:** New, unseen data points are projected into the feature space using the same transformation as the training data (and the kernel function if used). The SVM model then classifies the new data point based on which side of the decision boundary (hyperplane) it falls on.

**4.Development Environment**

This work presents a Language Identification (LID) system for speech data developed using a combination of user-friendly and powerful Python libraries. The system leverages pre-trained models from SpeechBrain, a deep learning toolkit for speech processing, for feature extraction and initial model training. The extracted features are then used to build and train a Support Vector Machine (SVM) classifier using scikit-learn, a popular machine learning library.

**4.1 SpeechBrain**

SpeechBrain plays a crucial role in feature extraction and initial model training. SpeechBrain offers a variety of pre-trained models for different speech processing tasks, including feature extraction. These models are trained on large datasets and can capture valuable information from speech signals. SpeechBrain models often provide outputs beyond just features one of them is ECAPA-TDNN. It generates language embeddings that represent the overall language characteristics of the speech input. These embeddings, along with features, could be used as input for the SVM classifier.

**4.2 Librosa**

Librosa is a versatile and user-friendly Python library specifically designed for audio and music analysis tasks. It offers a comprehensive set of functionalities for various operations, making it a valuable tool for projects involving sound processing. Reads audio files in various formats like WAV, MP3, FLAC, etc.Provides functionalities for resampling, normalization, and other audio preprocessing techniques. Librosa offers a well-documented and easy-to-use API, making it accessible even for beginners in audio processing. The library covers a wide range of audio processing tasks, from basic feature extraction to music information retrieval. Being open-source, Librosa benefits from a large community of developers, ensuring continuous improvement and a wealth of online resources.

## 4.3 Scikit-learn: A Powerful Machine Learning Library for Python

## Scikit-learn (pronounced "scikit learn") is a popular and robust Python library for machine learning tasks. It provides a comprehensive set of tools for various stages of the machine learning workflow, making it a go-to choice for building and deploying production-ready machine learning models.

 **Supervised Learning:**

* **Classification:** Algorithms for classifying data points into different categories, such as Support Vector Machines (SVMs), Random Forests, K-Nearest Neighbors (KNN), and Logistic Regression.
* **Regression:** Algorithms for predicting continuous target values based on input features, including Linear Regression, Decision Trees, and Gradient Boosting.

 **Unsupervised Learning:**

* **Clustering:** Techniques for grouping similar data points together without predefined labels, such as K-Means clustering, Hierarchical clustering, and DBSCAN.
* **Dimensionality Reduction:** Methods for reducing the number of features in a dataset while preserving essential information, including Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA).

 **Model Selection and Evaluation:**

* **Train-Test Split:** Tools for splitting data into training and testing sets to evaluate model performance.
* **Cross-validation:** Techniques to assess model generalizability and avoid overfitting.
* **Evaluation Metrics:** Functions to calculate various metrics for evaluating model performance, like accuracy, precision, recall, F1-score, and confusion matrix.

 **Preprocessing:**

* **Scaling:** Techniques for scaling features to a common range to improve model performance.
* **Encoding:** Methods for handling categorical data by converting them into numerical representations suitable for machine learning algorithms.

 **Pipelines:** Tools for chaining preprocessing and model fitting steps into a single workflow for streamlined model building.

##  ****Model Persistence:**** Functionality to save trained models for future use and deployment.

## Scikit-learn offers a consistent and well-documented API, making it easy to learn and use, even for beginners in machine learning.The library provides a broad range of algorithms for supervised and unsupervised learning tasks, catering to diverse machine learning problems.Scikit-learn integrates well with other scientific Python libraries like NumPy, Pandas, and Matplotlib, creating a powerful ecosystem for data science and machine learning.