**Machine Language Identifier**

**LANGUAGE IDENTIFIER:**

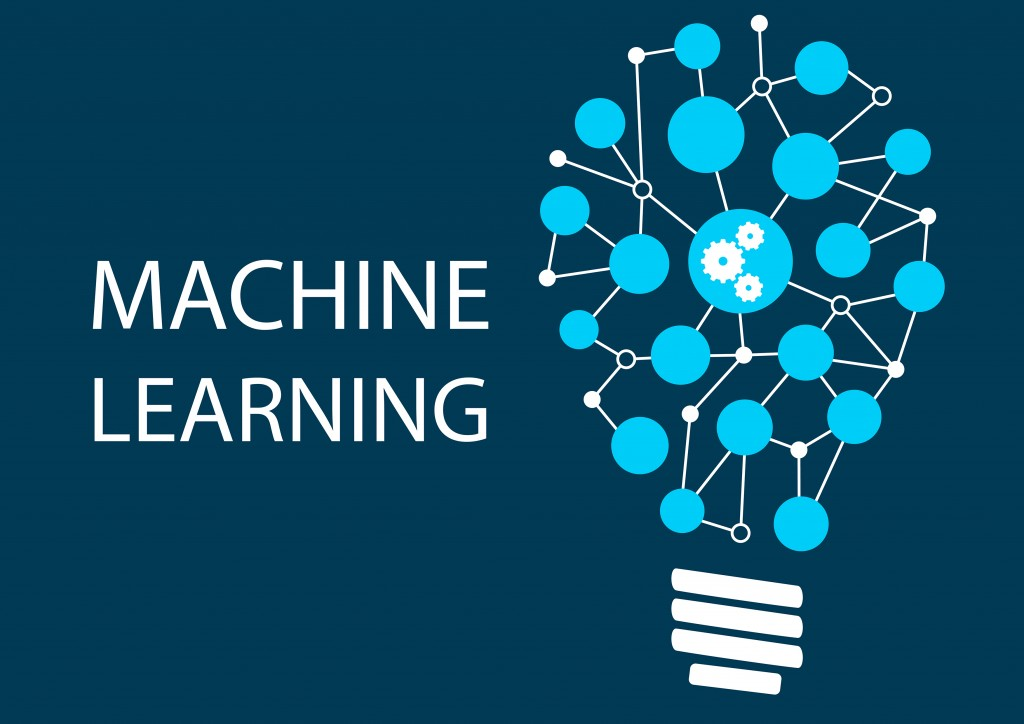
**ABSTRACT:**

"PUSH YOURSELF BECAUSE NO ONE ELSE IS GOING TO DO IT FOR YOU"

To begin the topic, we must know the knowledge on Machine Language. The behind the ML is simple and says it in one line as a tool in research on theorem proving compiler technology, and program analysis.

Spoken Language identification is the process of detecting the language of an utterance by an anonymous speaker, irrespective of gender, accent, and pronunciation. Implementation of an acoustic model for Spoken Language Identification is to be carried out in this project. The major task is to identify those d=features or parameters which could be used to clearly distinguish between languages.

**GRATIFY OF BLOG:**



General Introduction

Problem Statement

Testing and Results

Functional Requirements

Software Requirements

Hardware Requirements

System Design

Conclusion

References

Github Link

**General Introduction:**

Spoken Language Identification (LiD) is the process of identification of the language spoken in an utterance. Automatic language identification is the problem of identifying the language being spoken from a sample of speech by a speaker.



Language identification can be useful when working with user-provided text, which often doesn't come with any language information. They are supporter relates to language as follows.

Broad language support: Identifies over one hundred different languages.

Romanized text support: Identifies Arabic, Bulgarian, Greek, Hindi, Japanese, Russian, and Chinese text in both native and romanized script.

**Problem Statement:**



The language says for example set considered in our project includes English, French, Japanese, Hindi, and Kannada. The generic LiD should be adopted to include regional languages like Hindi and Kannada. The system should evolve over time with better accuracy and use a continuous learning mechanism by incorporating machine learning techniques.

The concept gets a scope of LiD system lies in making the LiD the pre-processing stage for Language translation. This would be the worst contribution to the field of speech translation and many other applications can make use of speech as a more dominant input to machines.

**Testing and Results:**

ML Kit recognizes text in more than 100 different languages in their native scripts. In addition, romanized text can be recognized for Arabic, Bulgarian, Chinese, Greek, Hindi, Japanese, and Russian.

To get into the level of identifier build. Gradle file, It's an important role to include Google's Maven repository in both build script and another project too.

Let me explain with you an ML kit Android libraries to this module.

Code for language identifier:

dependencies{

implementation 'com.google.mlkit:language-id:16.1.1'

}

To catch up into language of a string:

Usually string calls LanguageIdentification.getClient() to get an instance of LanguageIdentifier. It passes the string method.

Let us assume the JAVA code for example:

LanguageIdentifier languageIdentifier =

LanguageIdentification.getClient():

languageIdentifier.identifyLanguage(text)

.addOnSuccessListener(

new OnSuccessListener<String>() {

@Override

public void OnSuccess(@Nullable String languageCode) {

if else condition is accuried

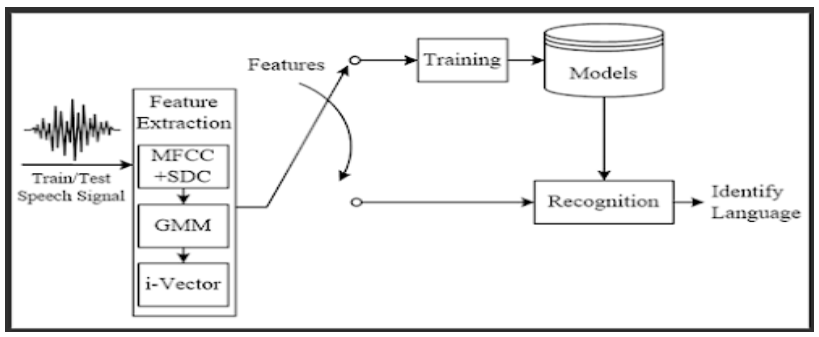
It's the default condition, ML kit returns a value other than confidence value of at least 0.01

This threshold by passing a LanguageIdentificationOptionss object to getClient():

**HOW IT WORKS???**

When using a LID system several types of information are considered. Furthermore, human understanding has inspired the classification of information, and several studies have applied methods that people have used to different languages, whether consciously or not. A broad classification has been to separate or split speech features into a low level and high level.

Identification: Language Identifying the type and location of digital evidence can be a challenge.



The core classification unit is an important part of any LID system. The role of the classification unit is to map the audio sets and extract features from the i-vector system to enable its corresponding language to be identified. Instantiate the ML kit Language Identifier is located in the main folder. Navigate to the file add the following field to the identifier. This is how you get a handle on the language identifier.

To view into Python code:

func createFile(language, sample, filename string) error {

file, err :=os.Create(lanuage + "/" + filename + ".txt")

if err ! = nil {

log.Fatal("CANNOT ACCESS FILE", err)

}

The above code details the memorable of the file running this script, which is stored in a hard drive named for a different language. Inside each folder were text files with the language samples are given.

STEP 1: Process all the samples

STEP 2: Split 20% of the samples into a validation set

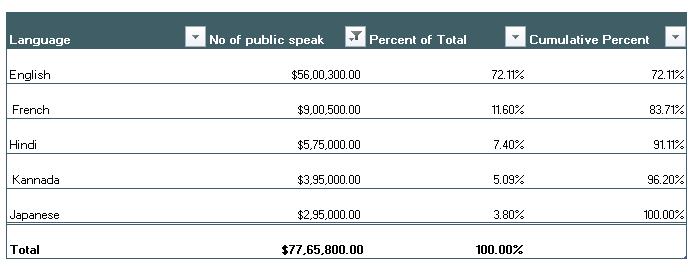
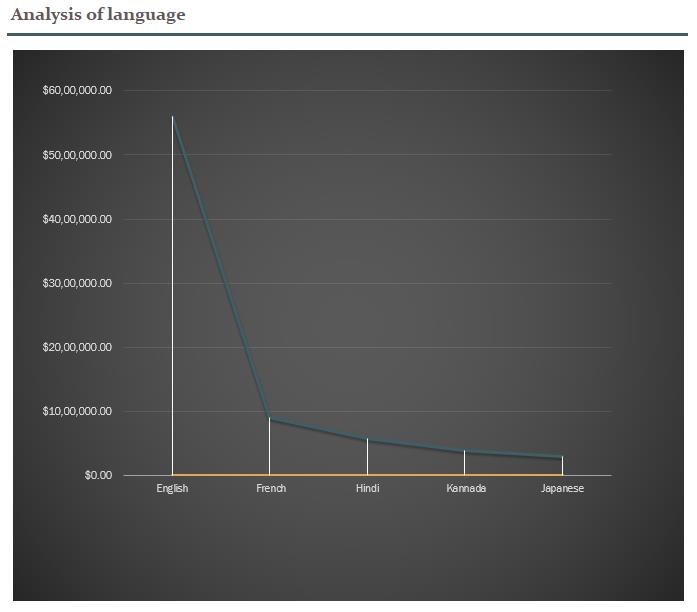
STEP 3: Train Classificationbox with the training set

STEP 4: Validate with the validation set

Collection: All over the world language are collected in one single package. The collection step is critical since this is the first real contact with evidence.

The experiments are conducted to analyze the response of the proposed LiD against the considered language (English, Hindi, French, Japanese, and Kannada). The result is depicted in the form of a confusion matrix.

This is given below the graph requires for the Analysis of language using identifier ML kit.



MainViewModel.kt

private val languageIdentifier = LanguageIdentification.getClient()

In addition, you also want to be sure that the clients are properly shut down when it's no longer needed.

NOTE: The first attempt at native language identification was a simple n-gram based approach, which is one of the most common approaches to this problem. The code for this attempt is stored in the directory. While it is known that this approach gives fairly good results, the implemented naive Bayes has been included for the sake of completion. The XGBoost model demonstrates an attempt at using a state-of-the-art machine learning model for language classification with 235 languages without any ngram parsing.

**FUNCTIONAL REQUIREMENTS:**

In the field of language identification, there have been several attempts at building an ELM-based language classifier to replace the classical SVM developed a new variant of an extreme learning machine applied to language identification. The improved algorithm is known as the regularized Minimum Class Variance Extreme Learning Machine(RMCVELM). The core concept of the algorithm is to minimize the empirical risk, structural risk, and intra-class variance.

Requirement, which is related to the functional aspect of software falls into define function, the user should be able to mail any report to management.

The system recognizes the language irrespective of the type of speaker, gender, and accent of the speaker. Given a speech sample input to the system, it should extract the acoustic feature MFCC and use the information to detect the language. The input speech sample should not contain abrupt utterance, noise, or background music.

**SOFTWARE REQUIREMENTS:**

The process of gathering the software requirements from clients, analyze and document them is known as requirement engineering.

Analysis communicates with clients and end-users to know their ideas on what the software should provide and which features they want the software to include. Software Requirement Specification is a document created by the system analyst after the requirement is collected from various stakeholders. It deals with the client is written in natural language. It is the responsibility of system analysts to document the requirement in technical language so that they can be comprehended.

**HARDWARE REQUIREMENTS**:

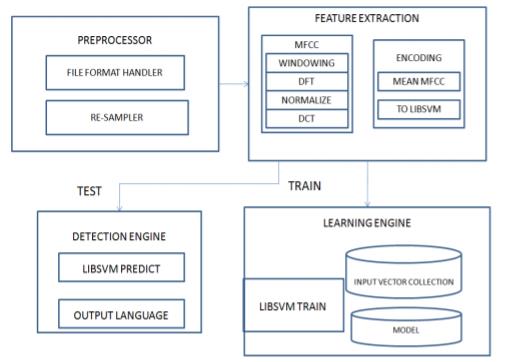
Average load rarely presents problems that cannot be addressed by gradually increasing the hardware capacity of the system. The burst load is when a form or a report is subjected to vary heavy usage within a short period of time. Burst load is less predictable and can lead to a slowdown in the system that can affect all users.

**SYSTEM DESIGN**:

The process of defining an interface, algorithms, data infrastructure, and hardware for ML.Learning system to meet specific requirements of reliability, scalability, maintainability, and adaptability.

The client-side comprises a portal that allows the user to upload an audio speech sample using a browser. The sample is sent to the network to the remote server running a LiD system. The LiD system processes this request and detects the language of the speech sample. The server returns a response indicating the language identified.

Modern speech technology systems rely on interdisciplinary research in the area of multimodal signal processing and artificial intelligence, and a number of methods and algorithms have been developed with the aim of solving the various problems: dialogue system based on speech recognition and synthesis, including emotional speech, speaker identification, and verification as well as speech signal coding and transmission, denoising and detection of the human voice



The LiD system takes speech samples as the input. There are three processing blocks in the architecture, they are Pre-Processing Block, Feature Extraction Block, and Machine Learning Block. Pre-Processing block is concerned with making the speech sample suitable for acoustic feature extraction. At this stage, the audio file format is checked and converted to .wav if the input file format is .mp3. The final output of this block is a Pre-Processed Speech sample.

The main task of the following extraction block is to extraction block. The feature extraction block initially incorporation a windowing function, in our case a Hamming window aimed at making the signal zero-valued outside the chosen interval.

Speech models based on either speech production or auditory perception were inherent parts of most successful algorithms. The most recent neuro-inspired computational models are based on knowledge of the cognitive speech processing model. After a brief review of speech production and auditory perception including cognitive and linguistic points of view will be elaborated.

**CONCLUSION:**

All on we come to an end of this blog, we observed that a LID architect can indeed lead to improved accuracy across multiple scripts, Such results confirm the viability of the approach for a fast and accurate language identifier operating. Regardless, the LID system has a smaller memory footprint compared to n-gram and is scalable with the addition of large amounts of data without becoming prohibitively large.

"SUCCESS IS NO ACCIDENT!"

And so be a harder person ask yourself if what you are doing today, is getting you to closure to where you want to be tomorrow...

**REFERENCES:**

https://www.tutorialspoint.com/software\_engineering/software\_requirements.html

https://www.google.com/search?q=SRS+should+come+up+with+following+features:&rlz=1C1CHBD\_enIN909IN909&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiR-Jviw4DyAhVYWX0KHSYlBwsQ\_AUoA3oECAEQBQ&biw=759&bih=718#imgrc=Ufd0TzJr8JjmSM

https://trustedcomputinggroup.org/wp-content/uploads/Hardware-Requirements-for-Device-Identifier-Composition-Engine-r78\_For-Publication.pdf

https://www.perfectforms.com/Documentation/manual/html/server\_side\_hardware\_requireme.html

https://www.analyticsvidhya.com/blog/2021/01/a-look-at-machine-learning-system-design/

https://machinelearning.apple.com/research/language-identification-from-very-short-strings

https://www.hindawi.com/journals/cin/2019/4368036/

**GITHUB LINK:**