**Speech - To - Text**

Automatic Speech Recognition (ASR) is a system where a computer takes a speech from a recorded audio signal and then converts it into the corresponding text. Getting a computer to understand and react appropriately to spoken language. It is the process by which a computer recognizes what a person said. The Speech recognition process involves an acoustic signal captured by a microphone and then it is converted to a set of words accordingly. A computer system is enabled to identify and respond to the sound produced during human speech known as speech recognition.

Speech to Text are of two types:

* The first are hybrid systems such as Kaldi that train a deep acoustic model to predict phonemes from audio processed into Mel Frequency Cepstral Coefficients (MFCCs), combine the phonemes using a pronunciation dictionary and finally pick the most likely results using a language model(both count based LM and RNN based LM).
* The second are end-to-end systems using a deep neural network to predict words directly from the audio or MFCCs. Such systems like RNN-T or wav2vec require a lot more training data and GPU resources for training.

Due to the massive data requirements of end-to-end systems only the biggest companies have used them to date. The data requirements also make it hard to train for new domains (even in the same language) and new languages or accents. Using a hybrid system, it is much easier to create a model for a new domain using minimal training data and a pronunciation dictionary with words added for that domain.

ASR systems are an essential part of various research fields and there are different asr systems found in literature. ASR systems can be classified into following types:

**1) Based on utterances:**

a) Isolated Words:

This recognition system recognizes only a single word at one time. User needs to give only one word response or command. The main advantage of this system is: It is simple and easy to implement because word boundaries are obvious that can be easily detected and the words are pronounced very clearly.

b) Connected Words:

This system is the same as an isolated word system, but it permits separate words to run-together with a minimum stop between them.

c) Continuous Speech This recognition system permits an individual to speak almost in a natural manner, during which the system computes its content. Generally, it is a computer dictation where the closest words run together without any pause or division between them. Such systems are more complex. d) Spontaneous Speech In spontaneous speech recognition, the system recognizes the natural speech. Spontaneous speech is a natural speech that comes suddenly through the mouth. A spontaneous speech ASR system is capable of handling a variety of natural speech features i.e. words being run together along with mispronunciation, stutters and false starts etc.

**2) Based on Speaker Model Speech recognition system can be divided into three main categories as follows:**

a) Speaker Dependent Models:

Speaker-dependent system works only for a particular type of speaker. They are more accurate for a particular speaker, but are less accurate for other types of speakers. They are cheaper and are easier to develop. But they are not as flexible as speaker independent systems. They can be used for security purposes.

b) Speaker Independent Models:

Speaker-independent system can recognize a variety of speakers without any prior training. It can be used in Interactive Voice Response System (IVRS) that must accept input from a large number of different users. But it limits the number of words in a vocabulary and implementation is also very difficult. It is expensive and it is less accurate than speaker-dependent systems.

c) Speaker Adaptive Models:

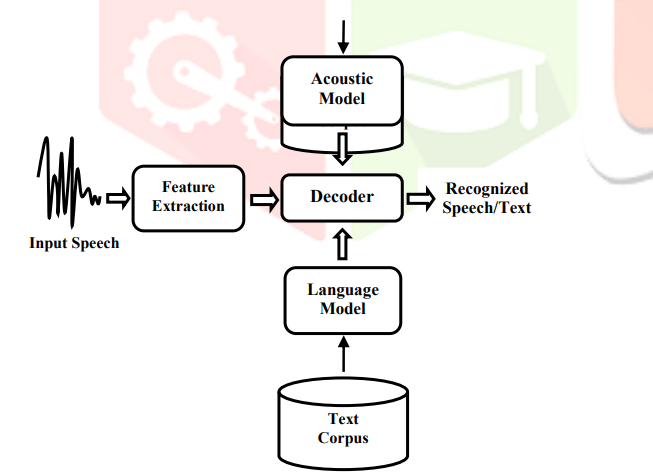
In such systems, the speaker-dependent data is used and is matched with the best-suited speaker to recognize the speech and to decrease an error rate after adoption. They adapt operation according to characteristics of the speakers.

**3) Based on Vocabulary**

The size of vocabulary can affect the complexity, processing rate and the rate of recognition of ASR systems. ASR systems are classified as follows:

* Small Vocabulary: contains 1 to 100 words or sentences.
* Medium Vocabulary: contains 101 to 1000 words or sentences.
* Large Vocabulary: contains 1001 to 10,000 words or sentences.
* Very-large vocabulary: contains more than 10,000 words or sentences.

**Block Diagram of ASR System**



1) Input Speech:

It is basically the recorded acoustic signal from different speakers. The acoustic signal is an analog signal. The analog signal cannot be directly transferred to the ASR system. So these signals are transformed into digital signals. This digital signal can now be processed.

2) Feature extraction:

It helps to find the set of parameters of utterances that have acoustic relation with speech signals. These parameters are called features. The main goal of feature extractor is to discard the irrelevant information and keep only relevant one. There are several methods for feature extraction such as Mel-Frequency Cepstral Coefficient (MFCC) [6], Perceptual Linear Prediction (PLP), Linear Predictive Cepstral Coefficient (LPCC) and RASTA-PLP (Relative Spectral Transform) etc. Figure 1 Block diagram of ASR System

3) Acoustic model:

Acoustic modeling is the fundamental part of the ASR system. It is the main part of Training. The acoustic model provides a connection between the acoustic Information and phonetics. Acoustic model plays an important role in the performance of the ASR system and is responsible for computational load. Acoustic model uses speech signals from the training database. Hidden Markov Model (HMM) is a widely used and accepted model as it is an efficient algorithm for training and recognition.

4) Language model:

It is also part of training. A language model contains the structural constraints available in the language to generate the probabilities of occurrence of a word followed by the sequence of n-1 words. Various models are used to find the exact word sequence like bi-gram, tri-gram, n-gram language models.

This is done by predicting the likelihood of the nth word, using the n-1 earlier words. The language model finds and differentiates between words and phrases that have similar sound.

5) Decoder:

Decoding (or recognition) is the process of comparing the unknown test pattern with each sound class reference pattern and computing the similarity between them to find the best match. After the completion of the training phase, system testing is performed. In testing phase patterns are classified to recognize the speech.

**Wav2Vec 2.0**

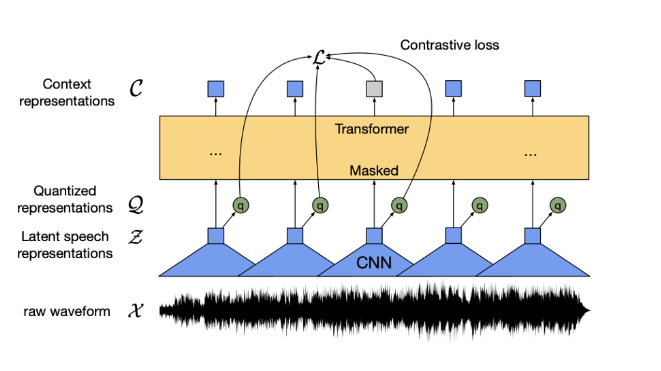
The promise of wav2vec 2.0 is pre-training without the supervised data using a large data set of recordings in the target domain. Afterwards, the model can be tuned using the supervised approach to maximize the accuracy. Wav2vec 2.0 shows that it’s possible to achieve low WER on LibriSpeech validation datasets using only ten minutes of labeled audio data. Another option is to use the pre-trained model (such as the libri-speech model) and just fine tune it for your domain with a few hours of labeled audio.

**The Architecture of Wav2Vec 2.0**

The breakthrough wav2vec 2.0 achieved is in adopting the masked pre-training method of the massive language model BERT. BERT masks a few words in each training sentence and the model trains by attempting to fill the gaps.

Instead of masking words, wav2vec 2.0 masks a part of the audio representation and requires the transformer network to fill in the gap.

The figure below shows the wav2vec 2.0 architecture with its two major components: CNN layers and transformer layers.



**Self Supervised learning**

The Self-Supervised learning works in the process of passing the raw audio waveform through CNN layers, and we get latent speech representations (Z in the figure above). Now, two things happen in parallel:

* We mask a random subset of Z, let’s call it masked\_Z. We pass masked\_Z into transformer layers. The output of the transformer layers is called context representations (C in the figure above).
* We apply product quantization [5] on Z and get quantized representations (Q in the figure above).

We expect C to be close to Q over the masked parts. The “error” between C and Q over the masked parts is called the contrastive loss. Minimizing contrastive loss enables transformer layers to learn the structure inside latent speech representations(Z).

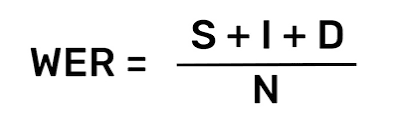
**Process:**

In the figure above, we saw that context representations were the output of transformer layers. Wav2vec 2.0 passes these context representations into a linear layer, followed by a softmax operation. The final output contains probability distributions over 32 tokens. A token can be a character, or it can represent word and sentence boundaries, as well as unknowns.

How do we convert these probability distributions into text? The answer is a decoder! The authors of wav2vec 2.0 used a beam search decoder. Below, we show you how to use a Viterbi decoder to convert the output of wav2vec 2.0 into text.

**Testing:**

We test the Speech Recognition system using Word Error Rate (WER) of the output text we received from the model.



S - substitution of words, I - Insertion of words, D - deletion of words, N - total words. This gives us the WER of the above model.

There is another library called **Jiwer** Which is based on the **Levenshtein distance.** the Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other.

“”” !pip install jiwer ”””. To use the library.