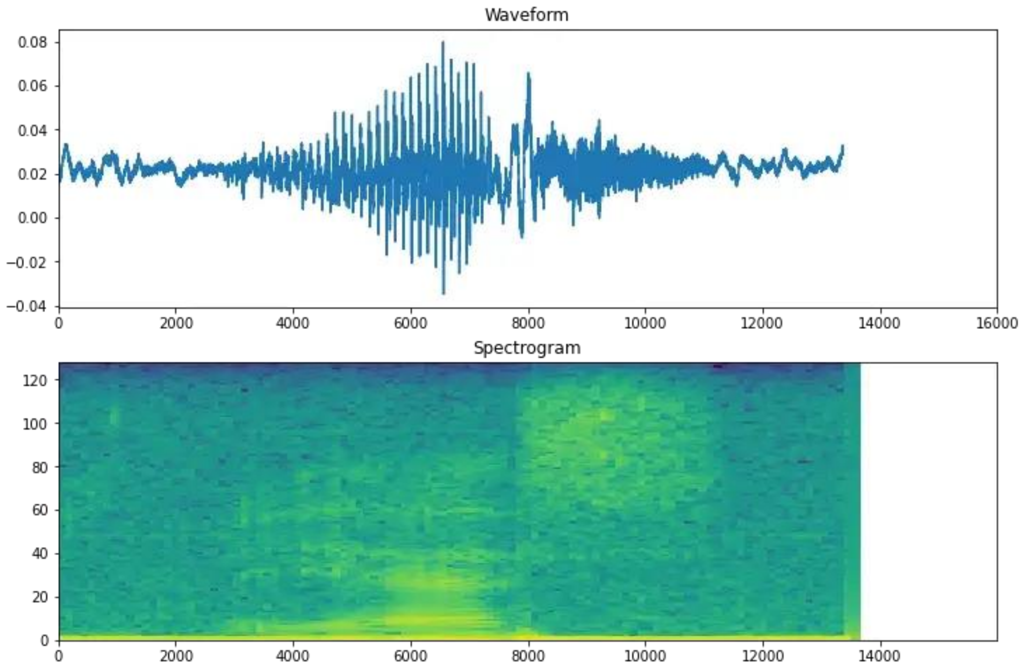
Automatic Speech Recognition

Dargier Antoine

Ponchon Martin

20/02/2023

Speech-to-text, or the Automatic Speech Recognition (ASR) problem



"Hello"

Audio Signal

Transcription

Introduction, Motivation & Related Work

1952
Recognition of 10 digits

1970s
IBM works with Jelinek

1972
First recognition system

2008
Google voice-activated search tool

2011
Apple and Siri

2017
Microsoft announces that it's a solved problem

Methodology

Dataset

kaggle

French Single Speaker Speech Dataset: 8,600 audios of *Les Misérables* and *Arsène Lupin contre Herlock Sholmes* and their transcripts in French

Metric

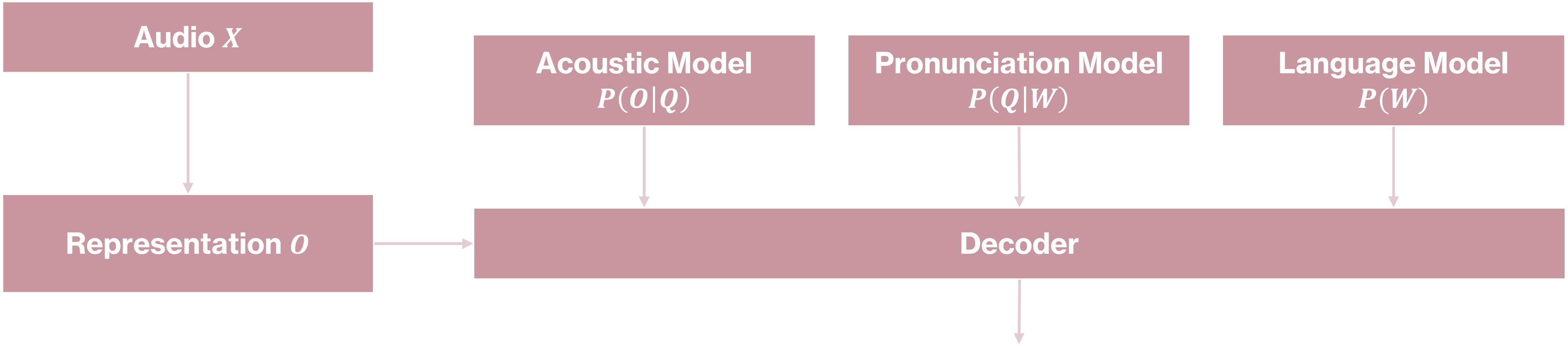
$$WER = \frac{S + D + I}{N}$$

where:

- S = number of substitutions (word replaced by another)
- D = number of deletions (word removed)
- I = number of insertions (word added)
- N = number of words in the reference

Traditional pipeline breaks down the ASR task into independent components

Words are decomposed into **phonemes** : Hello → HH AH L OW



$$W^* = \mathrm{argmax}_W P(W|X) = \mathrm{argmax}_W \sum_Q P(O|Q)\, P(Q|W) P(W)$$

Recently, Deep Learning has enabled significant improvements

Acoustic Model $P(O \mid Q)$	Pronunciation Model $P(Q \mid W)$	Language Model $P(W)$
Gaussian Mixture Models	Pronunciation tables	N-gram models
Deep Neural Networks Hidden Markov Models	Neural Network based pronunciation models	Neural Language Models

Going further

- These components are trained with different objective functions
- Error in one component may not behave well with error in another

Can we train end-to-end models that include all these components?

The CNN model

Pre-processing → Building the train/test sets → Building the CNN model → Validation

Compute **Mel-frequency cepstral coefficients** (MFCC) features on audios: short term power spectrum of a sound

Build input and output of the model computing the MFCC and converting vectors into binary matrix

Conv2D – MaxPooling2D – Conv2D – MaxPooling2D – Flatten – Dense – Dropout – Dense

8 hours of training on 2,000 audios of Kaggle dataset

Results

Results not usable because the learning time was too limited

WER > 20% in the literature

An ASR end-to-end system: Connectionist Temporal Classification (CTC)

Characteristics

- Usually implemented as a **grapheme based** model
- Solves the issue of **different lengths between input and output** transcription (using blank characters)

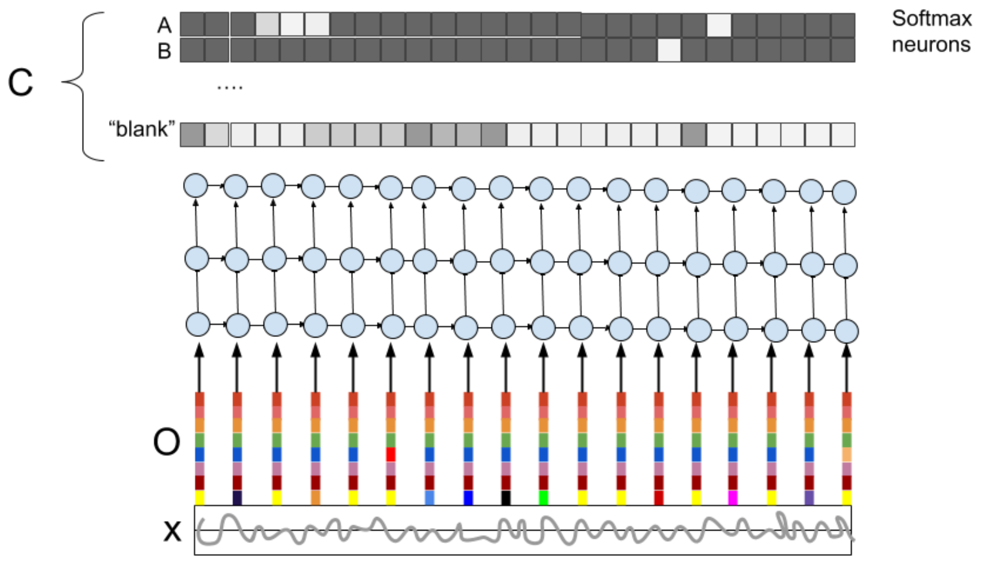
Idea

1. Use RNN, whose output neurons encode a **distribution over symbols** $c \in \{A, B, \ldots, space, blank\}$

2. Define a mapping **encoding c** $\mapsto y$ **word**:
   HHH_EE_LL_LO $\mapsto$ HELLO

3. Network parameters are updated to **maximize likelihood of choosing the correct label**:

$$\boldsymbol{L}(\boldsymbol{\theta}) = \log \mathbb{P}\left(\boldsymbol{y}^{*(i)} \middle| \boldsymbol{x}^{(i)}\right) = \boldsymbol{CTC}(\boldsymbol{c}^{(i)}, \boldsymbol{y}^{*(i)})$$



Results

Results not usable because of insufficient processing power

WER = 16% in the literature

Wav2Vec2.0

Developed by Meta in 2019

Authors: Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, Michael Auli

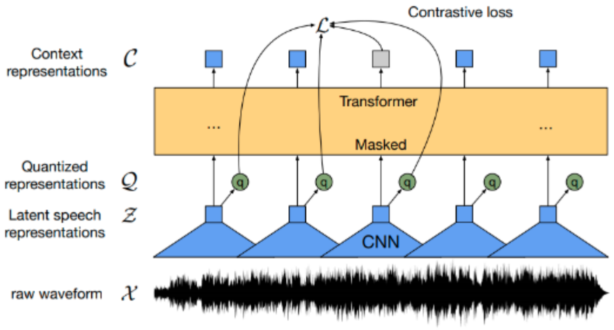
Advantages:

- **Self-supervised learning** so train without labelled data, low-resource: trained with 53k hours of unlabelled data, and just 10 minutes of labelled data. Trained on CommonVoice
- **Easy-implementation** with HuggingSound
- Used for **several languages** even without a lot of transcribed audios

Architecture:

CNN with 7 layers to learn the latent speech representations Z
Quantization: try to match the Z with codebooks, sort of phonemes
Transformers to learn contextual representations

Feature encoder: temporal convolution followed by a GELU activation function
Training: add noise to the audios so that the model learns to distinguish the voice
Decoder: 2 language models: 4-gram and transformer



Results

Announced: **WER=1.8** on clear data, **WER=3.3** on others

Our test: on 582 audios of 10s, **WER=21.7%** (because of the punctuation)

Conclusion

- Construct and test our own models
- Familiarize ourselves with traditional ASR
- Understand state-of-the-art algorithm

- Not usable results because insufficient processing power
- Transcriptions are very slow (Real Time Factor = 1)