FOUNDERS AND CODERS

Week 5

Machine Learning Institute

Audio is all you need

Adding another modality to our tool box

Task One: Classify


A Dataset and Taxonomy for Urban Sound Research

Justin Salamon[1,2], Christopher Jacoby[1], Juan Pablo Bello[1]
[1]Music and Audio Research Laboratory, New York University
[2]Center for Urban Science and Progress, New York University
{justin.salamon, cbj238, jpbello}@nyu.edu

ABSTRACT

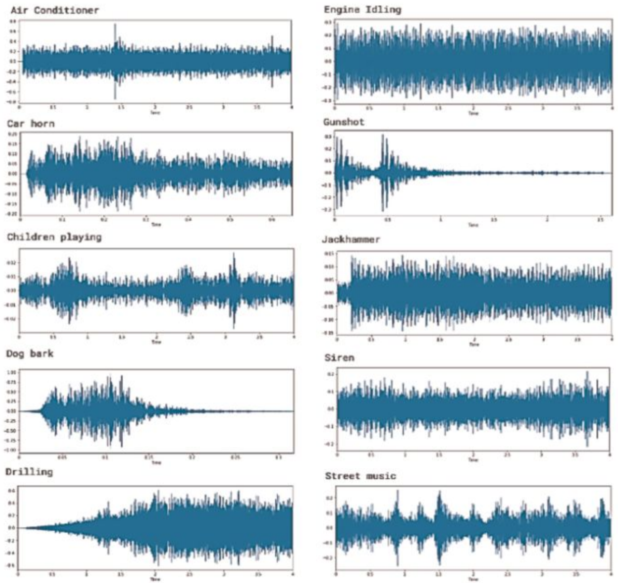
Categories and Subject Descriptors

Keywords

1. INTRODUCTION

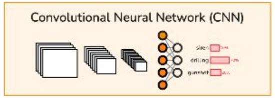
2. URBAN SOUND TAXONOMY

Salamon et al. 2014

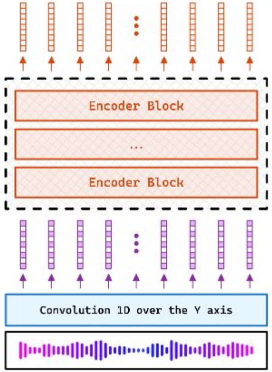


Clearly images! :)

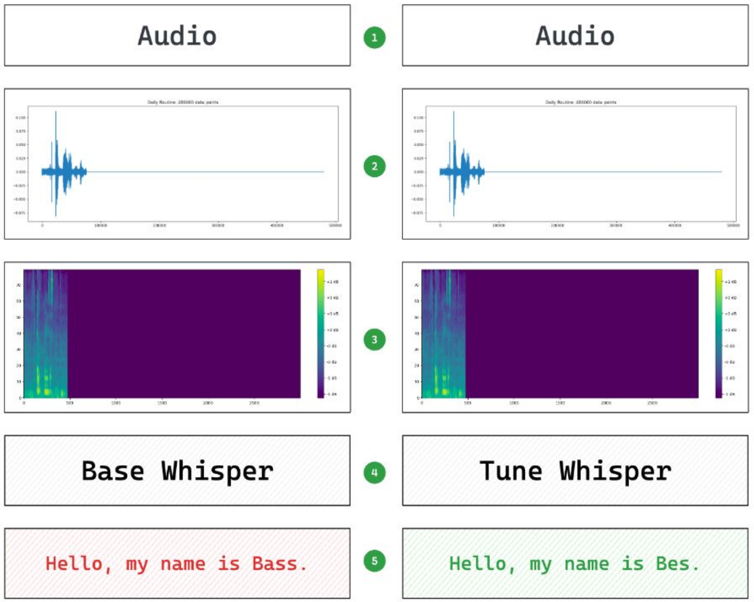
Classify using a CNN

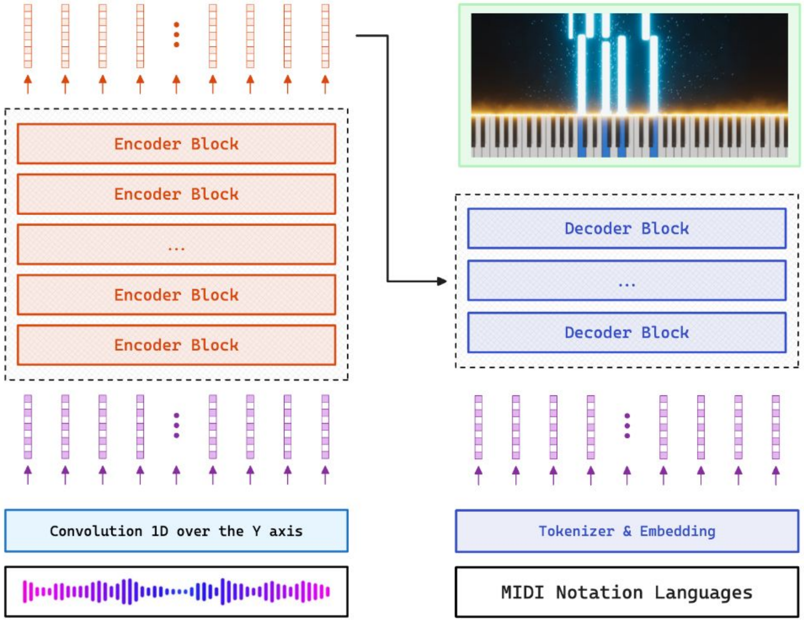


Classify using a Transformer

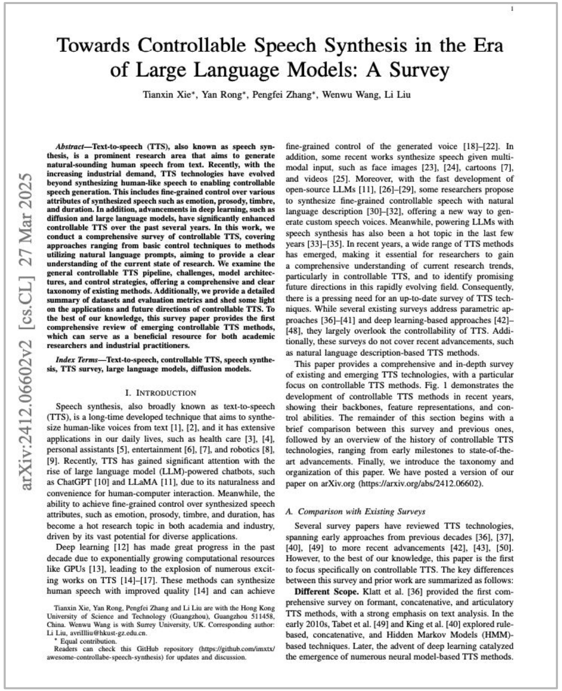


Task Two: Fine-tune





TTS Survey



arXiv:2412.06602v2 [cs.CL] 27 Mar 2025

1

Towards Controllable Speech Synthesis in the Era of Large Language Models: A Survey

Tianxin Xie*, Yan Rong*, Pengfei Zhang*, Wenwu Wang, Li Liu

***Abstract*—Text-to-speech (TTS), also known as speech synthesis, is a prominent research area that aims to generate natural-sounding human speech from text. Recently, with the increasing industrial demand, TTS technologies have evolved beyond synthesizing human-like speech to enabling controllable speech generation. This includes fine-grained control over various attributes of synthesized speech such as emotion, prosody, timbre, and duration. In addition, advancements in deep learning, such as diffusion and large language models, have significantly enhanced controllable TTS over the past several years. In this work, we conduct a comprehensive survey of controllable TTS, covering approaches ranging from basic control techniques to methods utilizing natural language prompts, aiming to provide a clear understanding of the current state of research. We examine the general controllable TTS pipeline, challenges, model architectures, and control strategies, offering a comprehensive and clear taxonomy of existing methods. Additionally, we provide a detailed summary of datasets and evaluation metrics and shed some light on the applications and future directions of controllable TTS. To the best of our knowledge, this survey paper provides the first comprehensive review of emerging controllable TTS methods, which can serve as a beneficial resource for both academic researchers and industrial practitioners.**

***Index Terms*—Text-to-speech, controllable TTS, speech synthesis, TTS survey, large language models, diffusion models.**

I. Introduction

Speech synthesis, also broadly known as text-to-speech (TTS), is a long-time developed technique that aims to synthesize human-like voices from text [1], [2], and it has extensive applications in our daily lives, such as health care [3], [4], personal assistants [5], entertainment [6], [7], and robotics [8], [9]. Recently, TTS has gained significant attention with the rise of large language model (LLM)-powered chatbots, such as ChatGPT [10] and LLaMA [11], due to its naturalness and convenience for human-computer interaction. Meanwhile, the ability to achieve fine-grained control over synthesized speech attributes, such as emotion, prosody, timbre, and duration, has become a hot research topic in both academia and industry, driven by its vast potential for diverse applications.

Deep learning [12] has made great progress in the past decade due to exponentially growing computational resources like GPUs [13], leading to the explosion of numerous exciting works on TTS [14]–[17]. These methods can synthesize human speech with improved quality [14] and can achieve fine-grained control of the generated voice [18]–[22]. In addition, some recent works synthesize speech given multi-modal input, such as face images [23], [24], cartoons [7], and videos [25]. Moreover, with the fast development of open-source LLMs [11], [26]–[29], some researchers propose to synthesize fine-grained controllable speech with natural language description [30]–[32], offering a new way to generate custom speech voices. Meanwhile, powering LLMs with speech synthesis has also been a hot topic in the last few years [33]–[35]. In recent years, a wide range of TTS methods has emerged, making it essential for researchers to gain a comprehensive understanding of current research trends, particularly in controllable TTS, and to identify promising future directions in this rapidly evolving field. Consequently, there is a pressing need for an up-to-date survey of TTS techniques. While several existing surveys address parametric approaches [36]–[41] and deep learning-based approaches [42]–[48], they largely overlook the controllability of TTS. Additionally, these surveys do not cover recent advancements, such as natural language description-based TTS methods.

This paper provides a comprehensive and in-depth survey of existing and emerging TTS technologies, with a particular focus on controllable TTS methods. Fig. 1 demonstrates the development of controllable TTS methods in recent years, showing their backbones, feature representations, and control abilities. The remainder of this section begins with a brief comparison between this survey and previous ones, followed by an overview of the history of controllable TTS technologies, ranging from early milestones to state-of-the-art advancements. Finally, we introduce the taxonomy and organization of this paper. We have posted a version of our paper on arXiv.org (https://arxiv.org/abs/2412.06602).

A. Comparison with Existing Surveys

Several survey papers have reviewed TTS technologies, spanning early approaches from previous decades [36], [37], [40], [49] to more recent advancements [42], [43], [50]. However, to the best of our knowledge, this paper is the first to focus specifically on controllable TTS. The key differences between this survey and prior work are summarized as follows:

Different Scope. Klatt et al. [36] provided the first comprehensive survey on formant, concatenative, and articulatory TTS methods, with a strong emphasis on text analysis. In the early 2010s, Tabet et al. [49] and King et al. [40] explored rule-based, concatenative, and Hidden Markov Models (HMM)-based techniques. Later, the advent of deep learning catalyzed the emergence of numerous neural model-based TTS methods.

Tianxin Xie, Yan Rong, Pengfei Zhang and Li Liu are with the Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511458, China. Wenwu Wang is with Surrey University, UK. Corresponding author: Li Liu, avrillliu@hkust-gz.edu.cn.

* Equal contribution.

Readers can check this GitHub repository (https://github.com/imxtx/awesome-controllabe-speech-synthesis) for updates and discussion.

Xie et al. 2025

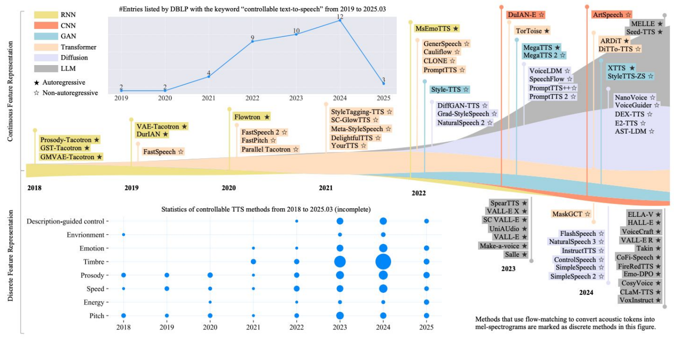


Fig. 1. A summary of representative controllable TTS methods in recent years and their model architectures, feature representations, and control abilities. Additional network structures, such as VAE and flow-based models, are not included in this figure. For more details, refer to Tables IV and III.

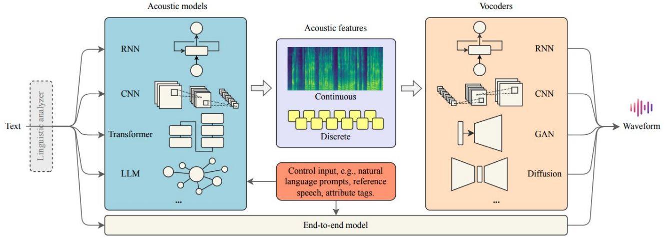
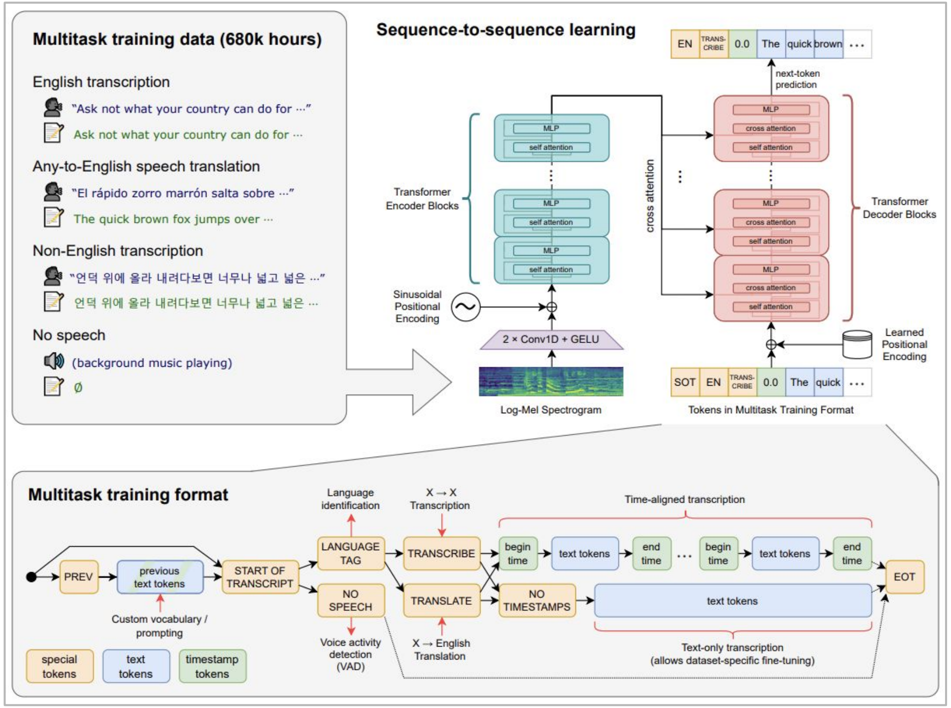
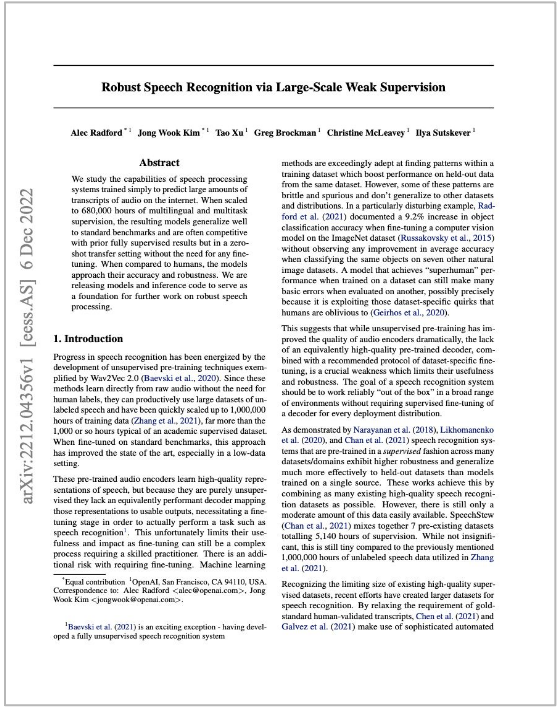


Fig. 2. General pipeline of controllable TTS from the perspective of network structure. Linguistic analysis is necessary for parametric and a few neural methods but is no longer needed for most modern neural methods. In this paper, we only review neural model-based controllable TTS methods and do not investigate acoustic features (e.g., MFCC [107], LSP [108], F0 [109]) used in early TTS methods.

Whisper





arXiv:2212.04356v1 [eess.AS] 6 Dec 2022

Robust Speech Recognition via Large-Scale Weak Supervision

Alec Radford [*1] Jong Wook Kim [*1] Tao Xu [1] Greg Brockman [1] Christine McLeavey [1] Ilya Sutskever [1]

Abstract

We study the capabilities of speech processing systems trained simply to predict large amounts of transcripts of audio on the internet. When scaled to 680,000 hours of multilingual and multitask supervision, the resulting models generalize well to standard benchmarks and are often competitive with prior fully supervised results but in a zero-shot transfer setting without the need for any fine-tuning. When compared to humans, the models approach their accuracy and robustness. We are releasing models and inference code to serve as a foundation for further work on robust speech processing.

1. Introduction

Progress in speech recognition has been energized by the development of unsupervised pre-training techniques exemplified by Wav2Vec 2.0 (Baevski et al., 2020). Since these methods learn directly from raw audio without the need for human labels, they can productively use large datasets of unlabeled speech and have been quickly scaled up to 1,000,000 hours of training data (Zhang et al., 2021), far more than the 1,000 or so hours typical of an academic supervised dataset. When fine-tuned on standard benchmarks, this approach has improved the state of the art, especially in a low-data setting.

These pre-trained audio encoders learn high-quality representations of speech, but because they are purely unsupervised they lack an equivalently performant decoder mapping those representations to usable outputs, necessitating a fine-tuning stage in order to actually perform a task such as speech recognition[1]. This unfortunately limits their usefulness and impact as fine-tuning can still be a complex process requiring a skilled practitioner. There is an additional risk with requiring fine-tuning. Machine learning methods are exceedingly adept at finding patterns within a training dataset which boost performance on held-out data from the same dataset. However, some of these patterns are brittle and spurious and don't generalize to other datasets and distributions. In a particularly disturbing example, Radford et al. (2021) documented a 9.2% increase in object classification accuracy when fine-tuning a computer vision model on the ImageNet dataset (Russakovsky et al., 2015) without observing any improvement in average accuracy when classifying the same objects on seven other natural image datasets. A model that achieves "superhuman" performance when trained on a dataset can still make many basic errors when evaluated on another, possibly precisely because it is exploiting those dataset-specific quirks that humans are oblivious to (Geirhos et al., 2020).

This suggests that while unsupervised pre-training has improved the quality of audio encoders dramatically, the lack of an equivalently high-quality pre-trained decoder, combined with a recommended protocol of dataset-specific fine-tuning, is a crucial weakness which limits their usefulness and robustness. The goal of a speech recognition system should be to work reliably "out of the box" in a broad range of environments without requiring supervised fine-tuning of a decoder for every deployment distribution.

As demonstrated by Narayanan et al. (2018), Likhomanenko et al. (2020), and Chan et al. (2021) speech recognition systems that are pre-trained in a *supervised* fashion across many datasets/domains exhibit higher robustness and generalize much more effectively to held-out datasets than models trained on a single source. These works achieve this by combining as many existing high-quality speech recognition datasets as possible. However, there is still only a moderate amount of this data easily available. SpeechStew (Chan et al., 2021) mixes together 7 pre-existing datasets totalling 5,140 hours of supervision. While not insignificant, this is still tiny compared to the previously mentioned 1,000,000 hours of unlabeled speech data utilized in Zhang et al. (2021).

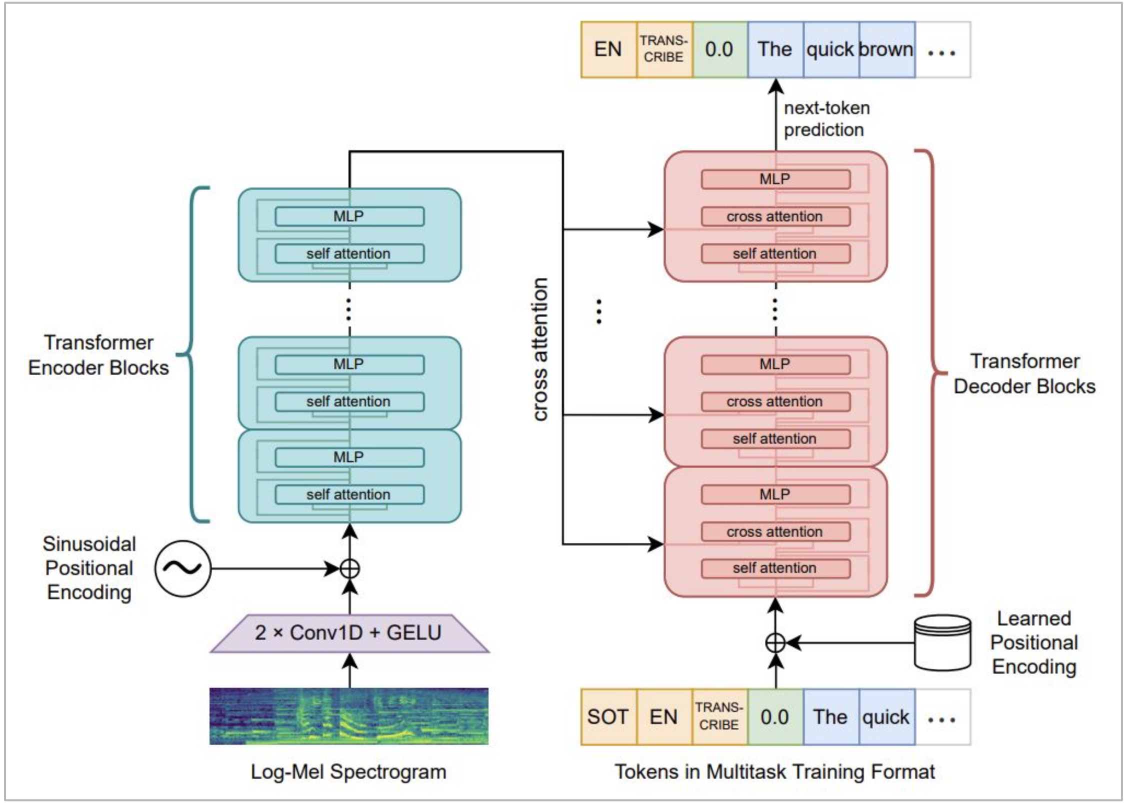
Recognizing the limiting size of existing high-quality supervised datasets, recent efforts have created larger datasets for speech recognition. By relaxing the requirement of gold-standard human-validated transcripts, Chen et al. (2021) and Galvez et al. (2021) make use of sophisticated automated

[*]Equal contribution [1]OpenAI, San Francisco, CA 94110, USA. Correspondence to: Alec Radford <alec@openai.com>, Jong Wook Kim <jongwook@openai.com>.

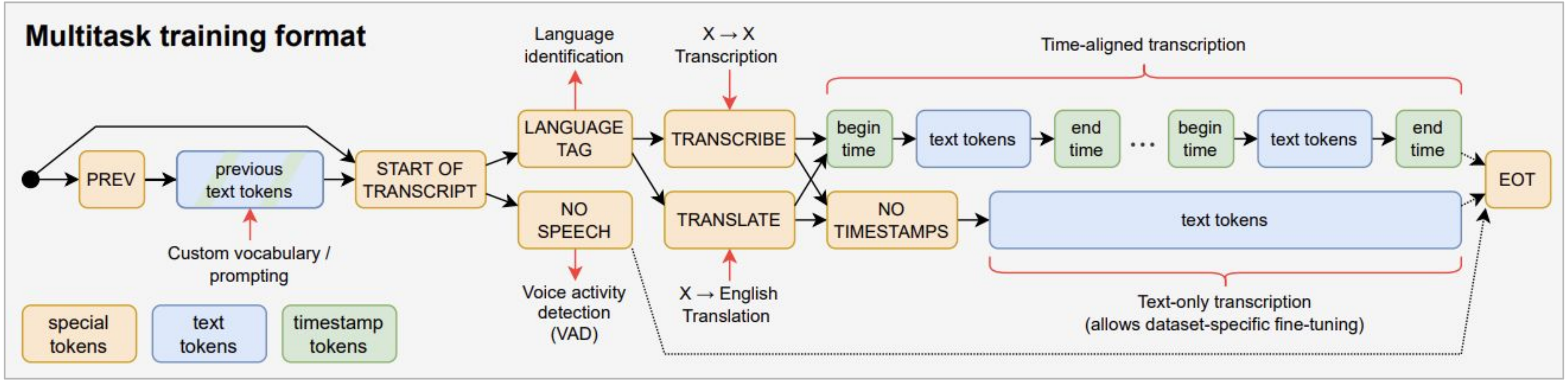
[1]Baevski et al. (2021) is an exciting exception - having developed a fully unsupervised speech recognition system

Radford et al. 2022

Transformer



Transformer



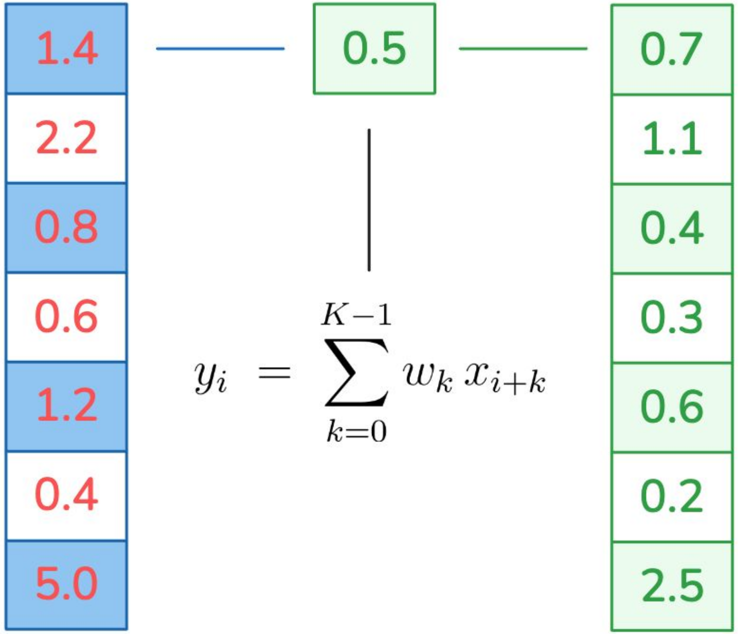
`<|startoftranscript|><|en|><|transcribe|><|notimestamps|>Hello, my name is Bes.<|endoftext|>`

`[50258, 50259, 50359, 50363, 15947, 11, 452, 1315, 307, 8190, 13, 50257]`

FOUNDERS AND CODERS

Convolutions

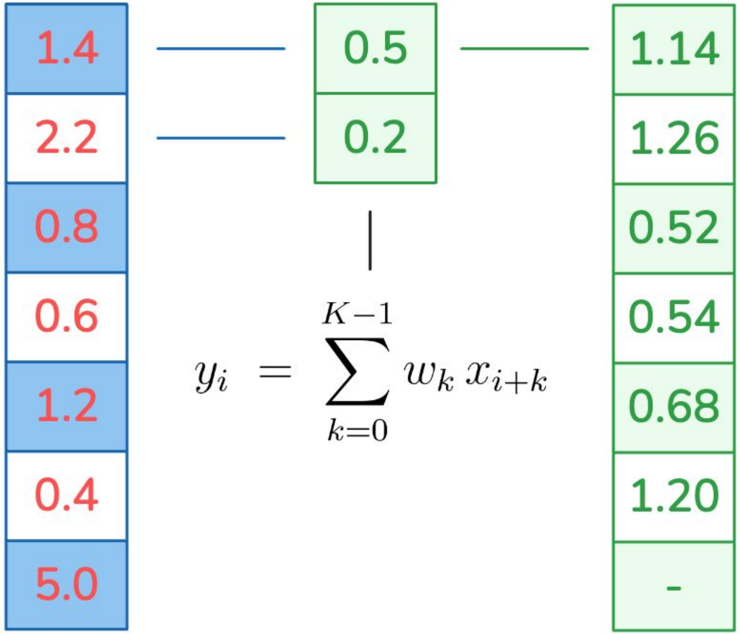
Machine Learning Institute



$$y_i = \sum_{k=0}^{K-1} w_k \, x_{i+k}$$

```python
#
#
import torch

#
#
cov = torch.nn.Conv1d(
    in_channels=1,
    out_channels=1,
    kernel_size=1,
    padding=0,
    bias=False
)

#
#
with torch.no_grad():
    cov.weight[:] = 0.5

#
#
bar = torch.tensor([[1.4, 2.2, 0.8, 0.6, 1.2, 0.4, 5.0]])
out = cov(bar)
print("Out:", out)

```

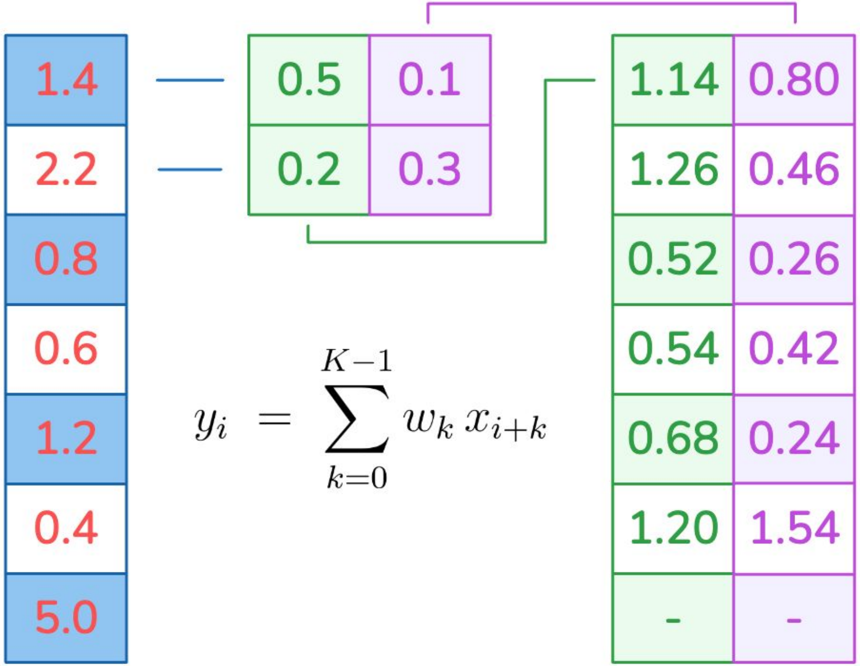
Convolutions



$$y_i = \sum_{k=0}^{K-1} w_k \, x_{i+k}$$

```python
#
#
import torch

#
#
cov = torch.nn.Conv1d(
    in_channels=1,
    out_channels=1,
    kernel_size=2,
    padding=0,
    bias=False
)

#
#
with torch.no_grad():
    cov.weight[:] = torch.tensor([[[0.5, 0.2]]])

#
#
bar = torch.tensor([[1.4, 2.2, 0.8, 0.6, 1.2, 0.4, 5.0]])
out = cov(bar)
print("Out:", out)

```

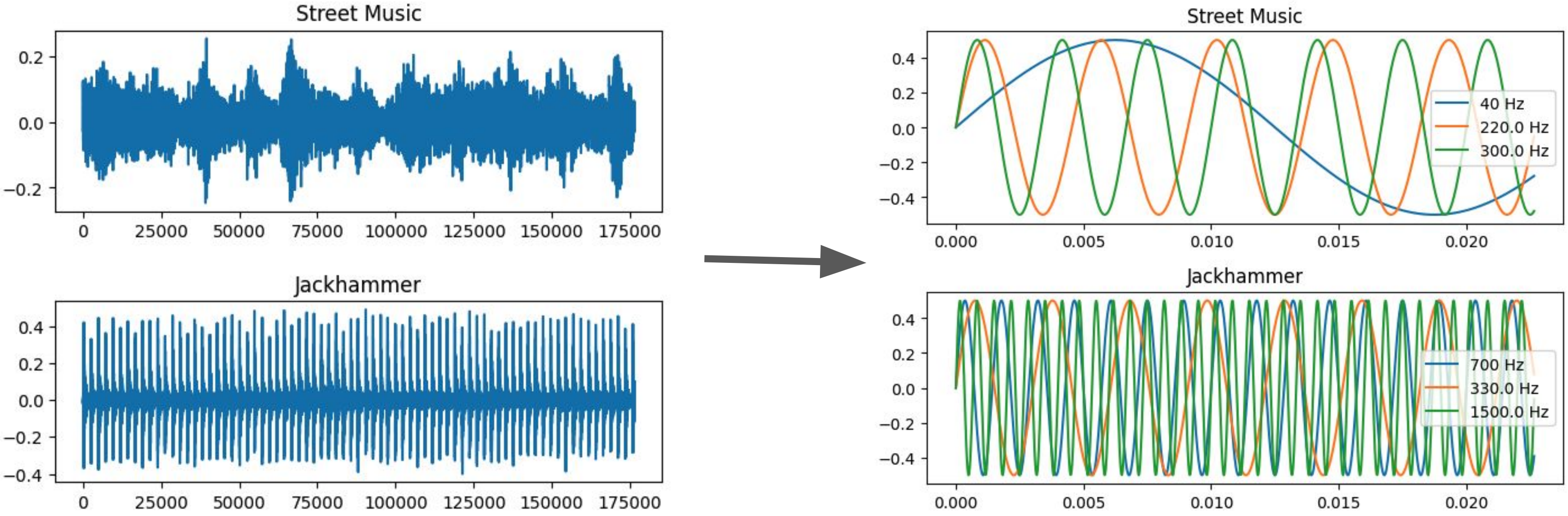
Convolutions



$$y_i = \sum_{k=0}^{K-1} w_k \, x_{i+k}$$

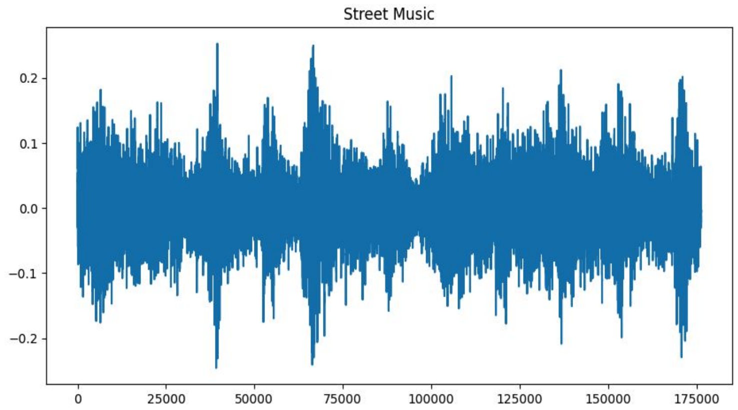
```python
#
#
import torch

#
#
cov = torch.nn.Conv1d(
    in_channels=1,
    out_channels=2,
    kernel_size=2,
    padding=0,
    bias=False
)

#
#
with torch.no_grad():
    cov.weight[0, 0] = torch.tensor([0.5, 0.2])
    cov.weight[1, 0] = torch.tensor([0.1, 0.3])

#
#
bar = torch.tensor([[1.4, 2.2, 0.8, 0.6, 1.2, 0.4, 5.0]])
out = cov(bar)

```

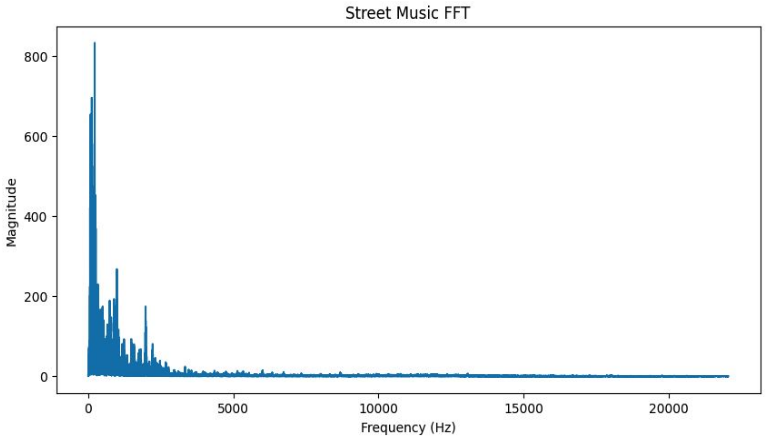
What is audio?



- Sound is a bundle of vibrations. It is complex
- As usual we need to extract features

Fourier Transform





- Fourier Transform unbundles audio by going through each frequency to test how much each frequency is contributing to the overall signal
- In practice we use something called the Discrete Fourier Transform

FOUNDERS AND CODERS

Tune Code

Machine Learning Institute

```python
#
#
#
import torch
import whisper


#
#
#
model = whisper.load_model('tiny')
audio = whisper.load_audio('name.wav')
audio = whisper.pad_or_trim(audio)
lg_ml = whisper.log_mel_spectrogram(audio)
tknsr = whisper.tokenizer.get_tokenizer(multilingual=True)


#
#
#
opt = whisper.DecodingOptions()
res = whisper.decode(model, lg_ml.to(model.device), opt)
print('Baseline:', res.text) # Hello my name is Bass.
print('------')

#
#
#
ids = []
ids += [tknsr.sot]
ids += [tknsr.language_token]
ids += [tknsr.transcribe]
ids += [tknsr.no_timestamps]
ids += tknsr.encode(' Hello, my name is Bes.')
ids += [tknsr.eot]


#
#
#
optimizer = torch.optim.Adam(model.parameters(), lr=0.00001)
criterion = torch.nn.CrossEntropyLoss()


#
#
#
model.train()
tks = torch.tensor(ids).unsqueeze(0).to(model.device)
mel = whisper.log_mel_spectrogram(audio).unsqueeze(0).to(model.device)
#
#
pred = model(tokens=tks, mel=mel)
trgt = tks[:, 1:].contiguous()
pred = pred[:, :-1, :].contiguous()


#
#
#
print('Ids Target:', trgt.squeeze().tolist())
print('Ids Output:', torch.argmax(pred, dim=-1).squeeze().tolist())
print('Txt Target:', tknsr.decode(trgt.squeeze().tolist()))
print('Txt Output:', tknsr.decode(torch.argmax(pred, dim=-1).squeeze().tolist()))


#
#
#
loss = criterion(pred.transpose(1, 2), trgt)
print('Loss:', loss.item())
print('------')
optimizer.zero_grad()
loss.backward()
optimizer.step()


#
#
#
model.eval()
prd = model(tokens=tks, mel=mel)
prd = prd[:, :-1, :].contiguous()


#
#
#
print('Ids Target:', trgt.squeeze().tolist())
print('Ids Output:', torch.argmax(prd, dim=-1).squeeze().tolist())
print('Txt Target:', tknsr.decode(trgt.squeeze().tolist()))
print('Txt Output:', tknsr.decode(torch.argmax(prd, dim=-1).squeeze().tolist()))
loss = criterion(prd.transpose(1, 2), trgt)
print('Loss:', loss.item())
```

Tune Result

```
00: torch.Size([1, 80, 3000])
Baseline: Hello, my name is Bass.
==========================================================================
00: torch.Size([1, 80, 3000])
Ids Target: [50259, 50359, 50363, 2425, 11, 452, 1315, 307, 8190, 13, 50257]
Ids Output: [50259, 50359, 50363, 2425, 11, 452, 1315, 307, 29626, 13, 50257]
Txt Target: <|en|><|transcribe|><|notimestamps|> Hello, my name is Bes.<|endoftext|>
Txt Output: <|en|><|transcribe|><|notimestamps|> Hello, my name is Bass.<|endoftext|>
Loss: 0.5395039916038513
==========================================================================
00: torch.Size([1, 80, 3000])
Ids Target: [50259, 50359, 50363, 2425, 11, 452, 1315, 307, 8190, 13, 50257]
Ids Output: [50259, 50359, 50363, 2425, 11, 452, 1315, 307, 8190, 13, 50257]
Txt Target: <|en|><|transcribe|><|notimestamps|> Hello, my name is Bes.<|endoftext|>
Txt Output: <|en|><|transcribe|><|notimestamps|> Hello, my name is Bes.<|endoftext|>
Loss: 0.1766674667596817
```

FOUNDERS AND CODERS

Good luck!

Machine Learning Institute

Recurrent Rebels

Ewan Beattie
Tyrone Nicholas
Hikaru Tsujimura
Maria Sharif

Gradient Gigglers

Yali Pan
Nikolas Kuhn
Ben Williams
Aparna Pillai

Overfitting Overlords

Adam Beedell
Jingyan Chen
Tao Zamorano
Charles Cai

Hyperparameter Hippies

Jacob Jenner
Miguel Parracho
Esperanza Shi
Arjuna James

Perceptron Party

Clement Ha
Helen Zhou
Ethan Edwards
Rasched Haidari

Backprop Bunch

Ben Liong
Andrei Zhirnov
Kadriye Turkcan
Prima Gouse

Dropout Disco

Umut Sagir
Peter O'Keeffe
Marcin Tolysz
Tomas Krajcoviech

Kernel Kittens

Dan Goss
Joao Esteves
James Carter
James Yan

Bayesian Buccaneers

Andrew
David Edev
Melanie Wong
Anton Dergunov

Feature Fiestas

Ben Bethell
Rosh Beed
Felipe Lavratti