

---

# Speech Translation

Philipp Koehn  
based on slides from Xutai Ma

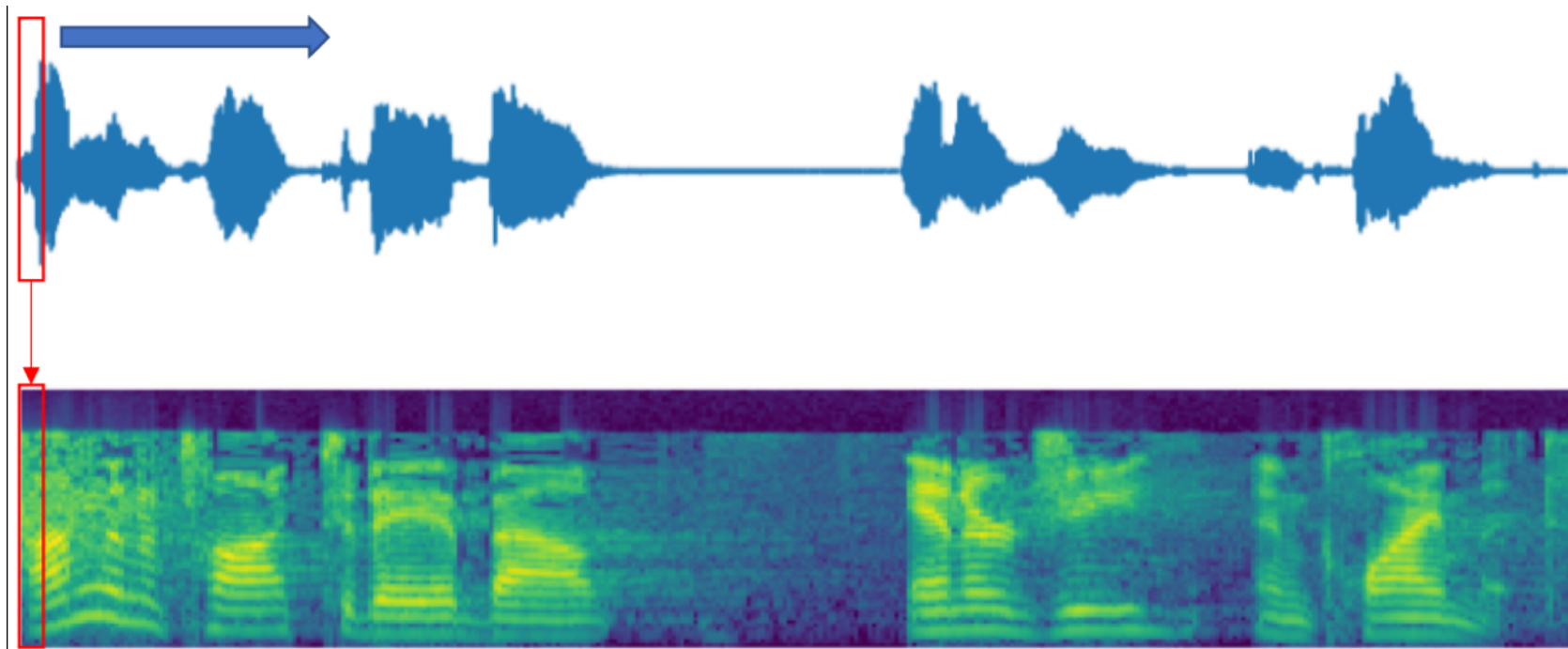
14 November 2023



# What is Speech?



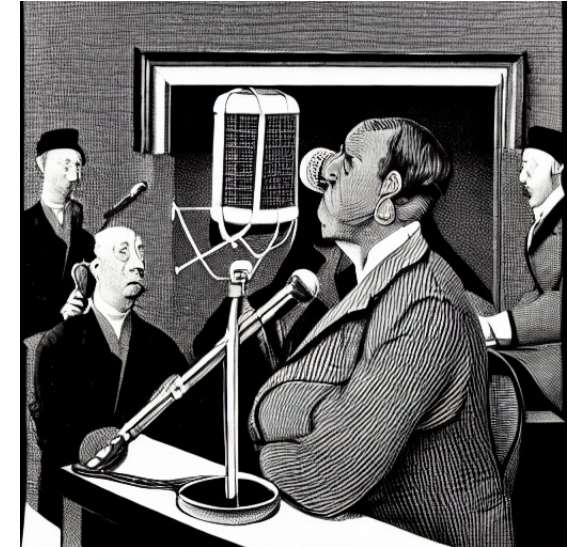
1



- Spectrogram: Loudness at different sound frequencies and time steps
  - Typically segmented into, say, 50 frames per second (50Hz)
- ⇒ can be used in sequence models

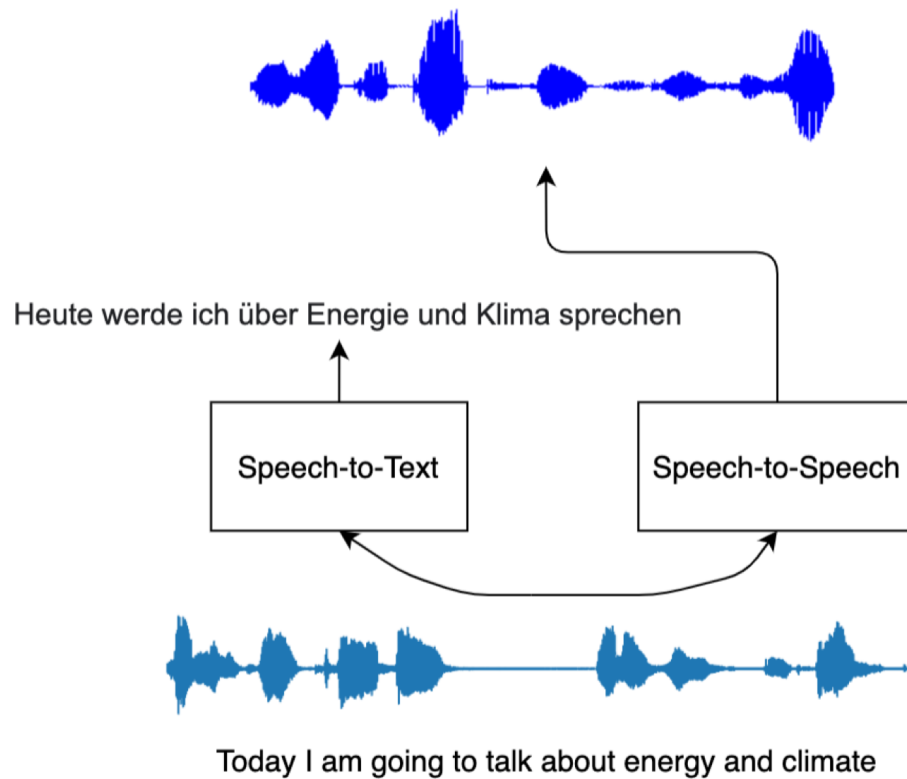
# What Makes Speech Hard?

- Disfluent language
  - Ungrammatical languages, restarts, repetitions
  - Pauses, filler words ("ah", "hm", "like"), ■
- Noise
  - background sounds, reverberation, etc.
  - cocktail party effect ■
- Recording conditions
  - sampling rate
  - which sound frequencies are filtered out
  - microphone placement, possibly multiple microphones ■
- More variety
  - different speakers
  - more spoken language varieties
  - tone, emotional content not captured in text ■
- Less data (also more privacy concerns about data)



# Introduction

- What is speech translation?
  - Translate speech in source language to text / speech in target language





# Introduction

Why/Where do we need speech translation?

- International conferences (e.g., UN, EU)
- Live video translation (e.g., YouTube, streaming)
- Personal translator (e.g., international travels)
  - Google translate (Conversation)

# speech recognition

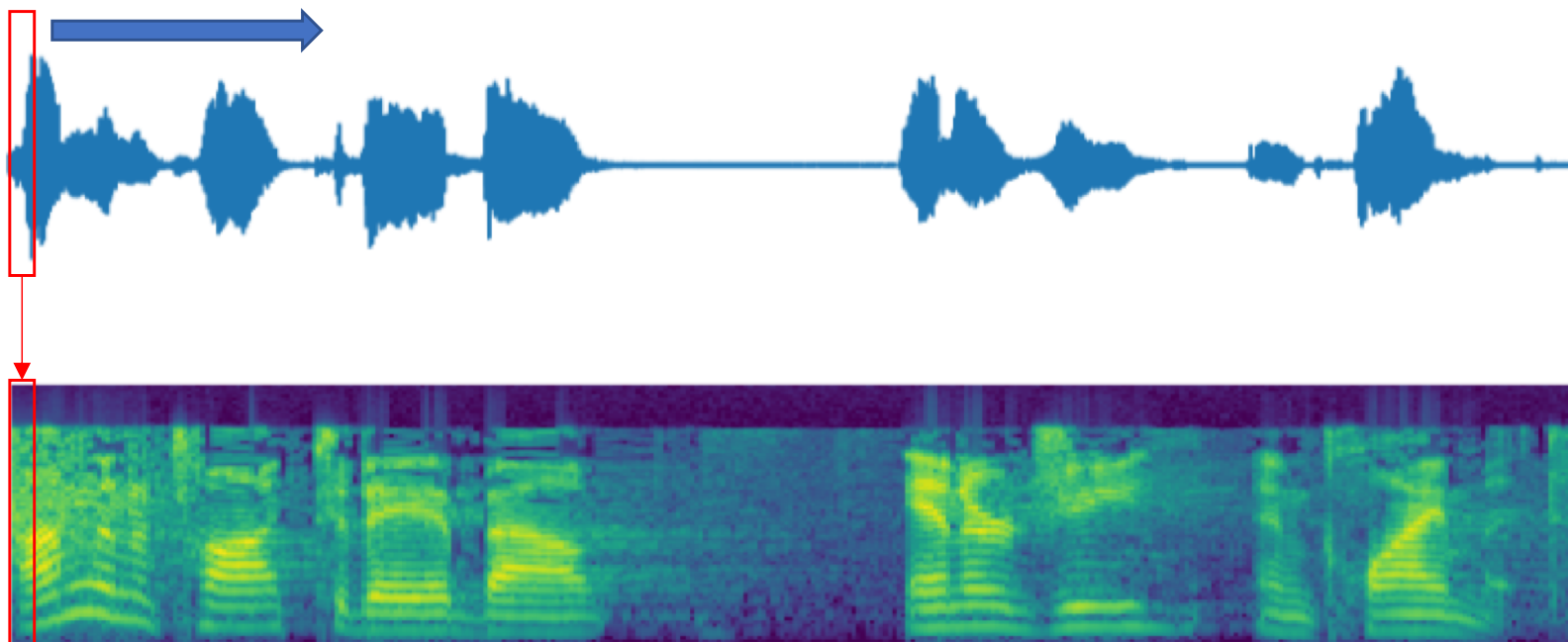
# Background:

## Speech Processing and Recognition

- Speech Processing
  - How to represent speech → feature extraction
- Automatic Speech Recognition (ASR)
  - Transcribe speech to text in one language
  - Seq2seq task, but input and output have the same order

# Feature Extractions

- Short-Term Spectrum
  - (Mel-frequency cepstral coefficients) MFCC
- Convert speech samples to sequence of vectors



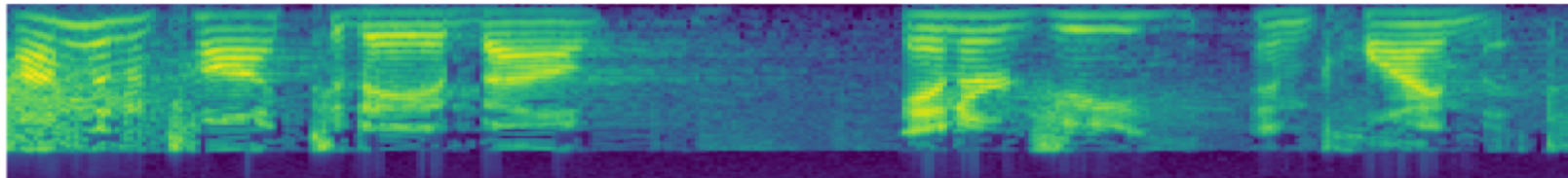
# Automatic Speech Recognition

- Acoustic Model
  - Neural-based models

Fully connect / recurrent layers

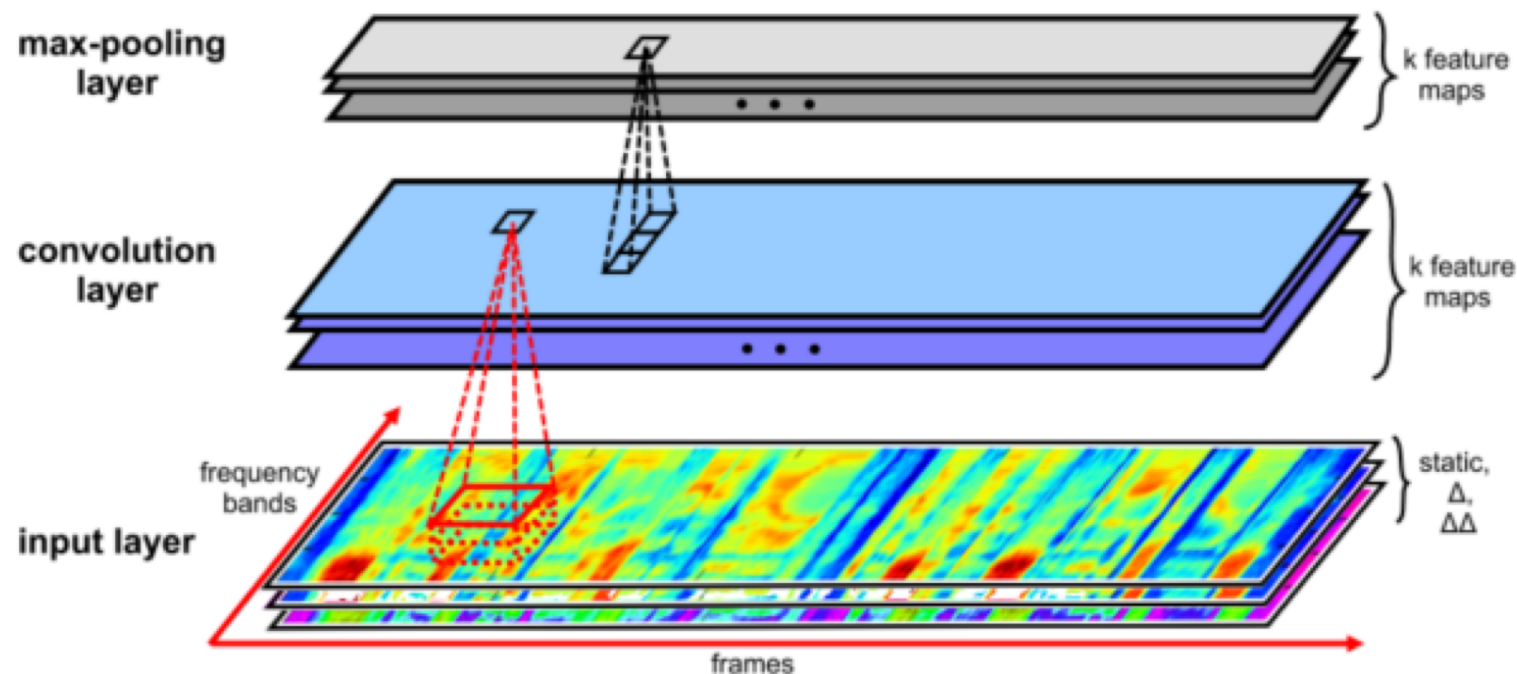
Pooling layers

Convolutional layers



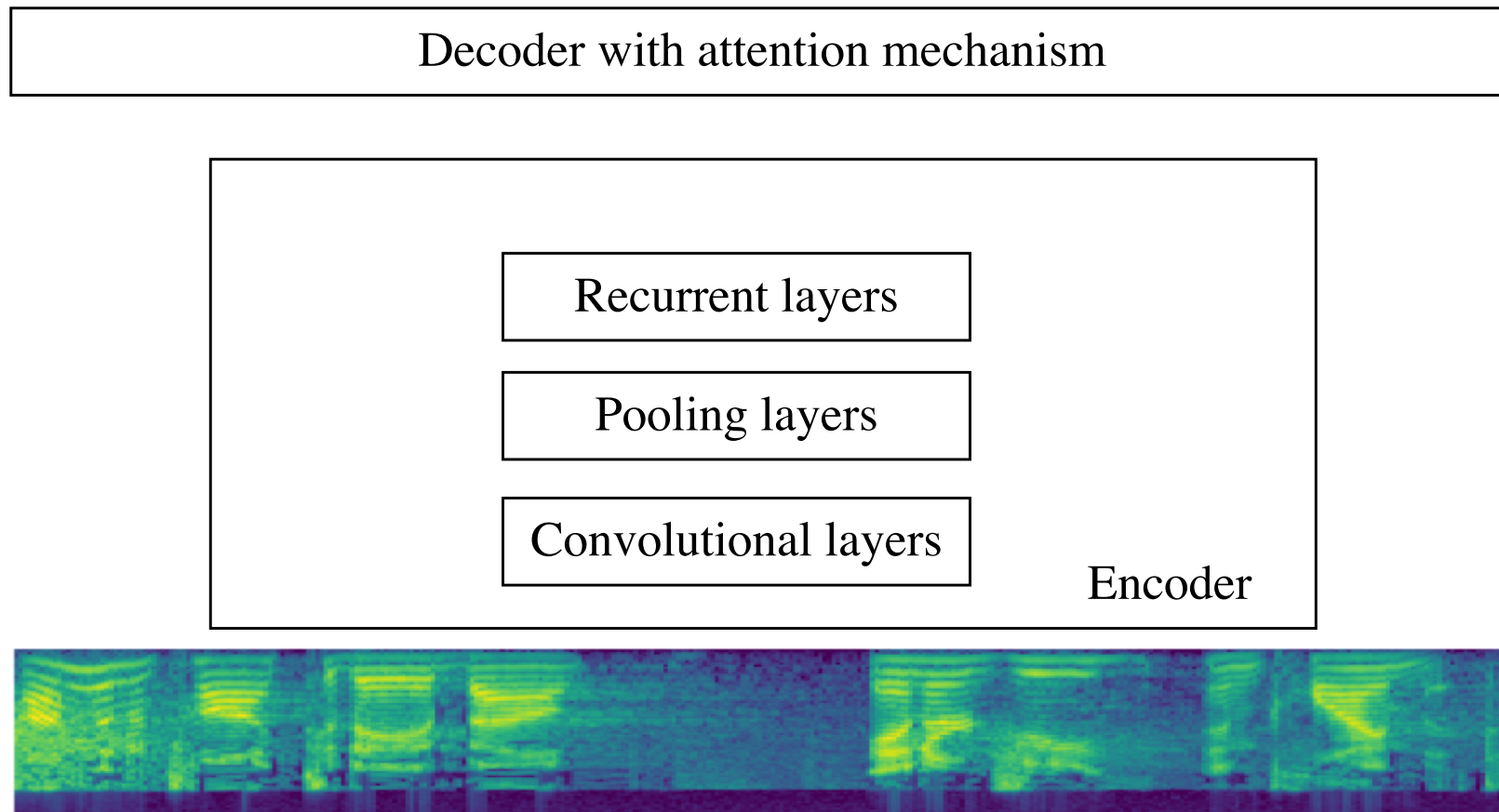
# Automatic Speech Recognition

- Convolutional layers



# Automatic Speech Recognition

- Seq2Seq model



# Automatic Speech Recognition



[Home](#) [Documentation](#) [Help!](#) [Models](#)

Kaldi's code lives at <https://github.com/kaldi-asr/kaldi>. To checkout (i.e. clone in the git terminology) the most recent changes, you can use this command `git clone https://github.com/kaldi-asr/kaldi` or follow the [github link](#) and click "Download in zip" on the github page (right hand side of the web page)

To browse the model builds that are available (not many), please click on [models](#).

If you have any suggestion of how to improve the site, please [contact me](#).



## Contact

[dpovey@gmail.com](mailto:dpovey@gmail.com)  
Phone: 425 247 4129  
(Daniel Povey)

## ESPnet: end-to-end speech processing toolkit



ESPnet is an end-to-end speech processing toolkit for automatic speech recognition and end-to-end text-to-speech.

README.md

### wav2letter++

FAILED [chat](#) [on github](#)

### Important Note:

wav2letter has been moved and con

Future wav2letter development will occur in Flashlight.

To build the old, pre-consolidation version of wav2letter, checkout the [wav2letter v0.2](#) release, which depends on the old [Flashlight v0.2](#) release. The [wav2letter-lua](#) project can be found on the [wav2letter-lua](#) branch, accordingly.

For more information on wav2letter++, see or cite [this arXiv paper](#).

### Recipes

README.md



license [MIT](#) release [v0.10.2](#) [build](#) failing [docs](#) failing

Fairseq(-py) is a sequence modeling toolkit that allows researchers and developers to train custom models for translation, summarization, language modeling and other text generation tasks.

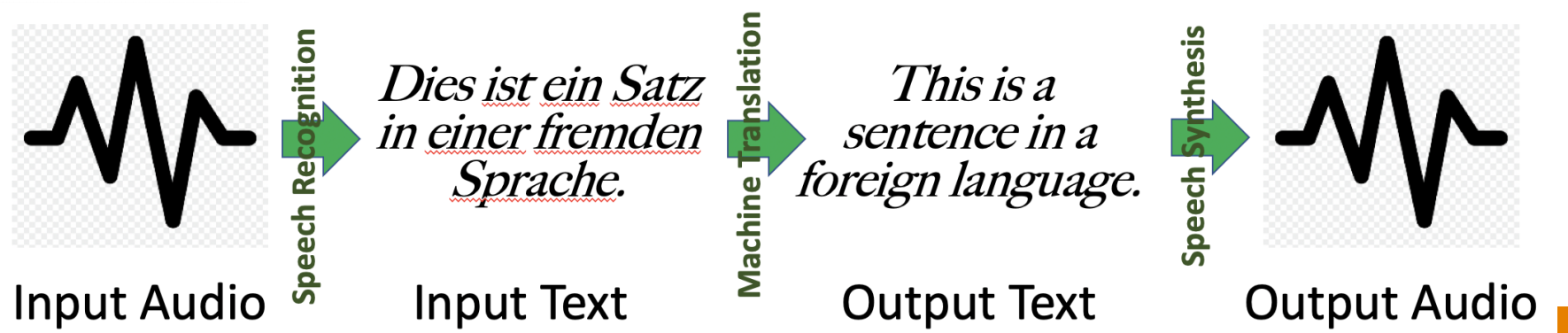
We provide reference implementations of various sequence modeling papers:



# speech translation

# Cascaded Speech Translation

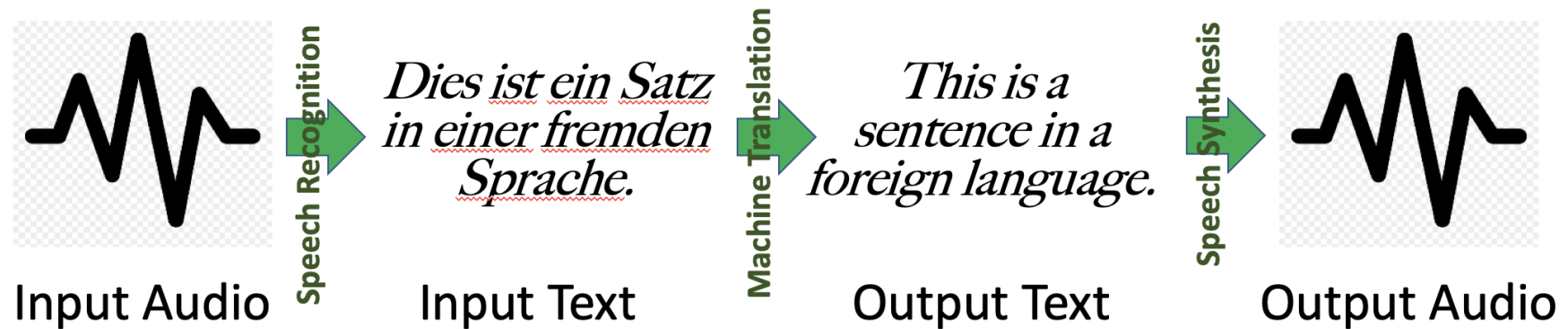
13



- Synchronize tokenization schemes
- Pass lattices between steps
- Main concern: error propagation

# End-to-End Systems

14



**Moving towards end-to-end systems:**



# Automatic Speech Recognition

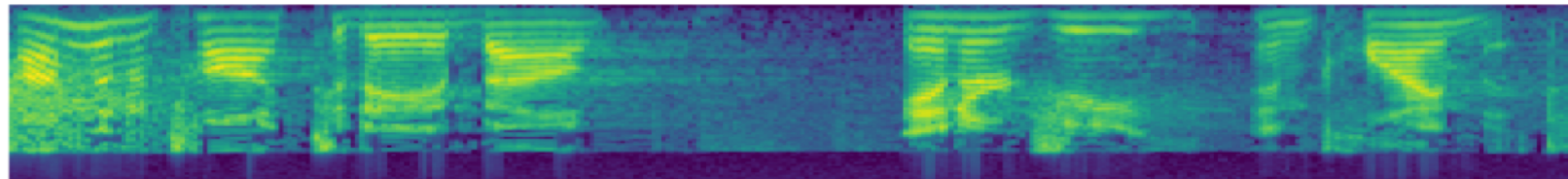
Decoder with attention mechanism

Recurrent layers

Pooling layers

Convolutional layers

Encoder



# End-to-End Speech Translation

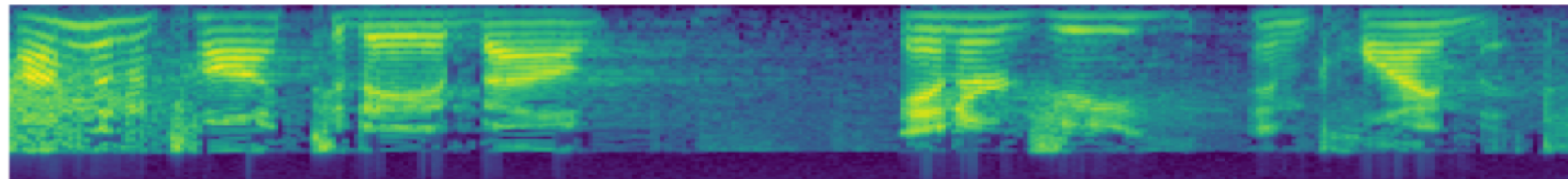
Decoder with attention mechanism

Recurrent layers

Pooling layers

Convolutional layers

Encoder



# The Data Aspect

17

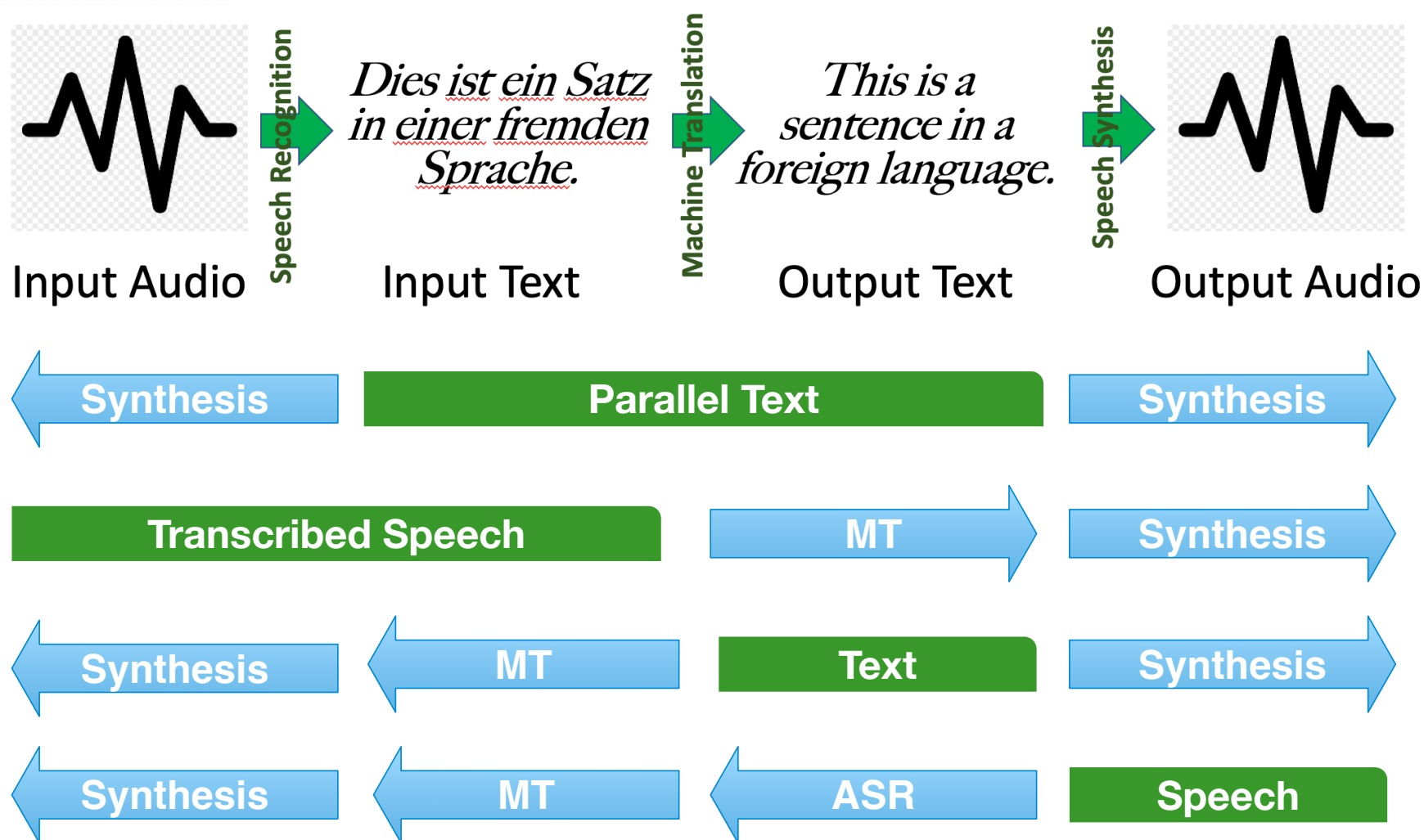


Moving towards end-to-end systems:

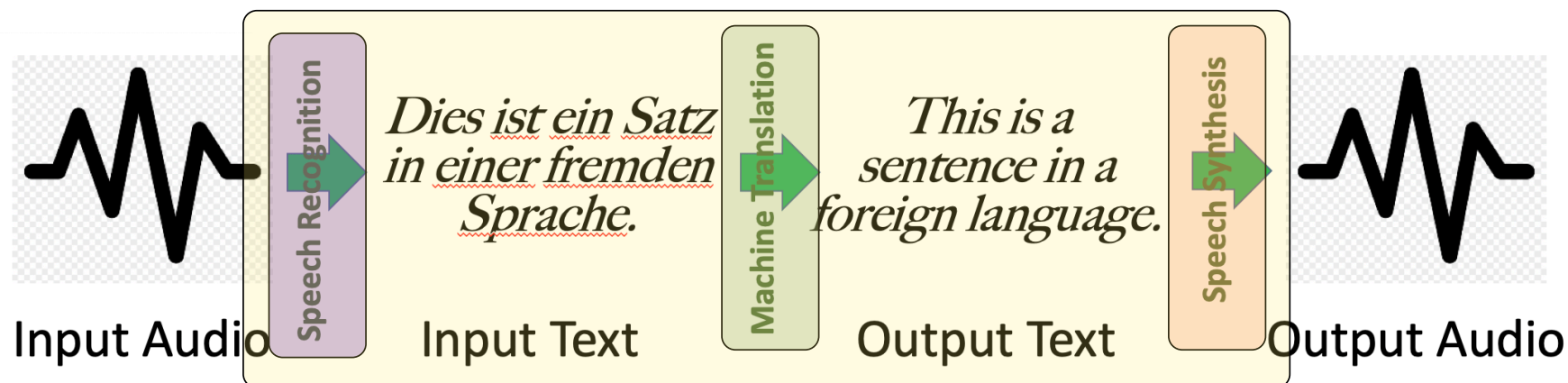


# Data Augmentation

18



# Pretraining Components



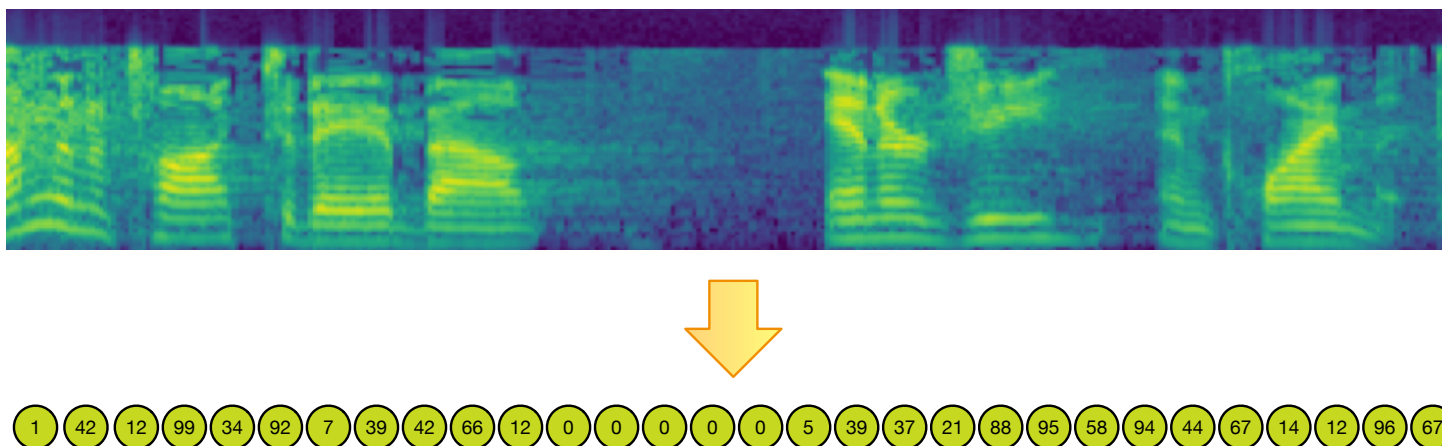
- Train components of the model separately
- Connect components
- Fine-tune on end-to-end data



# speech tokens

# Speech Tokens

- Goal: Represent speech as a sequence of discrete tokens



- Two Methods
  - Semantic tokens: wav2vec-BERT
  - Acoustic tokens: SpeechStream / SpeechStorm

# Semantic and Acoustic Tokens

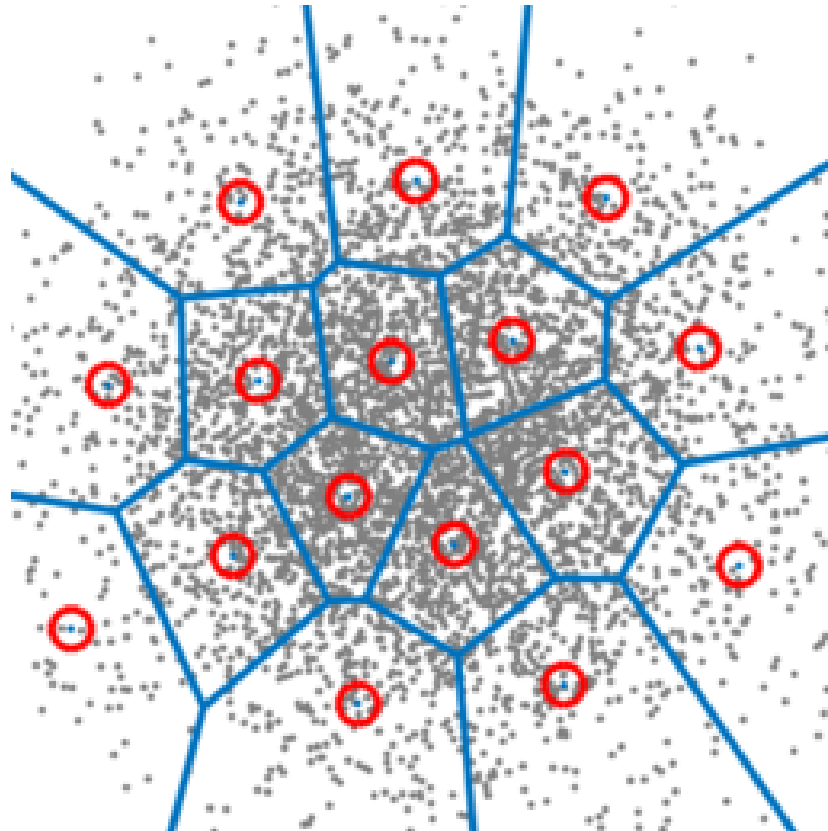
- Generating with semantic vs. acoustic tokens

	Acoustic	Semantic
Reconstruction	-	+
Phonetic discriminability	+	-

- Training model only on acoustic tokens creates “babbling”: no meaningful words

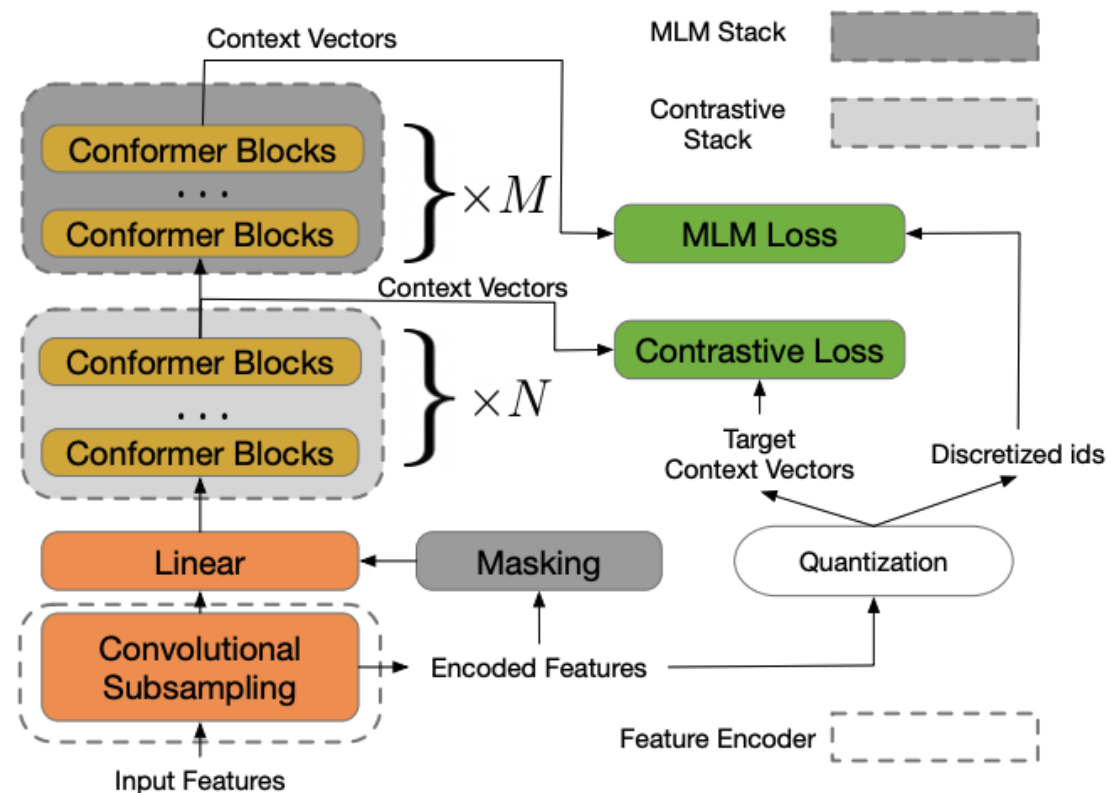
# Speech Tokens by Vector Quantization

- Input: high dimensional vector corresponding to a speech frame
- K-Means Clustering



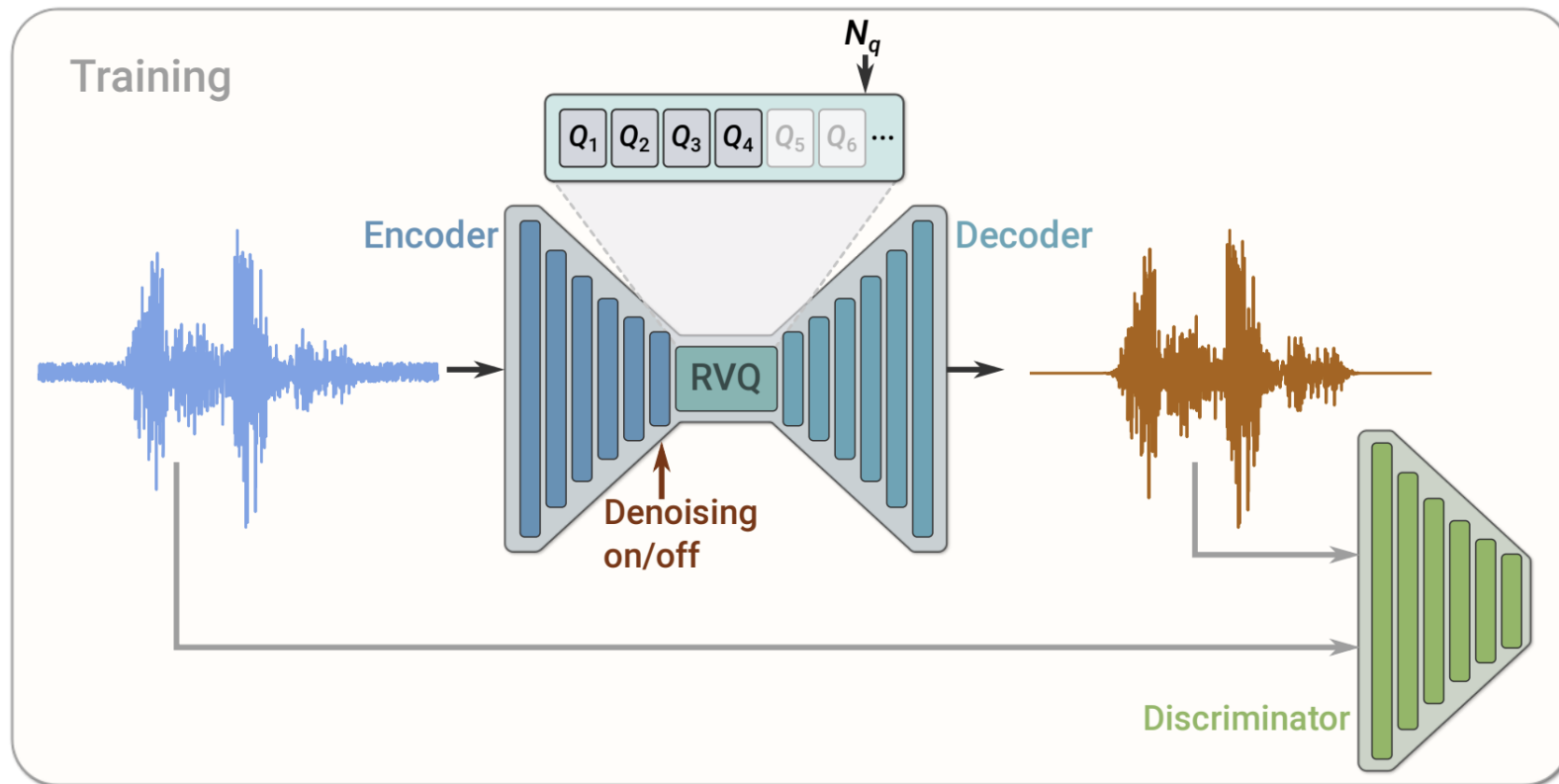
- Token is cluster ID

- Pre-trained model for speech, optimizing
  - masked language model loss on discrete tokens
  - contrastive loss: detect true vector from distractors (from same utterance)
  - also: codebook diversity loss (encourage uniform use of codes)



- Iteration between k-means clustering and retraining representations

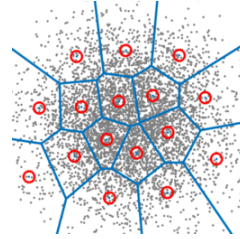
- A neural audio codec: goal is compression for transmitting less data



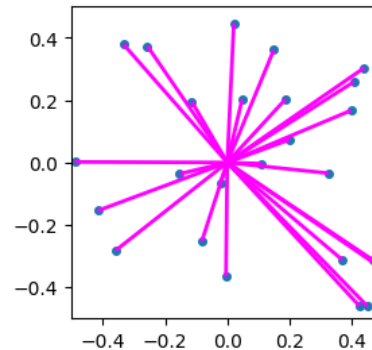
- Trained with reconstruction loss and discriminative training

# Residual Vector Quantization

- Vector Quantization: K-Means Clustering ( $\rightarrow$  token is cluster ID)



- Subtract centroid from **all** data points

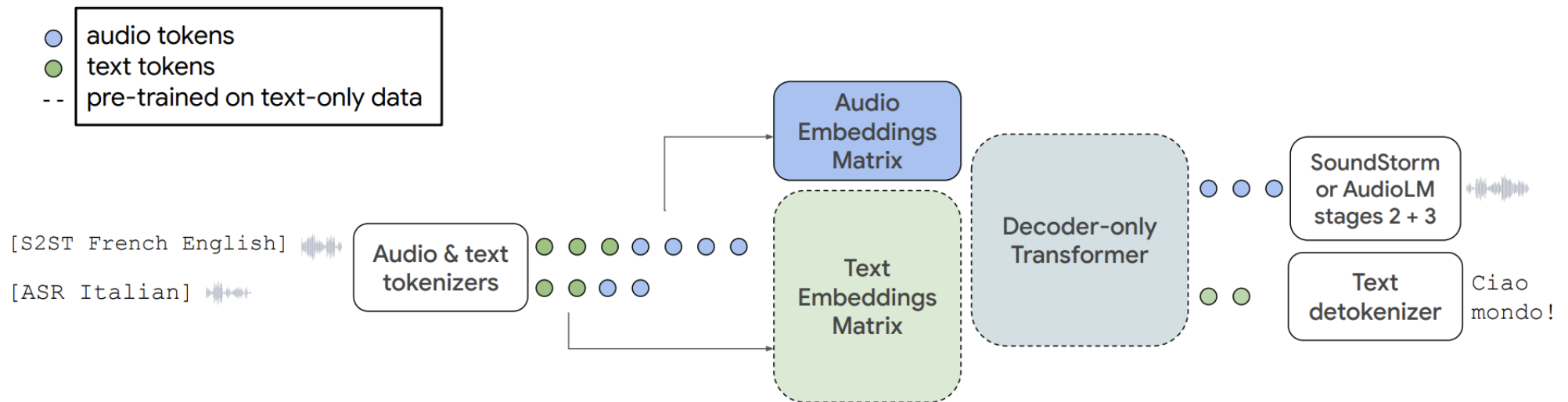


- Quantize **all** data points again (not hierarchical), rinse and repeat

(images from <https://drscotthawley.github.io/blog/posts/2023-06-12-RVQ.html>)

audiopalm (google)





- Step 1: Pretrained PaLM
- Step 2: Fine-Tuning on Speech Data (ASR, S2ST, S2TT, MT, TTS)

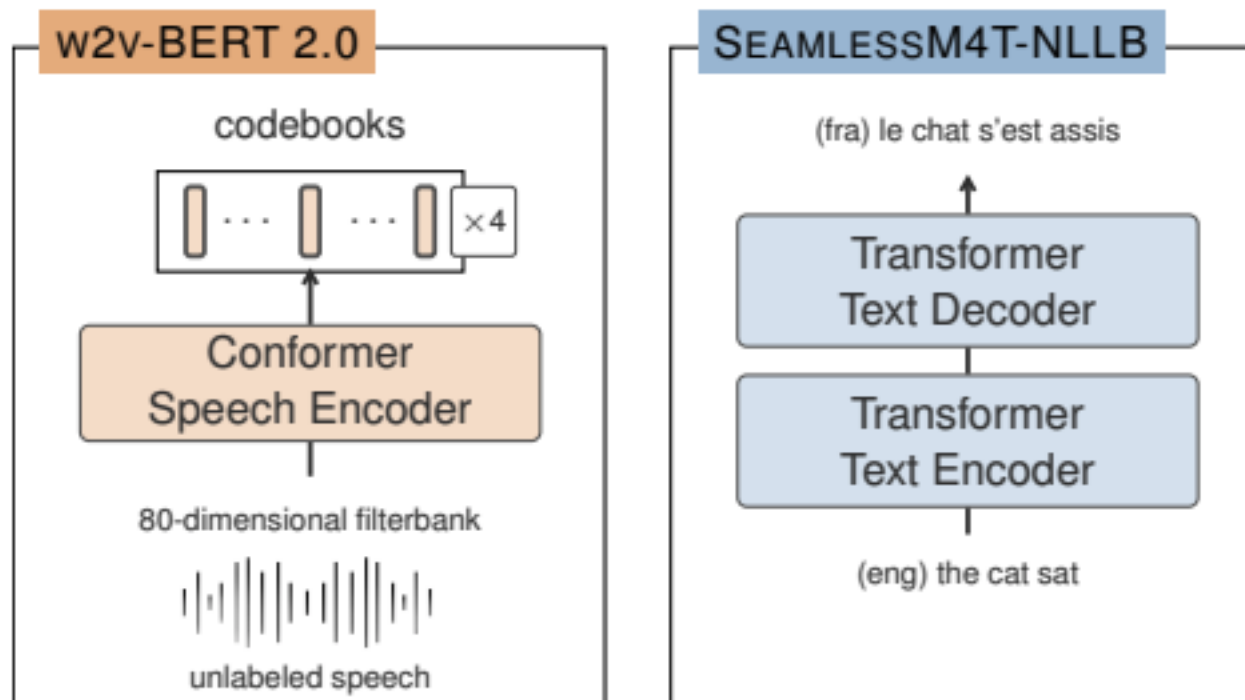
- Pre-trained PaLM
- Pre-trained Audio → audio tokens (wav2vec-BERT)
- Extend embedding matrix with speech tokens
  - audio embeddings initialized randomly
  - train all parameters with text+speech data
- Decode audio tokens
  - autoregressive as in AudioLM
  - non-autoregressive as in SoundStorm
  - prepend with 3 seconds of audio of desired speaker

- Tasks
  - ASR: audio  $\rightarrow$  text
  - AST (S2TT): audio  $\rightarrow$  translated text
  - S2ST: audio  $\rightarrow$  translated audio
  - MT: text  $\rightarrow$  translated text
  - TTS: text  $\rightarrow$  audio
- Task label at beginning of input, including languages  
e.g., [S2ST English French]
  - natural language prompts: no difference
  - naming language helpful for low resource languages
  - also combined labels: [ASR AST S2ST English French]

- Creating audio signal from semantic audio tokens (from wav2vec-BERT)
- Predict acoustic audio tokens
  - 50 Hertz (code rate 20 per second)
  - 12 quantization levels
  - 1024 vocabulary (clusters) per level
- Prepend speaker-specific audio tokens
- Predict waveforms with conformers & all that good stuff
- Non-autoregressive decoding

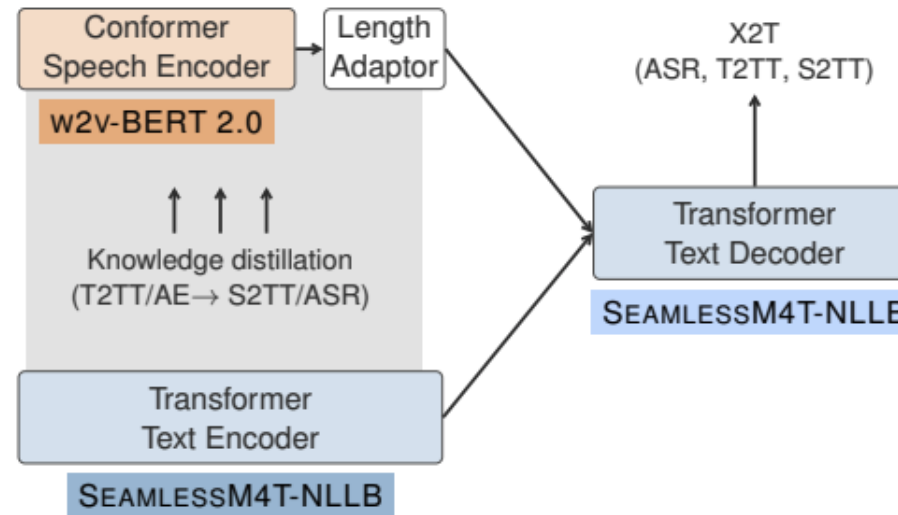
seamless4mt (meta)

# Pretrained Models



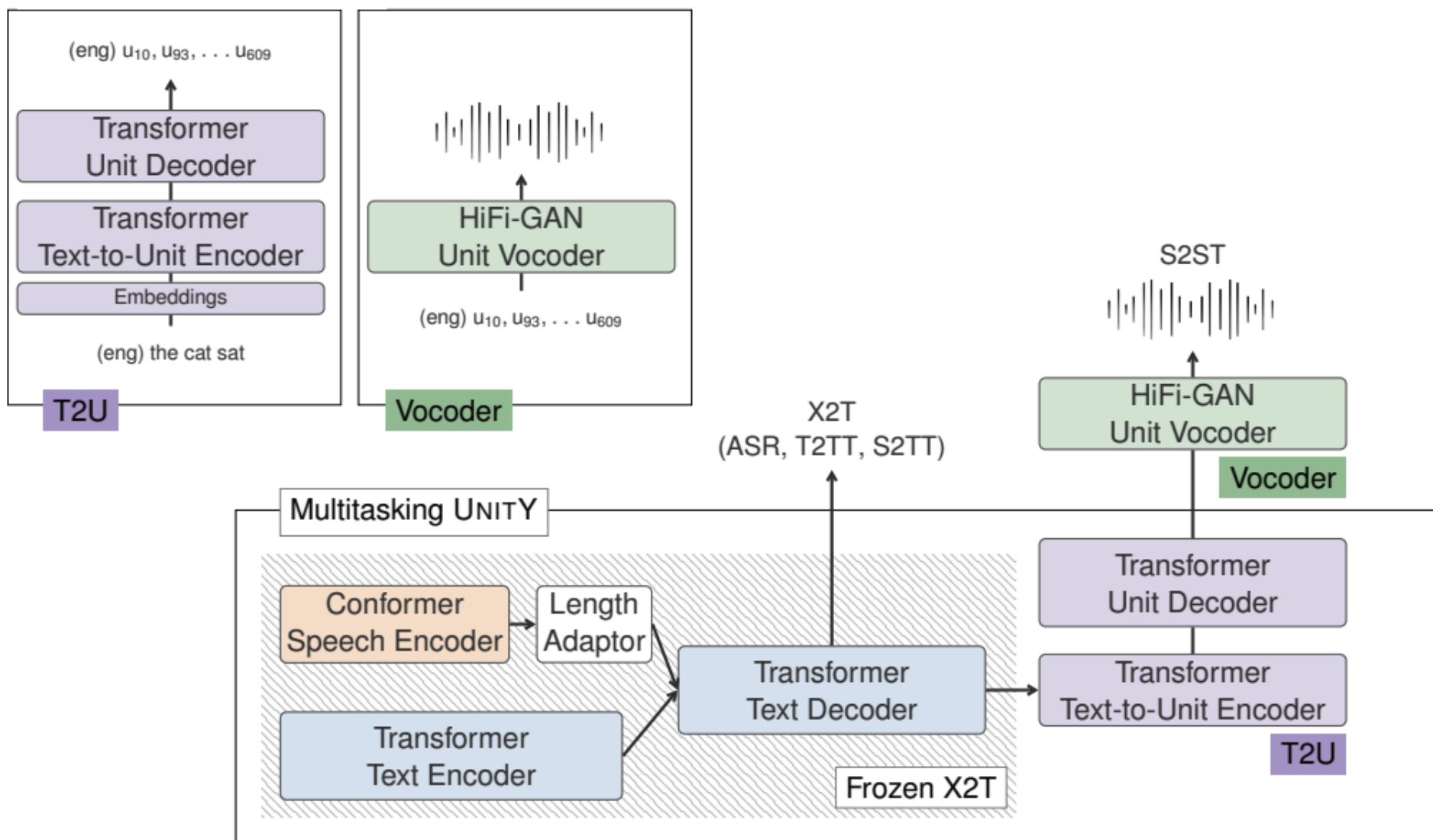
- w2v-BERT 2.0: speech tokenizer
- NLLB: multilingual text translation model

# Speech-to-Text Training



- Multi-task training: T2T, S2T, ASR
- Additional training objective:  
For (speech, transcription, translation) triples T2T and S2T should agree

# Speech-to-Speech Training





# simultaneous speech translation

# Simultaneous Speech Translation

- Start the translation before read all the input speech

I am going to talk today about energy and climate.

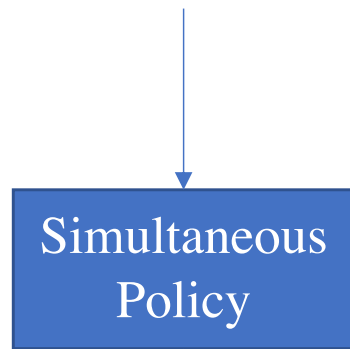
Heute spreche ich zu Ihnen über Energie und Klima.

# Simultaneous Speech Translation

Input

I am going to talk today

about energy and climate.



Read

Write

Output

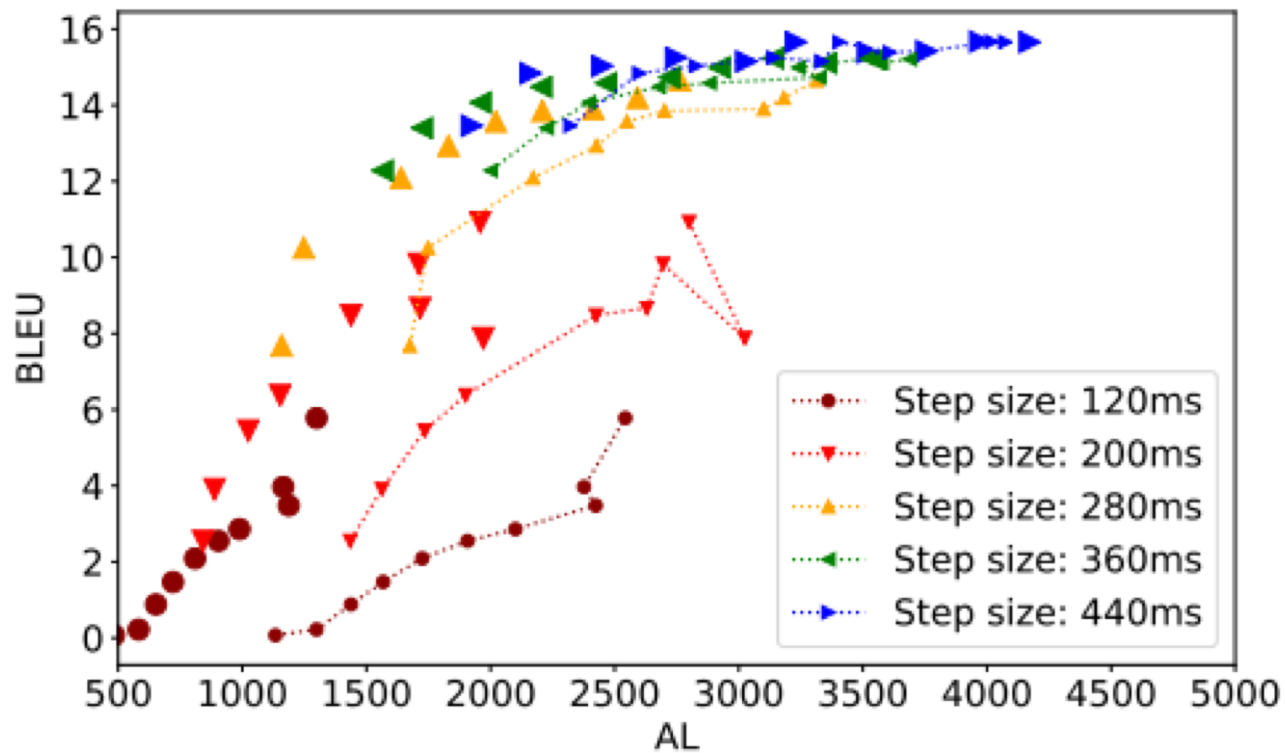
Heute spreche ich

zu Ihnen über Energie und Klima.

# Simultaneous Translation Policies

- **Reinforcement learning** (Gu et al. 2017; Luo et al. 2017; Lawson et al. 2018)
  - Less stable learning process.
- **Fixed policy** (Cho and Esipova 2016; Ma et al. 2019a)
  - Weaker performance, for instance Wait-K (Ma et al. 2019a).
- **Monotonic attention** (Raffel et al., 2017; Arivazhagan et al., 2019; Ma et al., 2020)
  - The State of the art for the task.

# Quality-Latency Trade-off



The lagging behind an oracle/perfect system

# Thank You

41



# questions?