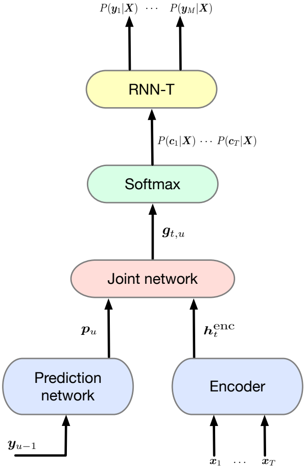
ASR with large language models

Peter Bell

Automatic Speech Recognition – ASR Lecture 18
20 March 2025

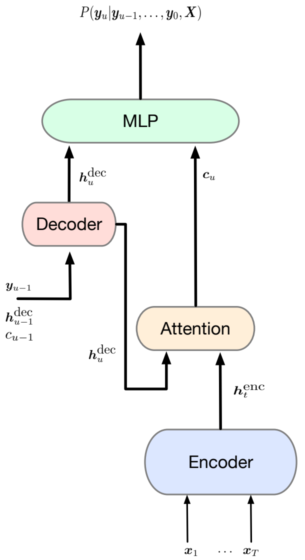
Recap: RNN-T

- **Encoder:** Acoustic model network mapping acoustic features to hidden vectors $h^{\text{enc}} = h_1^{\text{enc}}, \ldots, h_T^{\text{enc}}$.
- **Prediction network**: Recurrent network which takes the previous output subword label y_{u-1} as input and predicts the next subword label p_u
- **Joint network**: Computes a joint hidden vector by a applying a shallow feed-forward net to h^{enc} and p_u
- Inference operates using dynamic programming over time and output labels



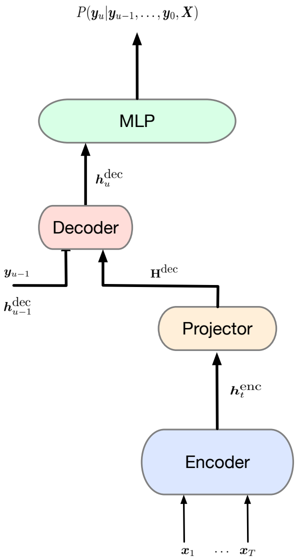
Recap: Encoder-Decoder Model

- **Encoder:** Acoustic model using a recurrent network to map acoustic features $X = x_1, \ldots, x_T$ to hidden vectors $h^{\text{enc}} = h_1^{\text{enc}}, \ldots, h_T^{\text{enc}}$.
- **Decoder**: Computes distribution over labels conditioned on previously predicted labels and the acoustics, $P(y_u | y_{u-1}, \ldots, y_0, X)$
- Inference operates using output label clock only
- Attention mechanism incorporates relevant information from encoded sequence, conditioned on decoder state



“Decoder only” model

- **Decoder**: Computes distribution over labels conditioned on previously predicted labels and the acoustics, $P(y_u | y_{u-1}, \ldots, y_0, X)$
- No (cross) attention mechanism: Information from encoded sequence $h_1^{\text{enc}}, \ldots, h_T^{\text{enc}}$ is project to a fixed embedding H^{enc}, or a sequence that is word-like in length.
- Projected encoder embedding is prepended to the decoder input
- Inference again operates using output label clock only



End-to-end vs factorised models

- Traditional HMM systems are generative models, easy to incorporate human knowledge
- Fully-differentiable E2E models allow all parameters to be optimised towards a single objective, but assume the presence of speech data
- Self-supervised speech models can learn good abstract representations of speech with a lot of audio data – but is it sufficient for ASR?

All models try to solve the problem that speech and text sequences are very different lengths, with unknown alignment and potentially long-span dependencies.

“Fundamental Equation of Speech Recognition”

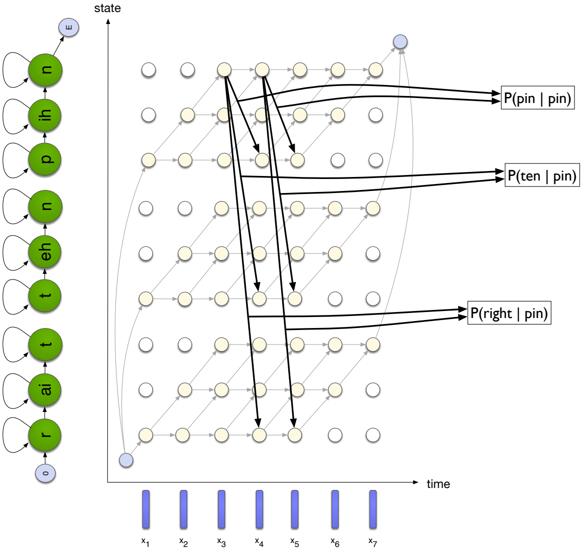
If X is the sequence of acoustic feature vectors (observations) and W denotes a word sequence, the most likely word sequence W^* is given by

$$W^* = \arg\max_W P(W \mid X)$$

Applying Bayes’ Theorem

$$W^* = \arg\max_W \underbrace{p(X \mid W)}_{\substack{\text{Acoustic}\\ \text{model}}} \quad \underbrace{P(W)}_{\substack{\text{Language}\\ \text{model}}}$$

Viterbi search with a bigram language model



Training data considerations

When building an state-of-the-art ASR system, it's important to consider what data and pre-trained models you have available, and how well each is matched to your use case

Limited transcribed data, restricted domain

$\rightarrow$ HMM-DNN model

Lots of transcribed speech data from target domain

$\rightarrow$ Neural E2E model

Lots of untranscribed audio

$\rightarrow$ self-supervised speech representation

General-purpose application

$\rightarrow$ large language model?

The neural decoder as a language model

A conventional LM models

$$P(W) = P(w_1, \dots, w_N) = \prod_{i=1}^{N} p(w_i | w_1, \dots w_{i-1})$$

Or equivalently:

$$P(Y) = P(y_1, \dots, y_U) = \prod_{u=1}^{U} p(y_i | y_0, \dots y_{u-1})$$

where Y is a sequence of tokens.

We can generate a word sequence by sampling from this distribution.

The decoder as an ASR system

We wish to condition the output generated from the LM on the acoustic sequence X:

$$P(Y|X) = P(y_1, \ldots y_U|X) = \prod_{u=1}^{U} p(y_i|y_0, \ldots y_{u-1}, X)$$

whilst still being able to train the LM on (lots of) text data. How?

The decoder as an ASR system

We wish to condition the output generated from the LM on the acoustic sequence X:

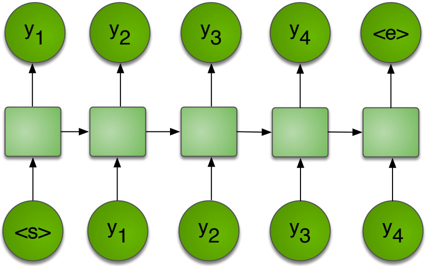
$$P(Y|X) = P(y_1, \dots y_U|X) = \prod_{u=1}^{U} p(y_i|y_0, \dots y_{u-1}, X)$$

whilst still being able to train the LM on (lots of) text data. How?

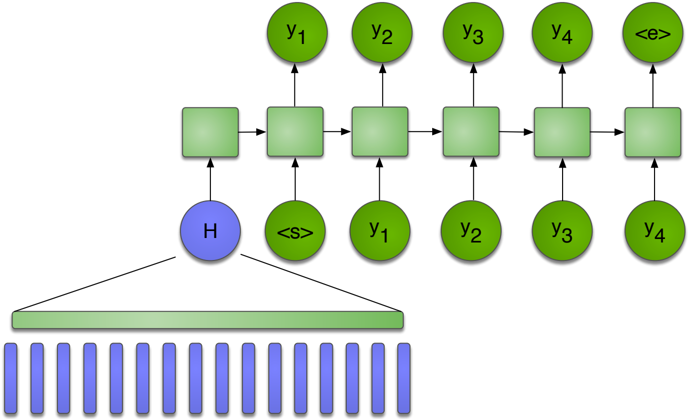
Solution:

- Use a pre-trained (and fixed) acoustic encoder
- Project the encoder output to the same length/embedding space as text $\rightarrow$ can be used directly as input to the LM

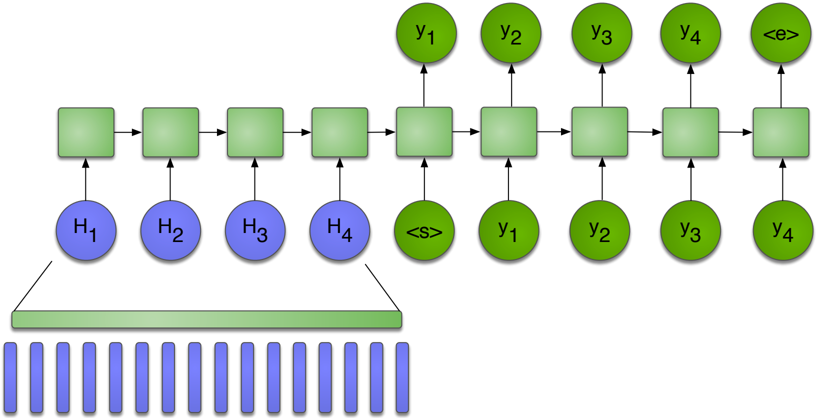
Decoder prepending



Decoder prepending



Decoder prepending



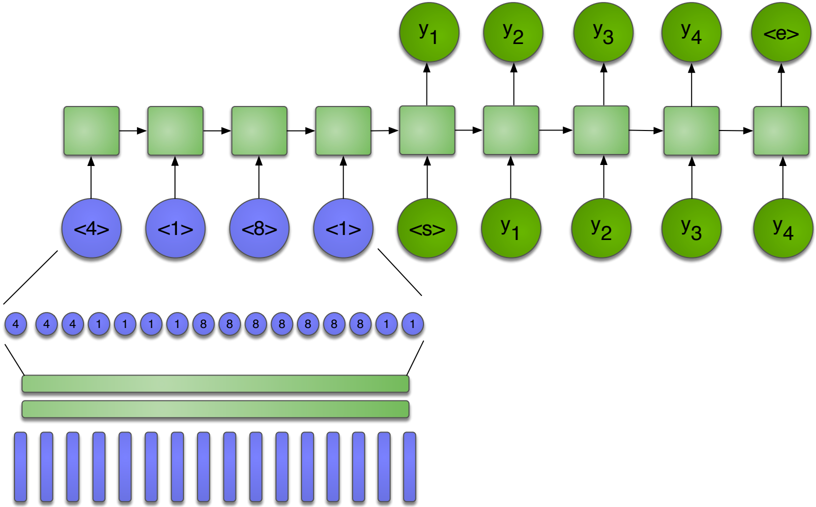
Methods for projecting the acoustic embedding

- Discretized representations (eg. Zhang et al)
- CTC-like compression (eg. Wu et al)
- Downsampling with a fixed factor

Discretized representations

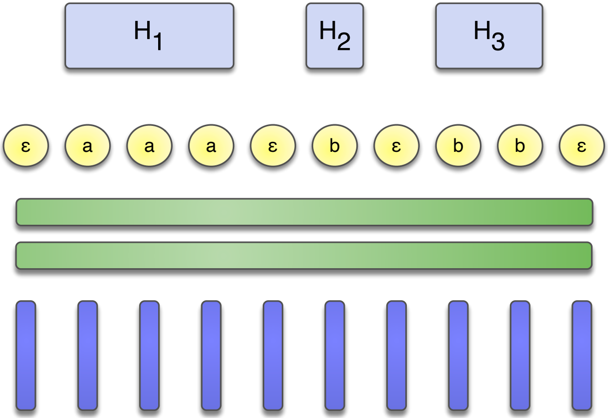
- Use a self-supervised speech representation that produces a sequence of discrete units (eg. HuBERT)
- Remove adjacent duplicate indices
- Expand the vocabulary of the LLM to incorporate the discrete unit inventory

Discretized representations



CTC compression

Use outputs of a pre-trained CTC model to determine which encoded frames to remove or merge.



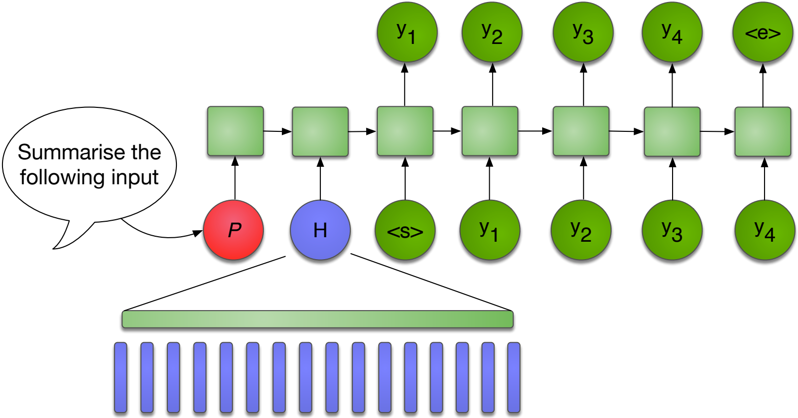
Instruction-tuned models

Instruction-tuning allows LMs to perform diverse NLP tasks in a "zero shot" fashion:

$$P(Y|X,\mathcal{P}) = P(y_1,\dots y_U|X) = \prod_{u=1}^{U} p(y_i|y_0,\dots y_{u-1},X,\mathcal{P})$$

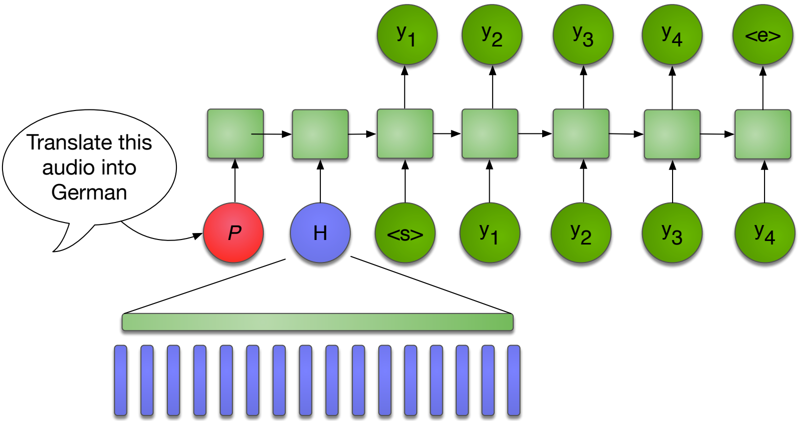
Instruction-tuned models

Instruction-tuning allows LMs to perform diverse NLP tasks in a "zero shot" fashion.



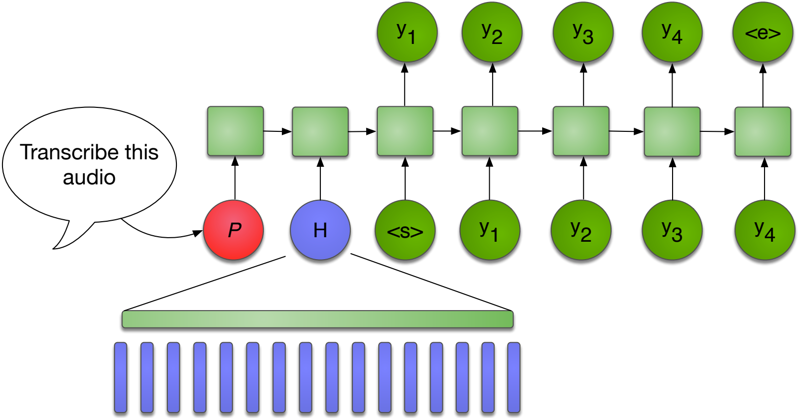
Instruction-tuned models

Can be used to integrate speech input into other downsteam systems $\rightarrow$ avoids error propagation that can happen with a cascaded system



Instruction-tuned models

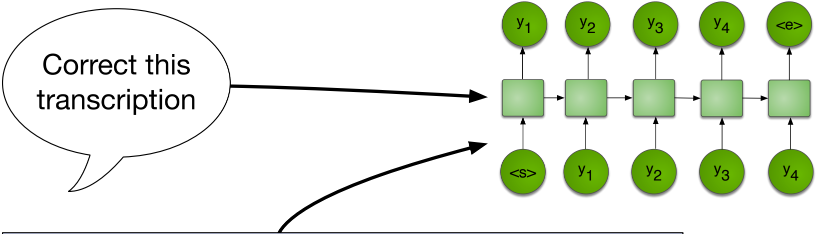
But it can also be used to produce speech transcriptions in a zero-shot fashion without any fine-tuning of the LLM.



Additional details

- Both self-supervised and supervised speech encoders have been successfully used
- Important that the compressed embeddings are monotonic to match the left-to-right nature of generative LMs
- Typically the LM parameters are frozen during projection or fine tuning of the encoder, but LoRA can be used to update the LM afterwards
- The exact training regime depends on the type of data available
- Many recent models are also capable of producing speech output

Directly correcting ASR output



`You look at drugs that competitively antagonise the nicotinic alkaline receptor`

Correcting ASR output: examples

ASR: so this patient does have signs of glaucomatsopsy neuropathy
LLM: so this patient does have signs of glaucomatous optic neuropathy

Correcting ASR output: examples

Uncorrected ASR Output

1: You look at drugs that competitively antagonise the nicotinic **alkaline** receptor.
2: What concentration of **stickmen** do you want to add?
3: So a reminder on the process of **a star calling** release.

terms: ["acetylcholinesterase", "**acetylcholine**", "acetate", "acetic", "acetyl", "energy", "nicotinic", "**neostigmine**", "presynaptic"]

LLM Output with List of Terms

- **1:** You look at drugs that competitively antagonise the nicotinic **acetylcholine** receptor.
-
- **2:** What concentration of **acetylcholine** do you want to add
-
- **3:** So a reminder on the process of **acetylcholine** release.
-

LLM Output without List of Terms

-
- **2**: What concentration of **stilbenes** do you want to add?

Correcting ASR output: examples

HUMAN: here we find Seung et al. and they looked at 144 eyes with early glaucoma
ASR: Here we find Sung Etel and they
LLM: here we find Sung et al. and they looked at 144 eyes with early glaucoma

Correcting ASR output: examples

REF:	so * *** ** cardiff cards will cost in the region of over 600 pounds whereas
LLM history:	so a set of cardiff cards will cost in the region of over 600 pounds whereas
LLM sentences:	so a card of cards will cost in the region of over 600 pounds whereas

Summary

- LLMs can be a powerful tool modern ASR
- Seamless integration of speech inputs many downstream tasks and avoid error propagation
- Even simple approaches can work very well when the LLM is very powerful
- But think carefully about what data is available when deciding on an approach to take

Backround Reading

- Zhang et al. (2023), "SpeechGPT: Empowering Large Language Models with Intrinsic Cross-Modal Conversational Abilities", Findings of EMNLP
  https://aclanthology.org/2023.findings-emnlp.1055.pdf
- Wu et al. (2023), "On decoder-only architecture for speech-to-text and large language model integration", Proc. ASRU https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10389705